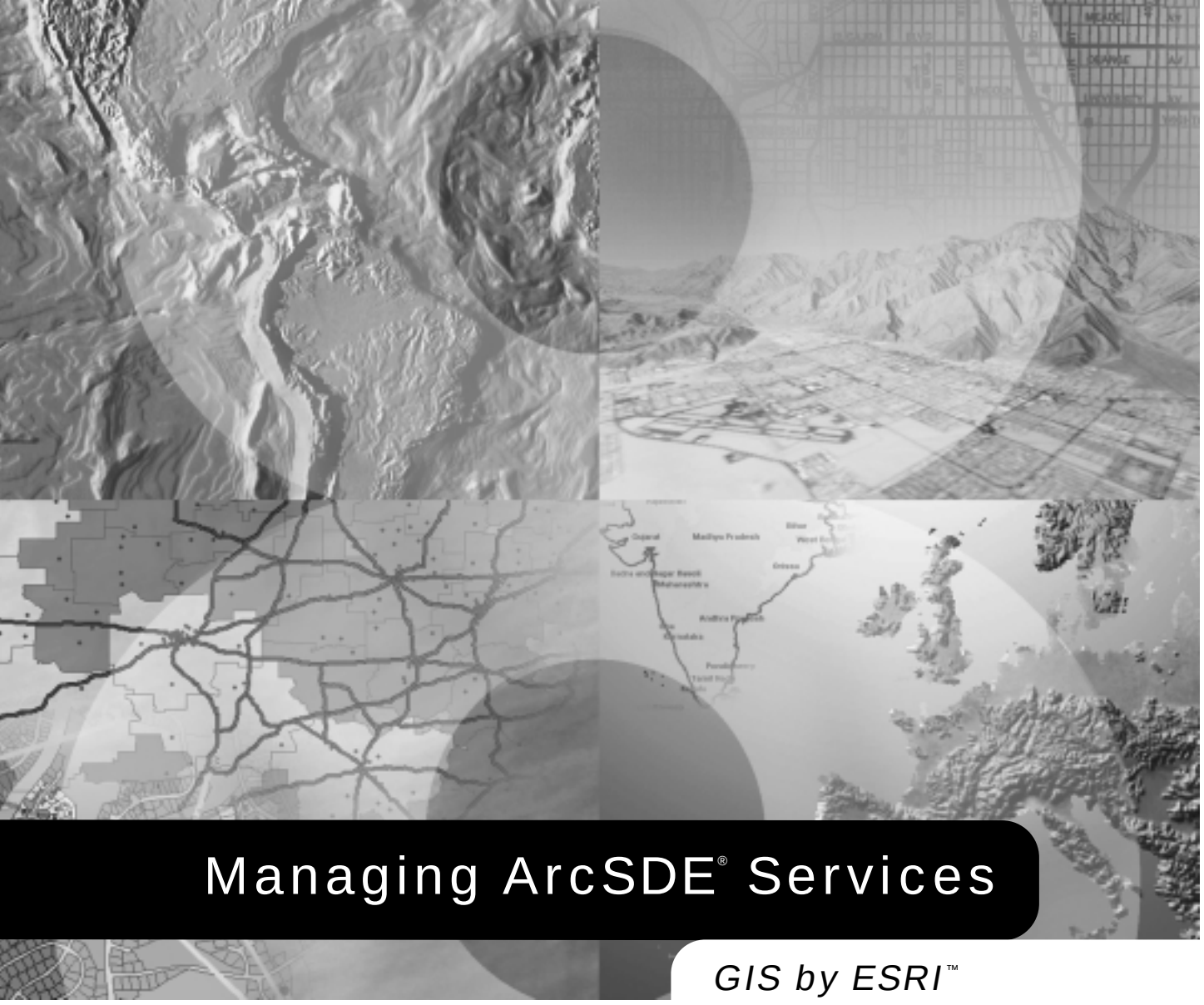


Managing ArcSDE® Services

GIS by ESRI™

Copyright © 1999, 2000–2002 ESRI
All rights reserved.
Printed in the United States of America.

The information contained in this document is the exclusive property of ESRI. This work is protected under United States copyright law and other international copyright treaties and conventions. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or by any information storage or retrieval system, except as expressly permitted in writing by ESRI. All requests should be sent to Attention: Contracts Manager, ESRI, 380 New York Street, Redlands, CA 92373-8100, USA.

The information contained in this document is subject to change without notice.

U.S. GOVERNMENT RESTRICTED/LIMITED RIGHTS

Any software, documentation, and/or data delivered hereunder is subject to the terms of the License Agreement. In no event shall the U.S. Government acquire greater than RESTRICTED/LIMITED RIGHTS. At a minimum, use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in FAR §52.227-14 Alternates I, II, and III (JUN 1987); FAR §52.227-19 (JUN 1987) and/or FAR §12.211/12.212 (Commercial Technical Data/Computer Software); and DFARS §252.227-7015 (NOV 1995) (Technical Data) and/or DFARS §227.7202 (Computer Software), as applicable. Contractor/Manufacturer is ESRI, 380 New York Street, Redlands, CA 92373-8100, USA.

ESRI, SDE, ArcView, MapObjects, the ESRI globe logo, ArcInfo, ArcSDE, ArcCatalog, ArcMap, ArcGIS, ArcEditor, ArcIMS, ArcStorm, GIS by ESRI, the ESRI Press logo, and www.esri.com are trademarks, registered trademarks, or service marks of ESRI in the United States, the European Community, or certain other jurisdictions.

The names of other companies and products herein are trademarks or registered trademarks of their respective trademark owners.

Contents

1 Introducing the ArcSDE service 1

The ArcSDE service 2
The ArcSDE service 3
Properties of an ArcSDE service 5
Tips on learning about the ArcSDE service 7

2 Creating ArcSDE services 9

The ArcSDE home directory 10
The DBMS database 11
The ArcSDE DBMS administration account 12
The ArcSDE service on UNIX and Windows NT 13
Windows NT installs 14
Accessing an ArcSDE service through a firewall 16

3 Configuring ArcSDE services 17

The services.sde file 18
Operating system services files 20
The ArcSDE service name on Windows NT 21
ArcSDE DBMS environment variables 22
ArcSDE system environment variables 23
The dbinit.sde file format 24
Displaying ArcSDE system environment variables 25
Adjusting ArcSDE service initialization parameters 26
Displaying the ArcSDE initialization parameters 36

4 Managing ArcSDE services 37

Before starting an ArcSDE service 38
Starting a local ArcSDE service on Windows NT 39
Starting a remote ArcSDE service on Windows NT 40
Starting a local ArcSDE service on UNIX 41
The sdemon command output 42
Starting a remote ArcSDE service on UNIX 43

Pausing, resuming, and shutting down an ArcSDE service 44
Removing ArcSDE sessions (Windows NT and UNIX) 47

5 Monitoring ArcSDE services 49

Displaying ArcSDE service status and lock table information 50
How ArcInfo uses ArcSDE locking 52
Displaying ArcSDE service statistics 53
Displaying ArcSDE user session information 54

6 Troubleshooting the ArcSDE service 55

What happens when you start an ArcSDE service 56
ArcSDE Windows NT license manager 58
Listing ArcSDE license manager settings and the license manager status on UNIX 59
What happens when an ArcSDE application connects (three-tiered) 61
What happens when an ArcSDE application connects (two-tiered) 63
Common ArcSDE startup problems on UNIX servers 64
Common ArcSDE startup problems on Windows NT servers 66
The Windows NT Event Viewer 70
Examining the ArcSDE error log files 71
ArcSDE intercept 72

Appendix A: ArcSDE home directory 73

Appendix B: ArcSDE data dictionary 83

Appendix C: ArcSDE table definitions 99

Appendix D: ArcSDE service command references 113

Appendix E: ArcSDE initialization parameters 123

Glossary 127

Index 137

Introducing the ArcSDE service

1

IN THIS CHAPTER

- **The ArcSDE service**
- **Properties of an ArcSDE service**
- **Tips on learning about the ArcSDE service**

This chapter introduces the key components of the ESRI® ArcSDE® service and provides an overview of the underlying service configuration. Subsequent chapters discuss the creation, configuration, and management of the ArcSDE service. The appendices contain descriptions of the ArcSDE home directory, the data dictionary, table definitions, listings of the ArcSDE commands, and the ArcSDE initialization parameters.

The ArcSDE service

An ArcSDE service conveys spatial data between geographic information system (GIS) applications and a database. The database may be any one of the supported database management systems (DBMSs) such as Oracle®, SQL Server™, Informix®, or DB2®. The database could also be a registered collection of ArcGIS™ coverages, or shapefiles. The applications that can connect to and access spatial data from an ArcSDE service include ArcGIS, ArcEditor™, MapObjects®, ArcIMS™, and ArcSDE CAD Client, as well as custom-built applications created by either you or an ESRI business partner.

Two-tiered vs. three-tiered architecture

At 8.1, ArcSDE for Oracle8*i*™ and SQL Server provide two alternative methods of connecting to the database instance—direct connection to the database instance in a two-tiered architecture or connecting to the ArcSDE service in a three-tiered architecture. If it is your intention to adopt the two-tiered approach then you need read no further. This guide is solely concerned with the administration of the ArcSDE service. Refer to the *ArcSDE Configuration and Tuning Guide* for your DBMS for more information regarding direct connection.

The diagram on page 3 illustrates the three-tiered architecture, while the diagram on page 4 illustrates the two-tiered (direct connect) architecture.

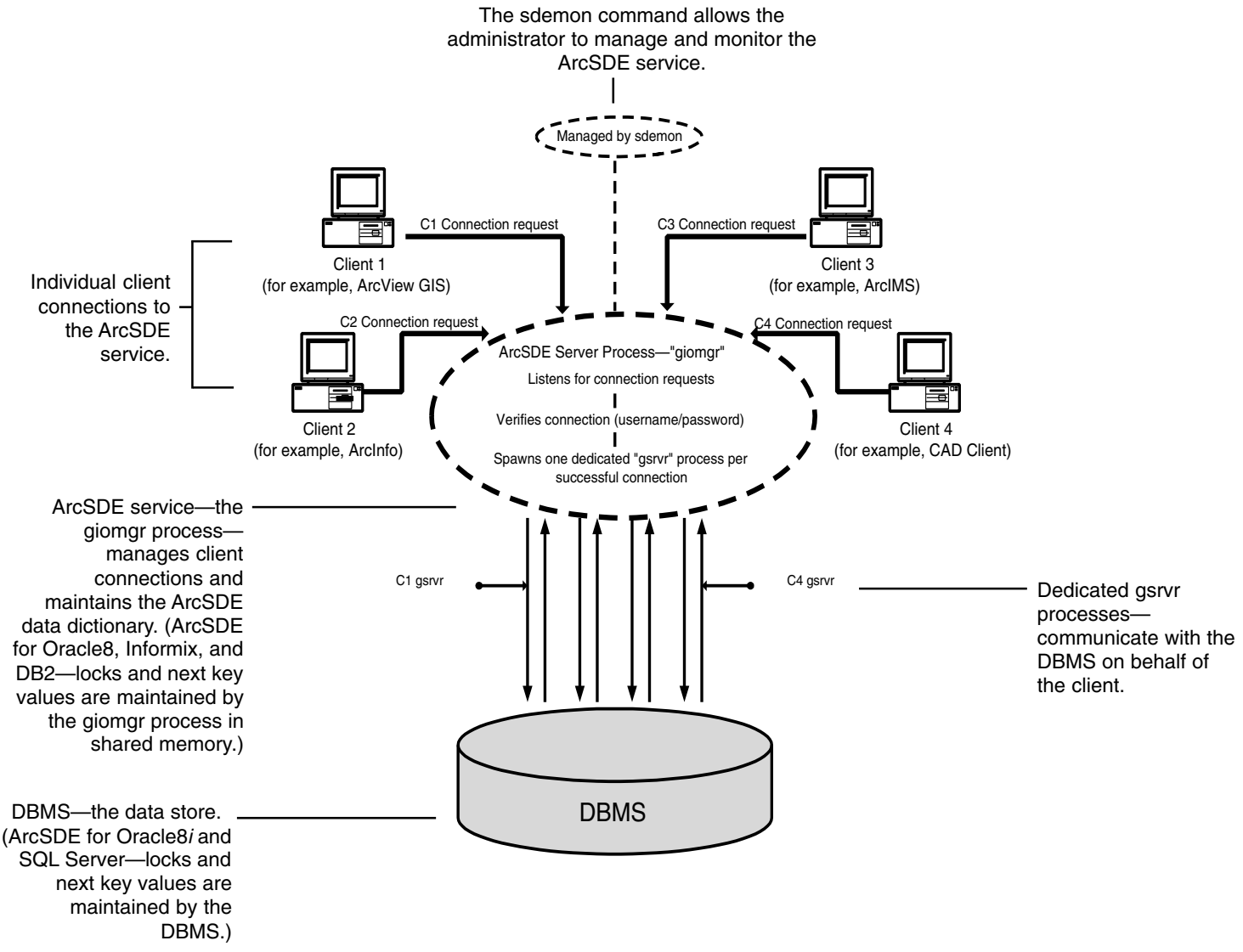
Generating the locks and the key values within the DBMS allows them to be replicated through a distributed database. The two-tiered architecture can only be enabled on an ArcSDE database that can support a Database Service. Therefore at 8.1, the two-tiered option is only available for ArcSDE for Oracle8*i* and SQL Server.

Lock and key value generation

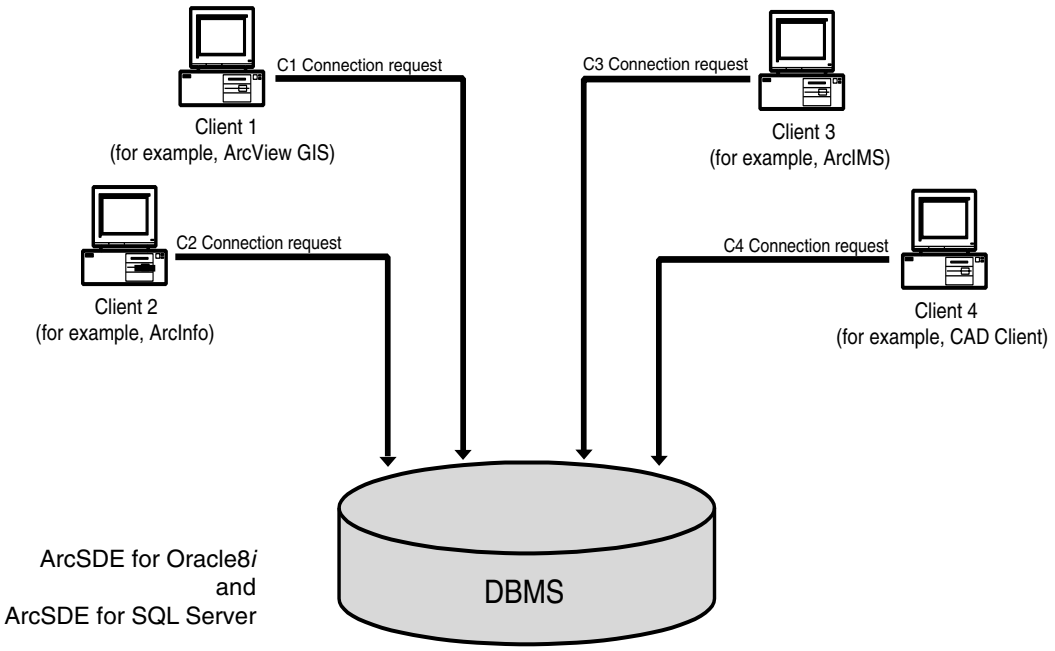
ArcSDE for Oracle8™, ArcSDE for Informix, and ArcSDE for DB2 generate locks and keys with the giomgr process. The DBMS is unaware of the locks of the next value in the key sequence.

ArcSDE 8.1 for Oracle8*i* and SQL Server generates locks and key values within the DBMS. This change in behavior was required to enable two-tiered architecture and to enable DBMS fail-over technology.

The ArcSDE service



Direct connect implementation



Properties of an ArcSDE service

The ArcSDE service, a client/server configuration, has a number of defining properties; these are introduced in the following sections.

The home directory

Each ArcSDE service has its own home directory, which is defined by the system variable, SDEHOME. The home directory contains the ArcSDE binary executable files, dynamic shared libraries, configuration files, and internationalization code pages. The contents of the ArcSDE home directory are listed in 'Appendix A: ArcSDE home directory'.

The ArcSDE server monitor—the giomgr process

Each ArcSDE service has one giomgr process. This process listens for user application connection requests, spawns gsrvr processes, cleans up disconnected user processes, and maintains the ArcSDE data dictionary.

The giomgr will not start if a valid ArcSDEServer feature is not in the license file. Contact ESRI customer support if you have not already obtained a valid license.

The gsrvr process

The giomgr process spawns a gsrvr process for each application connected to the ArcSDE service. The gsrvr process is dedicated to a single-user application connection. It communicates with the database on behalf of the connected application. The gsrvr process responds to the application's query and edit requests to the database.

An ArcSDE application connected directly to the ArcSDE database will not be able to write to it unless a valid ArcSDEServer feature is in the license file. However, a read-only direct connection can be established if the ArcSDEServer feature cannot be found.

The Transmission Control Protocol/Internet Protocol (TCP/IP) service name and port number

The ArcSDE service, through the giomgr process, listens for application connection requests on a dedicated TCP/IP service name and port number. The gsrvr process, in turn, communicates with the application on the same TCP/IP service and port number. All communication between the applications and the ArcSDE service take place using the TCP/IP service name and port number.

ArcSDE for Oracle8, Informix, and DB2 service

Each writable ArcSDE service connects to one database. A database can have only one writable ArcSDE service connected to it. Multiple writable connections are prevented through a locking mechanism. Once a writable ArcSDE service has started and the connection to the database has been established, the service locks the database. If any attempt is made to start a second writable ArcSDE service connected to the same database, an error results and the second ArcSDE service automatically shuts down.

Multiple read-only ArcSDE services can be started on a database. An ArcSDE service is made read-only by setting the READONLY parameter in the giomgr.defs file to TRUE. Therefore, you may start only one writable ArcSDE service and one or more read-only ArcSDE services on a single database.

ArcSDE for Oracle8*i* and SQL Server service

Any number of writable and read-only ArcSDE services can connect to one database. An ArcSDE service is made read-only by setting the giomgr.dbf parameter READONLY to TRUE.

The configuration files

The ArcSDE service can be configured to control the number of application connections as well as the operating system resources each connection may obtain. It can also be configured to optimize the data flow between the connected applications and the database.

Tips on learning about the ArcSDE service

About this book

This book is designed to help you create and maintain an ArcSDE service. It discusses the ArcSDE service—how to create the service, as well as how to monitor its usage. In addition to showing you how you can create, configure, and manage your ArcSDE service, it also shows you how to monitor your service and how to cope with problems that may arise.

The focus of this book is not the creation or administration of a DBMS. This is a separate topic; for information on it, see the *ArcSDE Configuration and Tuning Guide for <DBMS>* PDF file, config_tuining_guide_<DBMS>.pdf, located in the documentation folder on the ArcSDE CD.

Contacting ESRI

If you need to contact ESRI for technical support, see the product registration and support card you received with ArcSDE or refer to 'Getting technical support' in the online Help system's 'Getting more help' section.

You can also visit ESRI on the Web at www.esri.com for more information on ArcSDE and ArcInfo™.

ESRI education solutions

ESRI provides educational opportunities related to geographic information science, GIS applications, and technology. You can choose among instructor-led courses, Web-based courses, and self-study workbooks to find education solutions that fit your learning style and pocketbook. For more information, go to www.esri.com/education.

Creating ArcSDE services

2

IN THIS CHAPTER

- **The ArcSDE home directory**
- **The DBMS database**
- **The ArcSDE DBMS administration account**
- **The ArcSDE service on UNIX and Windows NT**
- **Windows NT installs**
- **Accessing an ArcSDE service through a firewall**

When you install ArcSDE, the default ArcSDE service is created. The directory into which the ArcSDE software and ancillary directories are copied is called the home directory, or SDEHOME, of the default ArcSDE service. For a complete list of the directories and files copied to the ArcSDE home directory, refer to 'Appendix A: ArcSDE home directory'.

Before you start an ArcSDE service, check the following:

- The ArcSDE software has been installed correctly, and the default home directory (SDEHOME) has been established. On Windows NT®, any installation errors will be reported to the <SDEHOME>\SDE81ORA.LOG file.
- There is a database management system (DBMS) database, which is required by ArcSDE products, other than ArcSDE for Coverages.
- An ArcSDE user account exists in the DBMS with enough free space available to store the ArcSDE repository.
- A unique Transmission Control Protocol/Internet Protocol (TCP/IP) service name and port number is available.

The ArcSDE home directory

The ArcSDE home directory, also known by the system variable that stores its location (either %SDEHOME% on Windows NT or $SDEHOME on UNIX®), contains all of the executable files and dynamic libraries required to run an ArcSDE service. This directory structure is automatically set up during the installation process. Each independent ArcSDE service requires its own home directory.

If the host computer on which ArcSDE is installed is going to run a single ArcSDE service, all that is required is to edit the configuration files. However, if the host computer is going to support multiple ArcSDE services, you must create additional home directories and copy the directories found under the default home directory into the new home directories using the operating system's Copy command.

The ArcSDE service's configuration files must then be updated to reflect the unique properties of the new service. Many ArcSDE services can run on a single computer, with each service requiring its own home directory, TCP/IP service name, and port number. The hardware platform must have sufficient resources available to support the processes of the ArcSDE software and the DBMS.

The DBMS database

ArcSDE stores data in a DBMS database. This database must exist before the ArcSDE service can be created. Consult your DBMS documentation for guidance on creating a DBMS database. You should also review the basic advice provided in the *ArcSDE Configuration and Tuning Guide for <DBMS>,* which is located in the documents directory of the CD–ROM installation media and in the ArcSDE home directory (SDEHOME).

Microsoft® SQL Server, Informix, and IBM DB2 support server may contain one or more databases, although be aware that DBMS vendors vary in their definition of a database. An ArcSDE service can only connect to one database server and to one database.

Configuration parameters that control the manner in which an ArcSDE service connects to a database may be stored in the dbinit.sde file. This file is located in the %SDEHOME%\etc directory on Windows NT and in the $SDEHOME/etc directory on UNIX. This avoids the need to rely on environment variables set when the user logs in. The dbinit.sde file is discussed in detail in Chapter 3, ‘Configuring ArcSDE services’.

For optimum performance, the database and the ArcSDE service should coexist on the same host. However, it is possible for ArcSDE to connect to a database hosted on a remote machine. For more information on establishing a remote connection to an ArcSDE service, please refer to the *ArcSDE Configuration and Tuning Guide for <DBMS>*.

The ArcSDE DBMS administration account

ArcSDE service requires the creation of an ArcSDE DBMS administration account. On Windows NT, the ArcSDE installation process gave you the opportunity to create an ArcSDE DBMS administration account and the necessary DBMS storage space to store the data dictionary tables owned by the ArcSDE administrator account. If this step was bypassed during the installation, it must be completed before the ArcSDE service can be started up.

The procedure for creating the ArcSDE DBMS administration account differs for each ArcSDE product. For information on how to create the ArcSDE DBMS administration account, please refer to *install_guide* in the documentation directory under the home directory (SDEHOME) for your particular ArcSDE installation. On Windows NT platforms this is a CHM file, while on UNIX platforms the install guide is an HTM file.

Under the ArcSDE DBMS administration account, ArcSDE stores its data dictionary tables. The data dictionary tables may require as much as 100 MB of free space, so it is important that the ArcSDE DBMS user has access to sufficient resources to accommodate this.

The ArcSDE service on UNIX and Windows NT

Each ArcSDE service listens for user connections on a dedicated TCP/IP service name and port number through the giomgr process.

On UNIX systems, edit the $SDEHOME/etc/services.sde file and the /etc/services file to include the name of the ArcSDE service.

On Windows NT systems, the ArcSDE service is created during the ArcSDE software installation as a Windows NT service. The service name is stored in the registry under HKEY_LOCAL_MACHINE\SOFTWARE\ESRI\ArcInfo\ArcSde\8.1\Arcsde for <product>\<service name>.

Tip

Port number

Although it is not used at this stage, the port number in the SDEHOME/etc/services.sde file is included to remind the ArcSDE administrator that the service name is assigned to the 5151/TCP port in the operating system services file.

Registering an ArcSDE service on UNIX

1. Edit the $SDEHOME/etc/services.sde file.
2. Add the same service name and port number to the operating system /etc/services file.

```
#
# ESRI ArcSDE service name and port number
#
esri_sde         5151/tcp ——————————— ❶
```

$SDEHOME/etc/services.sde file

```
.
.
.
nfsd      2049/udp  # NFS server daemon (clts)
nfsd      2049/tcp  # NFS server daemon (cots)
lockd     4045/udp  # NFS lock daemon/manager
lockd     4045/tcp
esri_sde  5151/tcp  # ArcSDE 8 service ——— ❷
dtspc     6112/tcp  # CDE subprocess control
fs        7100/tcp  # Font server
.
.
```

Operating system /etc/services file

Windows NT installs

The installation program automatically enters the service name and port number for the default ArcSDE service in both the registry and the Windows NT services file, C:\winnt\system32\drivers\etc\services.

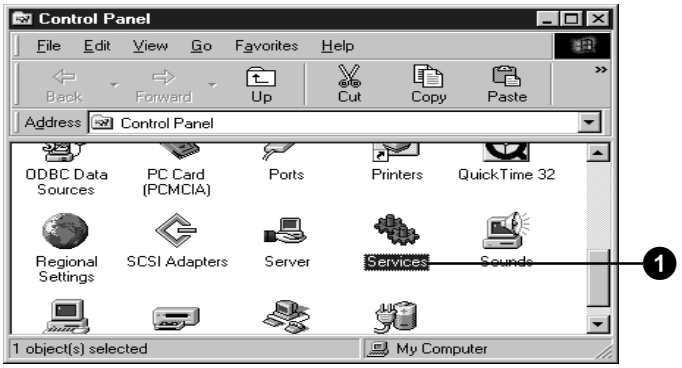
The installation program also creates the Windows NT service, setting it to automatic—that is, it will automatically start up when the server host is rebooted.

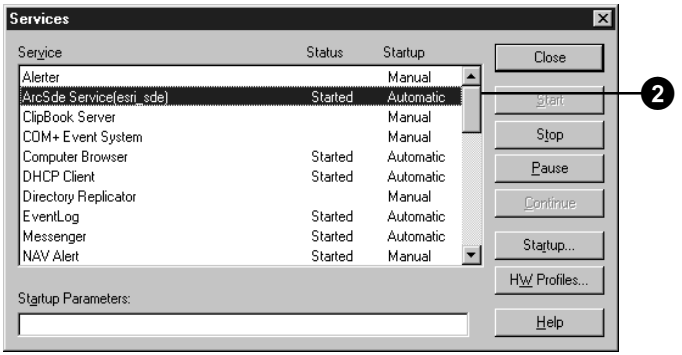
Like all Windows NT services, the ArcSDE service is started and stopped from the Windows NT services menu.

Although it is more convenient to start the ArcSDE service from the Windows NT service menu, it is sometimes necessary to use the sdemon command because error messages are printed directly to the MS–DOS command window instead of the event log. Using sdemon to start the service can be useful when trying to diagnose a startup problem. Unlike the Windows NT service, the sdemon command reads the service name from the %SDEHOME%\etc\services.sde file. Edit this file to ensure that it contains the correct information before using sdemon.

Verifying an ArcSDE service on Windows NT

1. Open the Control Panel and double-click Services.
2. Verify the ArcSDE service has been started.





Tip

Adding more ArcSDE services

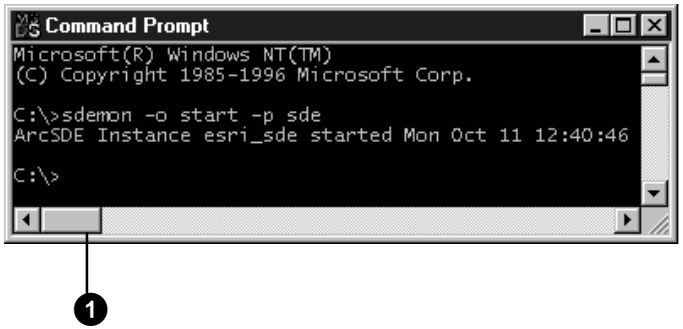
If additional ArcSDE services are required, use the sdeservice command to define the new Windows NT service. For a complete description of the sdeservice command, see 'Appendix D: ArcSDE service command references'.

See Also

ArcSDE will not start unless a license is available for it. When installing ArcSDE on a Windows NT platform, an install wizard guides you through the process of setting up the license manager. Installing ArcSDE on a UNIX platform involves manually setting up the license manager. If you are unfamiliar with the installation and operation of the license manager, please refer to the license_manager_guide.pdf *file located in the documentation directory of the CD–ROM installation media and in the ArcSDE home directory (SDEHOME).*

Monitoring the ArcSDE service

1. Use the sdemon command at the Windows® MS–DOS command prompt to trap ArcSDE service error messages.



Accessing an ArcSDE service through a firewall

To provide access to an ArcSDE service inside a system security *firewall,* the host computer on which the ArcSDE service is installed should be listed in your domain name server (DNS) database. The DNS must be registered with your Internet Service Provider (ISP) or directly with Network Solutions (formerly called InterNIC), the organization that registers Internet domain names.

Your DNS resolves the IP address of your computer to the name, or Universal Resource Locator (URL), you wish to make accessible to the Internet. In most cases, you will have more machines within your local network than you will have Internet IP addresses for. In this case, you would maintain your own set of internal IP addresses known only to your local area network (LAN). Your firewall, or proxy server software, will translate your internal IP addresses to Internet IP addresses when you access computers outside your LAN.

Since ArcSDE services listen for connections on a TCP/IP port number that corresponds to your ArcSDE service name, you must also add the TCP/IP port number to the server name when connecting to it. You can specify an ArcSDE server name in two ways. You can either use the DNS name if it is available, or you can connect to it directly using its Internet IP address.

For example, our domain name—esri.com—has been registered with Network Solutions, and we identified our DNS as IP address 198.102.62.1. Our DNS has the IP address for the ArcSDE server Toshi in its DNS database. The internal IP address for Toshi is 46.1.2.324, which is translated to the IP address 198.102.62.5 when Toshi sends and receives information through the firewall. The ArcSDE service running on Toshi is listening for connections on the service name esri_sde3, which corresponds to TCP/IP port number 5165. So, if you wish to connect to that particular ArcSDE service, you must specify either the server name "toshi.esri.com:5165" or identify Toshi by its IP address "198.102.62.5:5165". In both cases you must also include the service port number, 5165.

The giomgr process bequeaths the port number to the gsrvr process following a successful connection. Therefore, all communication to the ArcSDE service occurs on the same TCP/IP port number.

If you cannot connect to an ArcSDE service through a firewall, test the accessibility of the remote ArcSDE server with your Internet browser by specifying either the server name and TCP/IP port number or the IP address and TCP/IP port number as the URL.

The correct syntax is:

```
<server name>:<port number>
```

```
<IP address>:<port number>
```

Configuring ArcSDE services

3

IN THIS CHAPTER

- The services.sde file
- Operating system services files
- The ArcSDE service name on Windows NT
- ArcSDE DBMS system environment variables
- ArcSDE system environment variables
- The dbinit.sde file format
- Displaying ArcSDE system environment variables
- Adjusting ArcSDE service initialization parameters
- Displaying the ArcSDE initialization parameters

ArcSDE conveys spatial data between a DBMS and applications. Configuring an ArcSDE service focuses on maximizing data transfer between the server and the client with limited shared resources. The requirements of the application, the number of users, and the amount of data requested influence the configuration of the ArcSDE service.

The ArcSDE service can be configured by modifying the parameters in the configuration files found under the SDEHOME\etc directory. The four configuration files are services.sde, dbinit.sde, giomgr.def, and dbtune.sde.

The services.sde file

Each ArcSDE service communicates with the database management system (DBMS) and the applications through a Transmission Control Protocol/Internet Protocol (TCP/IP) port. TCP/IP ports are defined by name in the operating system's services file.

The SDEHOME\etc\services.sde file contains the service name, the unique TCP/IP port number on which the ArcSDE service accepts connection requests. This port number is also assigned to each user or gsrvr process that the ArcSDE service initiates. The port number listed in the services.sde file is not used but is included by convention as a reminder of the port number assigned to the service name in the operating system's services file.

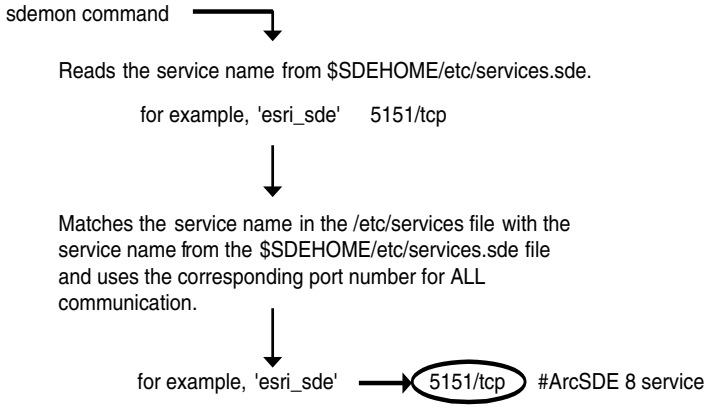
ESRI registered the default esri_sde service name and 5151 TCP/IP port number with the Information Sciences Institute, Internet Assigned Numbers Authority.

The default services.sde file created during the installation process will contain the following:

```
#
# ESRI ArcSDE Remote Protocol
#
esri_sde              5151/tcp
```

To change the default ArcSDE service name, simply edit the file and restart the service.

On UNIX systems the sdeservice.sde file is always used. On Windows NT systems, however, the services.sde file is used only when the service is started from the MS–DOS prompt using the sdemon command. If the service was started from the Windows NT service panel, ArcSDE uses the service name stored in the registry under HKEY_LOCAL_MACHINE\HARDWARE\ESRI\ArcInfo\ArcSDE\8.1\ArcSDE for Oracle\esri_sde.



Mapping the SDEHOME/etc/services.sde entries to the system services file entries

When a match is found, ArcSDE starts the giomgr process. It listens for user connection requests on the TCP/IP port number assigned to the service name.

When the ArcSDE service is started with the sdemon command, the service searches the system services file for a service name that matches the service name in the services.sde file.

If a match is not found, ArcSDE returns the following error on UNIX systems:

```
$ sdemon -o start
Please enter the SDE DBA password: *****
This instance's service name esri_sde not found
in system services file.
```

To diagnose startup problems on a Windows NT platform, examine the event log. For more information, see Chapter 6, 'Troubleshooting the ArcSDE service'.

Multiple ArcSDE services running on the same host require unique service names and port numbers. If you intend to create another ArcSDE service on the same host, you must edit one of the services.sde files and amend the service name and port number accordingly. You must also update the system services file to include the new ArcSDE service.

The services.sde file in each SDEHOME\etc directory should contain only one service name. If you add more than one service name to this file, ArcSDE uses the first one it encounters and ignores the rest.

The operating system services file contains many service names. At least one of them must match your ArcSDE service name in the services.sde file. Enter the service name into the operating system services file on the ArcSDE server and client machines.

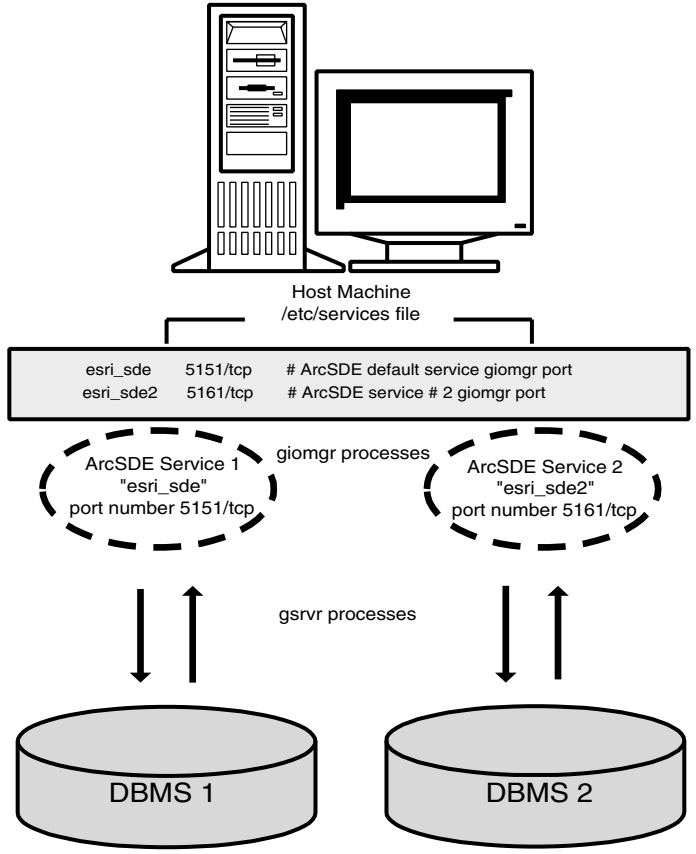
Here is an example from an operating system services file that contains two ArcSDE services:

```
esri_sde 5151/tcp       # ArcSDE def instance
                         # iomgr port.
esri_sde2 5161/tcp      # ArcSDE instance
                         # 2 iomgr port.
```

This defines two service names: esri_sde and esri_sde2.

On UNIX systems, you can use the NIS services file if you are running NIS to avoid unnecessary duplication of effort when updating the client and server local system services files.

The dbinit.sde file of the new service must also be edited so that it contains the connection information of the second DBMS. For more information on setting the DBMS connection variables in the dbinit.sde file, see ‘ArcSDE DBMS system environment variables’ in this chapter.



One host computer supporting two ArcSDE services

Operating system services files

The services file on a Windows NT platform is located under the winnt\system32\drivers\etc directory. Use an editor such as Notepad that does not embed format characters to edit the Windows NT services file.

The services file for UNIX systems is located at /etc/services.

Some UNIX systems direct applications to search the NIS services file rather than the local services file. If you wish to have the operating system search the local services file instead, you must force it to do so.

Forcing a search of local services on HP-UX®

1. Copy the nsswitch.conf file from the /usr/newconfig/etc directory to the /etc directory.
2. Edit the file and change the line 'services: nis files' to 'services: files nis'.

```
# /etc/nsswitch.conf:

#This file uses NIS (YP) in conjunction with
# files.

# "hosts:" and "services:" in this file are used
# only if the /etc/netconfig file has a "-" for
# nametoaddr_libs of "inet" transports.

# the following two lines obviate the "+" entry
# in /etc/passwd and /etc/group.

passwd:      files nis
group:       files nis
# consult /etc "files" only if nis is down.
hosts:       files nis dns
networks:    nis [NOTFOUND=return] files
protocols:   nis [NOTFOUND=return] files
rpc:         nis [NOTFOUND=return] files
ethers:      nis [NOTFOUND=return] files
netmasks:    nis [NOTFOUND=return] files
bootparams:  nis [NOTFOUND=return] files
publickey:   nis [NOTFOUND=return] files
netgroup:    nis
automount:   files nis
aliases:     files nis
# for efficient getservbyname() avoid nis
services:    files nis ——— 2
sendmailvars: files
```

Forcing a search of the local services on IBM® AIX®

1. In the /etc directory, create the file netsvc.conf.
2. Add the line "services=local.nis".

The ArcSDE service name on Windows NT

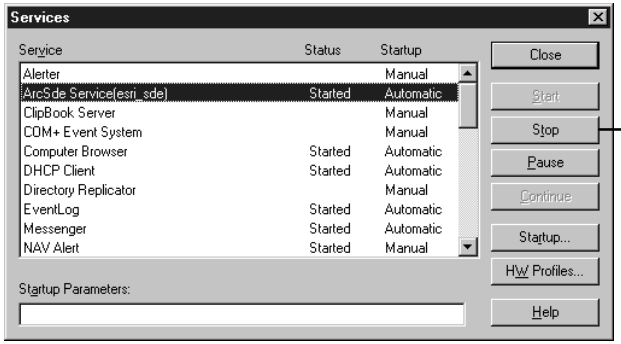
On Windows NT systems, the services.sde file is only accessed when the ArcSDE service is started with the sdemon command. When the service is started from the services menu, the service name is read from the system registry.

To change the service name in the Windows NT registry, use the sdeservice administration command.

See Also

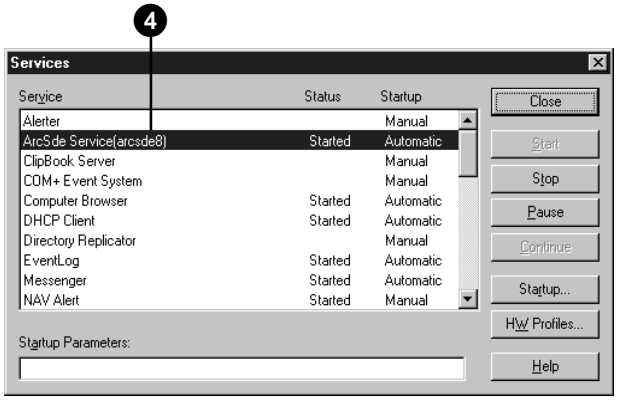
For more information about the sdeservice command, see 'Appendix D: ArcSDE service command references'.

1. Shut down the ArcSDE service from the Windows NT Services dialog box by clicking Stop.
2. From a DOS commmand prompt, remove the old service name using the sdeservice command with the delete option.
3. From a DOS command prompt, create the new service name with the sdeservice -o create option.
4. Restart the ArcSDE service from the Services dialog box. Notice the service name has changed.



2 `sdeservice -o delete -i esri_sde`

3 `sdeservice -o create -p password -l 27005@bongo -d ORACLE8I,bruno -i arcsde8`



ArcSDE DBMS environment variables

The ArcSDE service uses system environment variables in the SDEHOME\etc\dbinit.sde file to establish the DBMS server that it connects to and to limit the messages written to the sde.errlog file. If these system environment variables are not set in the dbinit.sde file, the ArcSDE service uses the variables set within the user's system environment. If the variables are not set in either the dbinit.sde file or the current user environment, default ArcSDE settings are used.

The system environment variables that control the ArcSDE services connection to a DBMS depend on each individual DBMS.

For more details on setting environment variables, see the Install_guide under the SDEHOME documentation folder.

SDEHOME determines the ArcSDE service that the sdemon command will operate on by default. This variable can be overridden by the -H option of the sdemon command.

```
setenv SDEHOME d:\arcexe81\arcsde
```

SDESERVER determines the host of the ArcSDE service for the connecting client application. This variable can be overridden by the -s option of the ArcSDE administration tools. If neither the -s option nor the SDESERVER variable is set, the client application attempts to connect to an ArcSDE service running on the local host.

```
setenv SDESERVER bruno
```

SDEINSTANCE is set in the environment of the client application to which it determines the ArcSDE service name will connect. The -i option of the ArcSDE administration tools overrides this variable. If this variable is not set and the -i option is not applied, the ArcSDE service defaults to the esri_sde service.

```
setenv SDEINSTANCE esri_sde81
```

SDEDATABASE can be entered for DBMS and has multiple database connection possibilities within a single instance. The -D option of the ArcSDE administration tools overrides this variable. If the variable is not set and the -D option is not used, the ArcSDE client connects to the default database.

```
setenv SDEDATABASE city_eng
```

SDEUSER specifies the user name that the ArcSDE client application will connect on. The -u option of the ArcSDE administration tools overrides this variable. If this variable is not set and the -u is not used, an error is returned. A user name must be specified.

```
setenv SDEUSER bob
```

SDEPASSWORD specifies the password for the user name entered by the ArcSDE client application. The -p option of the ArcSDE administration tools overrides this variable. If the variable is not set and the -p option is not provided, the application may prompt for the password. If the application does not prompt for the password, an error is returned.

```
setenv SDEPASSWORD fools.gold
```

SDETMP allows you to set the temp directory for the servers using this variable, but it will only be checked if there is no TEMP keyword in the giomgr.defs file.

```
setenv SDETMP c:\temp
```

SDEPROCSTATS determines the update interval, expressed in seconds, by which the DBS service updates the PROCESS_INFORMATION table with session statistics. By default it is set to -1, which means the statistics are not periodically written to the table but are only written to the giomgr.log file when the sessions disconnect.

```
setenv SDEPROCSTATS -1
```

Set SDEPROCSTATS to 0 if you want sessions to update the PROCESS_INFORMATION table whenever the statistics change.

```
setenv SDEPROCSTATS 0
```

Set SDEPROCSTATS to a value greater than 0 to periodically update the PROCESS_INFORMATION table with the session's statistics.

```
setenv SDEPROCSTATS 10
```

SDEDBECHO echoes the contents of the dbinit.sde file during startup. For ArcSDE services started locally on a UNIX system, the output of SDEDBECHO is written to the screen. The SDEDBECHO output for an ArcSDE service started on a remote UNIX ArcSDE service is written to its sde.errlog file.

```
setenv SDEDBECHO TRUE
```

SDEVERBOSE reports internal messaging to the screen upon startup and writes gsrvr process startup and shutdown messages to sde.errlog.

```
setenv SDEVERBOSE TRUE
```

The dbinit.sde file format

The dbinit.sde file consists of comments and commands. Comments are any lines preceded by the pound sign (#). For example:

```
# This is the system environment for
# the ArcSDE service esri_sde
```

The commands in the dbinit.sde file accept two keywords: set and unset. The Set command enables the system variable and assigns it the value following the equals sign. The syntax of the set command is:

```
set <variable>=<value>
```

In this example, the SDEDBECHO variable is set to true, which echoes the variables set in the dbinit.sde file when the ArcSDE service is started.

```
set SDEDBECHO=TRUE
```

The Unset command disables the system variable. It is useful because it ensures that an undesired variable set in the login environment is not set when ArcSDE starts. The syntax of the unset command is:

```
unset <variable>
```

The following example of the unset command ensures that the Oracle TWO_TASK variable is not set:

```
unset TWO_TASK
```

Displaying ArcSDE system environment variables

To display the environment variables for an ArcSDE service that is currently running, use:

```
$ sdemon -o info -I vars
```

A list of environment variables is returned in the format:

```
<variable>=<value>.
```

Part of the list will look something like this on a UNIX platform:

```
SDEHOME=/cymru1/sde80
ESRI_LICENSE_FILE=27005@ynysmon
```

The list will look something like this on a Windows NT platform:

```
SDEHOME=C:\ProgramFiles\ESRI\ArcInfo\arcsde\sdeexe80
ESRI_LICENSE_FILE=27005@ynysmon
```

The variables in either list are a combination of those found in the login system environment file (.cshrc or .profile file on UNIX systems and the environment tab of the Windows NT control panel's system menu) and the dbinit.sde. The variables set in the dbinit.sde file always override the system environment files settings.

Adjusting ArcSDE service initialization parameters

The ArcSDE service initialization parameters set in the SDEHOME\etc\giomgr.defs file affect such things as the number of connections that can be made to the ArcSDE service, the amount of shared memory to allocate to data buffers, and the connection timeout. The giomgr.defs file is read when the ArcSDE service starts. To change a parameter, edit the file and restart the service.

The default values listed in the giomgr.defs file after you install ArcSDE satisfy the needs of most ArcSDE installations. With the exception of adjusting the MINBUFFSIZE and MAXBUFFSIZE parameters to improve data loading performance, the remainder of the parameters should not be altered unless you are certain you need to do so.

See 'Appendix E: ArcSDE initialization parameters' for a full list of giomgr.defs parameters. The following sections discuss in more detail some of the individual parameters and their impact on the ArcSDE service.

READONLY parameter

The READONLY parameter allows you to start the ArcSDE service in READONLY mode. Set this parameter to TRUE if you want to disable all writes by ArcSDE clients that connect to the services.

When the READONLY parameter is set to FALSE, the default value, the ArcSDE service will allow users to edit the data of the ArcSDE feature classes and tables they have write permissions on.

Session parameters

The session parameters are CONNECTIONS, TEMP, and TCPKEEPALIVE.

The CONNECTIONS parameter restricts the number of concurrent connections a single ArcSDE service will allow. However, if multiple ArcSDE services are running, setting a value for this parameter allows the ArcSDE administrator to balance the load on the system by distributing connections across all available ArcSDE services.

The CONNECTIONS parameter should not be set to higher than the total number of ArcSDE client licenses that have been purchased, as ArcSDE allocates 780 bytes of shared memory for each potential connection. The default is 64.

The TEMP parameter specifies the full path to the directory that the ArcSDE service uses for temporary things, such as the shared memory file on UNIX systems and temporary storage for the attribute binary large objects (BLOBs), that are larger than specified by the BLOBMEM parameter. For more information on this parameter, please refer to the section 'Managing BLOB data' later in this chapter. A directory with at least 5 MB of available space should be specified. More space may be required for ArcSDE services that support applications allowing users to concurrently access binary large objects stored in attribute BLOB columns.

Sometimes the computer on which an application is running crashes or a user unexpectedly terminates an application. Unless the TCPKEEPALIVE parameter is set to TRUE, the ArcSDE server does not detect the absence of the client process and does not disconnect the user. When this happens, the ArcSDE client connection remains as a license checked out, and the ArcSDE administrator must manually terminate the client process to recover it.

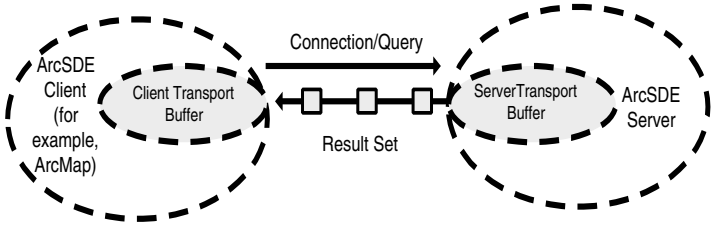
The TCP/IP KEEPALIVE interval is a systemwide parameter that affects not just the ArcSDE service, but every service running in the TCP/IP environment. By setting the ArcSDE TCPKEEPALIVE parameter to TRUE, the system TCP/IP KEEPALIVE settings are used. The default test interval is two hours of idle time—that is, the system checks the connected sessions every two hours.

When this happens, any ArcSDE connections whose client processes have been terminated are disconnected. If the network environment in which the ArcSDE service operates is reliable, TCPKEEPALIVE may be set to TRUE. However, please be aware that a disconnection may be triggered by short-term network outages (~10 minutes) when TCPKEEPALIVE is set to TRUE. By default, TCPKEEPALIVE is set to FALSE.

Transport buffer parameters

After a user connects to an ArcSDE service from an application such as ArcMap™ or ArcCatalog™, they access the feature classes and tables stored in the database. Each time any of these items are accessed, an ArcSDE stream is created. An ArcSDE stream is a mechanism that transports data between the database the ArcSDE service is currently connected to and an application such as ArcMap. For example, when a user connects to an ArcSDE service and adds three feature classes to ArcMap, three ArcSDE streams are created. For further information on streams and the parameters available to control their operation, see 'Streams' later in this chapter.

When an ArcSDE stream is created, the ArcSDE service allocates transport buffers on both the client and the server. Transport buffers reduce I/O and improve performance by accumulating records and sending them across the network in batches rather than as individual records.



Client/Server transport buffers

The records are collected in the ArcSDE server's transport buffer and sent to the ArcSDE client transport buffer when the application is querying the database. Alternatively, the records are collected in the client's transport buffer and sent to the server's transport buffer when the application is writing data to the database.

Three parameters in the giomgr.defs file control the transport buffers.

`MAXBUFSIZE` The total amount of memory allocated to each transport buffer

`MINBUFSIZE` The minimum threshold of each transport buffer

`MINBUFOBJECTS` The minimum number of records per transport buffer

The MAXBUFSIZE represents the total amount of memory that is allocated to each transport buffer. The transport buffer stops accumulating records once MAXBUFSIZE is reached and waits for the request to send the records to the other buffer.

The MINBUFSIZE and MINBUFOBJECTS parameters are lower thresholds that prevent the records from being sent until one of them is attained. For example, if the application requests data, the request is deferred until the server transport buffer reaches either MINBUFSIZE or MINBUFOBJECTS.

When creating a geodatabase, you should raise the parameters to increase the transmission speed of data loading. Once loading is complete, reduce the transport buffers.

Before raising the MAXBUFSIZE parameter too high, consider the maximum overall impact to the server's memory budget. The default MAXBUFSIZE adds 64 kilobytes per stream per ArcSDE user connection. For example, if there are 100 users using an application that displays seven feature classes, ArcSDE allocates a total of 44,800 kilobytes (100 * 7 * 64) of memory for the server's transport buffers. Doubling the MAXBUFSIZE in this example

would result in 89,600 kilobytes of memory allocated to the server's transport buffers. Excessive paging may result on the server if MAXBUFSIZE is set too high and physical memory is not available to satisfy it.

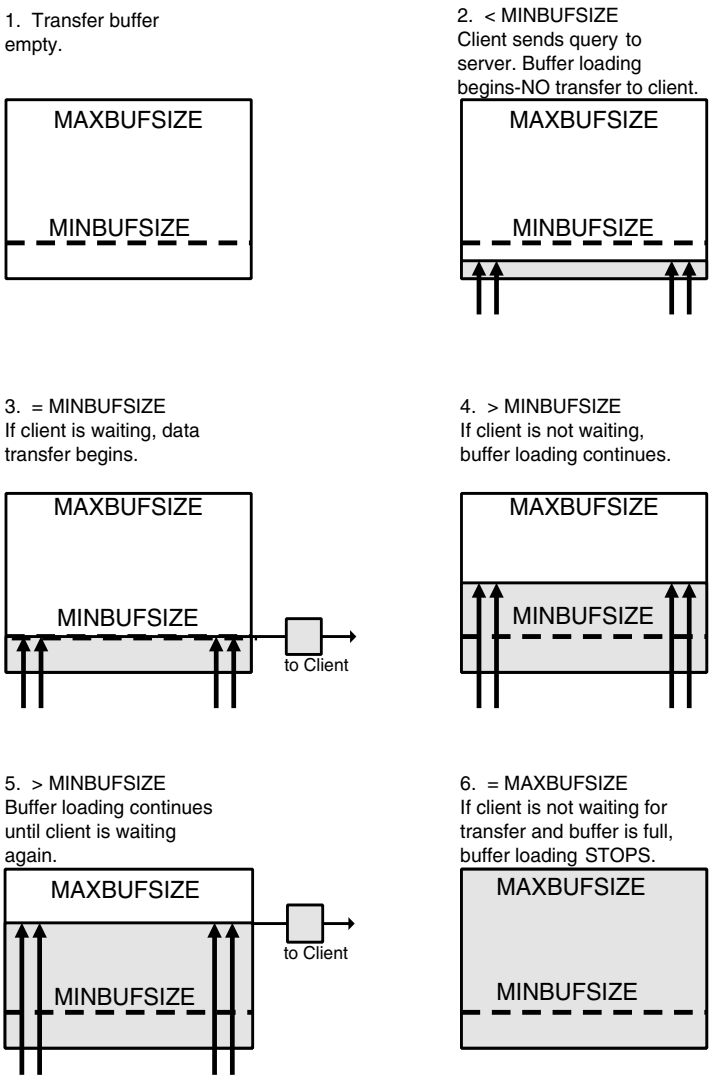
Set MINBUFSIZE to no more than one-half of the MAXBUFSIZE.

Setting MINBUFSIZE too high increases wait time. If MAXBUFSIZE is 64K and MINBUFSIZE is 56K, the client will wait until the 56K threshold is reached before sending the transport buffer.

Reducing the MINBUFSIZE parameter improves the cooperative processing between client and server.

MINBUFOBJECTS depends on the size (bytes) of a row of data. MINBUFOBJECTS is examined first and then MINBUFSIZE. If the threshold of either parameter is breached, the contents of the transport buffer are sent.

ArcSDE application developers can override the giomgr.defs transport buffer parameters with the C API function SE_connection_set_stream_spec.



The data transfer process from an ArcSDE server transfer buffer to an ArcSDE client

Array buffer parameters

For each ArcSDE stream that is created, ArcSDE allocates an array buffer.

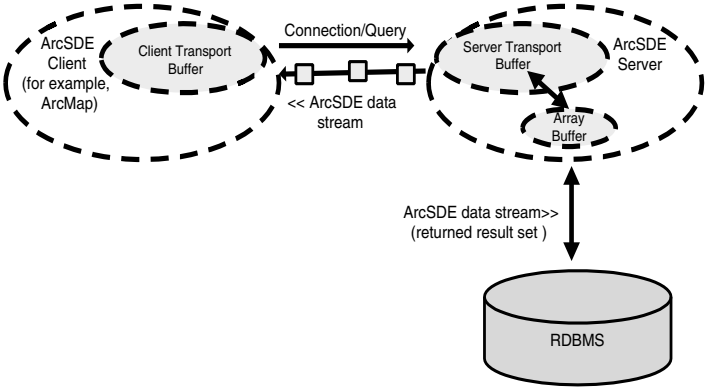
Whenever possible, groups of records are transferred between ArcSDE and the DBMS server. For DBMSs that have implemented array inserts, the ArcSDE server inserts records into the DBMS server array. All supported DBMSs, except Microsoft Access, have implemented array fetch mechanisms, so whenever ArcSDE issues a select statement on behalf of the user, it fetches or gets the records in an array.

Array buffers reduce the amount of I/O occurring between ArcSDE and the DBMS server by fetching and inserting data in larger chunks. Less I/O results in better performance. However, excessive paging can result if the array buffers are set larger than necessary. As a general rule, approximately 100 records is the optimum number of records to array process for most applications.

The array buffer parameters determine the size of the array buffers. Parameters that affect query performance are SHAPEPTSBUFSIZE, ATTRBUFSIZE, MAXARRAYSIZE, and MAXARRAYBYTES.

```
MAXARRAYSIZE        100      #Max.array fetch size
MAXARRAYBYTES    550000      #Max.array bytes
                              allocated per stream
SHAPEPTSBUFSIZE 400000       #Shape POINTS array
                              buffer size
ATTRBUFSIZE      50000       #Attribute array
buffer                                size
```

The array buffer parameters define the number of records transferred together for insert and query operations. During an array insert, ArcSDE fills the array buffer with records from the server transport buffer and transfers the entire array to the DBMS. ArcSDE queries data in an array by using array buffers to receive the data.



ArcSDE server array buffers

Estimating SHAPEPTSBUFSIZE

Array inserts of feature geometry are divided into two parts: feature metadata and point data. Feature metadata, including the feature ID, number of points, and entity type, requires the same amount of space for each feature. The space required to store the point data, on the other hand, varies according to the number of points in the feature.

SHAPEPTSBUFSIZE determines the buffer size for the point data. The default setting is based on an array size of 100. Tuning SHAPEPTSBUFSIZE to the optimal setting is critical to performance. The ArcSDE server estimates the average size of all features based on the array size (MAXARRAYSIZE) and the size of the points buffer (SHAPEPTSBUFSIZE). If a feature exceeds the average size, it is flagged as truncated and fetched separately.

For example, suppose SHAPEPTSBUFSIZE is 400,000 (400K), and the array size per fetch (MAXARRAYSIZE) is 100 rows (or 100 feature classes). The features in this example do not have annotation, z-values, or measures, so each point requires eight bytes (four bytes for the x-value and four bytes for the y-value).

Dividing SHAPEPTSBUFSIZE by MAXARRAYSIZE returns the maximum space available for each feature's points within the array buffer.

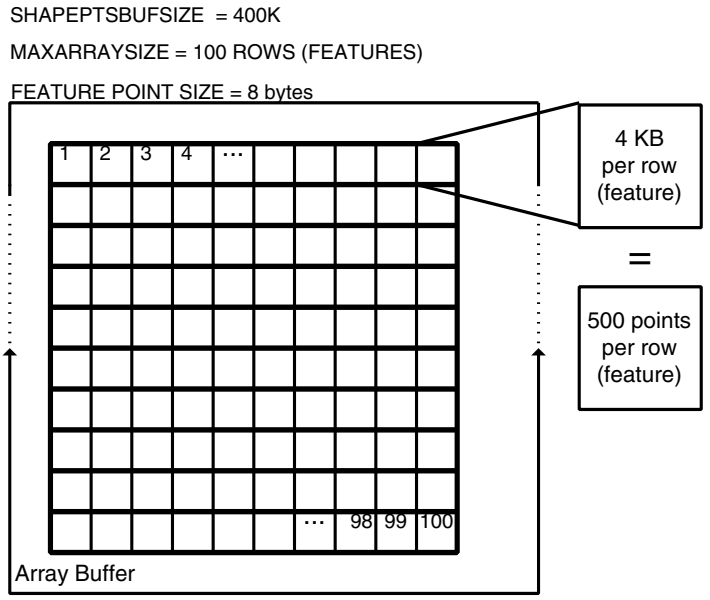
```
400000 / 100 = 4000
```

Further dividing the maximum space available to each feature by the number of bytes each point requires returns the total number of points that can be stored in the maximum space available to each feature.

```
4000 / 8 = 500
```

In this case, ArcSDE expects each feature to have no more than 500 points. When ArcSDE encounters features with 501 or more points, the feature is skipped and fetched separately, forcing an additional I/O fetch from the DBMS.

Setting the SHAPEPTSBUFSIZE parameter too small results in a higher number of features to be fetched individually, which diminishes performance. Setting this parameter too high wastes memory.



The SHAPEPTSBUFSIZE and MAXARRAYSIZE parameters determine the number of points that can be stored in the maximum space available for each feature within an ArcSDE array buffer.

Tip

Point size

The number of bytes required by each point depends on its coordinate type. If the feature class only has x,y coordinates, the number of bytes each point requires is 8. If either z-values or measures are present, 12 bytes are required. If both z-values and measures are present, 16 bytes are required.

Determining 90 percent of the most queried feature classes for an optimum SHAPEPTSBUFSIZE setting

1. Determine the total number of features in the feature class by issuing this SQL query.
2. Determine the largest number of points in the feature class to use as a starting point in the step.
3. Through iteration, determine the number of points 90 percent of the features have. Start by using an x-value that is 0.75 * maximum numofpts. Increase or decrease the x-value until the feature count is 90 percent of total features.
4. Multiply the number of points by the number of rows (MAXARRAYSIZE) the array will hold and the number of bytes required by each point.

   For instance, if MAXARRAYSIZE is set to 100, 90 percent of the features have 710 or fewer points, and each point requires 8 bytes, set the SHAPEPTSBUFSIZE to 568000.

1. `select count(*) 'total features' from f<n>;`
   `– non-object-relational model`

   `select count(*) 'total features' from table;`
   `– object relational model`

2. `select max(numofpts) 'maximum numofpts' from f<n>;`

   `select max(numpoints<spatial_column>)) 'maximum numofpts' from <table>;`

3. `select count(*) 'feature count' from f<n> where numofpts < X;`

   `select count(*) 'feature count' from table where numpoints(<spatial_column>) < X;`

4. `SHAPEPTSBUFSIZE = (MAXARRAYSIZE) * (number of points) * (point size)`

   `568000 = 100 * 710 * 8`

Estimating ATTRBUFSIZE

ATTRBUFSIZE defines the array buffer size for attribute (non-BLOB) data. Tuning the attribute array buffer is similar to the SHAPEPTSBUFSIZE. The default 50,000 setting allows 100 rows with 500 bytes of attribute data each.

Performance is affected when the number of rows that can be fetched into the attribute buffer does not match the MAXARRAYSIZE parameter setting. For queries involving multiple columns, add the number of bytes per column to get a total row size. If the ATTRBUFSIZE divided by row size is less than the MAXARRAYSIZE, the actual array size for the query (MAXARRAYSIZE) is less. This adjustment occurs because giomgr.defs parameters are not flexible. If the number of rows exceeds MAXARRAYSIZE, the array size remains the same.

Setting MAXARRAYSIZE

As mentioned, MAXARRAYSIZE sets the number of rows that the server will fetch per request. The recommended default value is 100. Optimal values can range between 20 and 150 on different platforms and DBMSs. Once the shape points data (SHAPEPTSBUFSIZE) and attribute buffer (ATTRBUFSIZE) are correctly tuned, try several array sizes to determine the optimal setting for each installation.

Setting MAXARRAYBYTES

Control the maximum number of bytes per stream with the MAXARRAYBYTES parameter. This value represents the total bytes that can be allocated to both ATTRBUFSIZE and SHAPEPTSBUFSIZE for each stream.

MAXARRAYBYTES is simply a way to manage the memory allocations for array buffers on the server. The sum of ATTRBUFSIZE and SHAPEPTSBUFSIZE must be less than or equal to MAXARRAYBYTES. If it isn't, the ArcSDE service will not start. If this problem occurs, either increase MAXARRAYBYTES or decrease either ATTRBUFSIZE or SHAPEPTSBUFSIZE.

This value cannot be changed with the SE_connection_set_stream_spec function and can only be altered in the giomgr.defs file by the ArcSDE administrator.

Managing BLOB data

The BLOB parameters, MAXBLOBSIZE and BLOBMEM, determine the server-side buffer requirements for BLOB data types. MAXBLOBSIZE determines the maximum number of bytes the server will accept. A BLOB is either written to memory or disk, depending on the size. BLOBMEM determines the maximum BLOB size for in-memory storage. That is, the server allocates BLOBMEM bytes to hold the BLOB. A BLOB that exceeds this size is written to disk.

Registered tables

The REGISTRATIONS parameter controls the total number of registered tables. You must either register tables if they are accessed by an application that uses logging or if the feature class or table is versioned. The ArcSDE service allocates 504 bytes of shared memory for each entry. For instance, if REGISTRATIONS is set to 500, 252,000 bytes are allocated to the ArcSDE service.

Layers

The LAYERS parameter specifies the maximum number of feature classes the ArcSDE service supports. Set the value higher if users that create feature classes report an SE_TOO_MANY_LAYERS error. Avoid setting this parameter any higher than needed, as it allocates 12 bytes of shared memory per entry.

Maximum initial features (Oracle only)

The MAXINITIALFEATS parameter controls the initial features argument that may be submitted to either the sdelayer command or the SE_layer_create function. Based on the value of initial features, sdelayer or SE_layer_create calculates the initial and next extent of the feature and spatial index table. To prevent the possibility of a user inadvertently entering a very large initial features value and thus acquiring an initial extent beyond what is needed, the administrator can limit the initial features value by setting the MAXINITIALFEATS parameter.

Statistics

The MAXDISTINCT parameter controls the number of distinct values a call to either the SE_table_calculate_stats or the SE_stream_calculate_table_statistics function can return. Setting this parameter to 0 allows an unlimited number of distinct values to be returned by the applications that call these functions. However, the distinct values are generated in memory on the server and passed to the client's memory when the list is complete. Calculating the statistics of a very large table could pose a threat to the client and server resources. A prudent database administrator will set this value high enough to allow most queries to complete, but not so high as to expose the server or the client application to a memory shortage. Should a user receive the error message SE_TOO_MANY_DISTINCTS, the MAXDISTINCT parameter may be raised, but this should be done cautiously as it impacts both client and server memory. It may be advisable to examine the applications to determine whether queries could be performed more efficiently.

Locking

The giomgr.defs file contains three parameters that control database locking: LOCKS, MAXTABLELOCKS, and STATELOCKS. Each parameter has a default value of 10,000, which should be adequate for most applications. However, if users report an SE_OUT_OF_LOCKS error, or if that error appears in the sde.errlog file, increase the associated lock parameter.

The LOCKS parameter specifies the maximum number of area locks allowed by the ArcSDE service. If a large number of users are editing feature classes and they are receiving SE_OUT_OF_LOCKS errors, increase this parameter and restart the ArcSDE server. Each lock entry in the LOCKS parameter requires 12 bytes of shared memory, so setting this value higher has a low impact on the overall performance of the ArcSDE service.

The MAXTABLELOCKS parameter controls the maximum number of business table row locks the ArcSDE server allows. If users are updating large numbers of business table records and they receive an SE_OUT_OF_LOCKS error, increase the MAXTABLELOCKS parameter. As each table row lock entry only requires 8 bytes of memory, increasing this value has very little impact on the ArcSDE service.

The STATELOCKS parameter limits the total number of locks that may be applied to states in a versioned database. Each user that queries a versioned feature class or table is allocated a state lock. Users who edit a feature class or table receive one state lock per edit operation. After 30 states are generated, ArcSDE automatically trims the first 15 states and compresses them into

the 16th state. However, the state locks generated within one version remain if a new version is started. State locks can accumulate for a user if they create several new versions during an edit session.

It is normal for state locks to accumulate when versions are used as save points. If users report that they are receiving an SE_OUT_OF_LOCKS error when editing a versioned feature class or table, increase the STATELOCKS parameter. As each table lock entry requires 8 bytes of memory, setting this value higher has very little impact on overall performance.

Streams

You have read about Streams in the section 'Transport buffer parameters' earlier in this chapter. To recap briefly, each time a user accesses a feature class or business table, an ArcSDE stream is created to transfer the information between the client and the ArcSDE server. For example, when a user connects to an ArcSDE service and adds three feature classes to ArcMap, three ArcSDE streams are created. An ArcSDE stream is a mechanism that transports data between the database the ArcSDE service is connected to and an application such as ArcMap. The resources allocated to each stream are significant; it is therefore important that the ArcSDE administrator understands how they operate and how to control the impact they have on the resources of the ArcSDE host computer.

The administrator has two parameters to control the behavior of streams: MAXSTREAMS and STREAMPOOLSIZE.

MAXSTREAMS

The MAXSTREAMS parameter controls the maximum number of concurrently opened streams allowed on an ArcSDE service. If users report they have received the SE_TOO_MANY_STREAMS error, consider raising the MAXSTREAMS parameter to allow more streams to be created. However, if resources on the server, especially memory, are at a premium, first examine the applications to determine whether some of the feature classes displayed are necessary.

Each additional stream is allocated memory by the ArcSDE service for its transport and array buffers as well as for its other structures.

STREAMPOOLSIZE

The process of creating a stream requires the allocation of memory and other resources. These are released when users close the stream, which normally happens when they disconnect from the ArcSDE service.

If the STREAMPOOLSIZE is set to a value greater than 0, ArcSDE creates a pool of released stream resources for reuse. In this case, when a user releases a stream, the ArcSDE service first checks the stream pool to determine if it is full. If it is not, the released stream resources are added to the pool; otherwise the resources are deallocated. When the same user application creates a stream, the ArcSDE service checks the stream pool for an available stream. If one exists, ArcSDE removes it from the pool and allocates it to the user application.

The STREAMPOOLSIZE parameter determines how many stream resources the stream pool should hold. If an application is designed such that users tend to connect, browse, and disconnect, set the STREAMPOOLSIZE to be at least half the size of MAXSTREAMS. On the other hand, if an application is designed such that users connect, display, and query the same data continuously, the STREAMPOOLSIZE probably does not need to be any larger than 10 percent of MAXSTREAMS.

Raster parameters

ArcSDE stores images similar to the way it stores features, in a DBMS BLOB column. The ArcSDE system table RASTER_COLUMNS keeps track of all of the raster columns in the databases, just as the LAYERS table keeps track of the feature class columns. For more information about raster tables, see 'Appendix C: ArcSDE table definitions'.

RASTERCOLUMNS

The RASTERCOLUMNS parameter determines the maximum number of raster columns the ArcSDE service allows within the database. If users report an SE_TOO_MANY_RASTERCOLUMNS error, increase the RASTERCOLUMNS parameter to allow for the creation of additional raster columns. The ArcSDE server allocates eight bytes of shared memory for each RASTERCOLUMNS entry.

RASTERBUFSIZE

The RASTERBUFSIZE parameter controls raster data transfer, which operates in a fashion similar to streamed data (see 'Transport buffer parameters' and 'Array buffer parameters' in this chapter).

The RASTERBUFSIZE is specified in bytes and must be large enough to hold the largest raster tile. The default ArcInfo tile size is 64 * 64 pixels. Data that is eight bits per pixel has a tile size of 4,096 bytes (64 * 64).

The raster transfer includes both an array buffer and transport buffers. The raster array buffer is set at two times the RASTERBUFSIZE parameter, while the raster transport buffers are set to the RASTERBUFSIZE. Therefore, the memory allocated to raster transfer on the server is three times RASTERBUFSIZE. On the client, RASTERBUFSIZE bytes of memory are allocated to the client raster transport buffer. The raster buffers are allocated when raster tiles are accessed by a stream. The raster buffers are not deallocated until the stream is closed (unless the stream is added to the stream pool—see STREAMPOOLSIZE). In other words, a user can elect to display an image and a feature class together in ArcMap. The raster buffers are allocated on demand when a user selects an image stored in the feature class's business table.

If the ArcSDE service encounters a raster tile that does not fit into the transport buffer, the SE_RASTERBUFFER_TOO_SMALL error is returned. If users report this error, determine the tile size they are using. If they are using a 256 x 256 tile size on an 8-bit image, the RASTERBUFSIZE must be at least (256 * 256) 65,536 bytes. The resolution of the image must be either 16-, 32-, or 64-bit resolutions. A 64-bit resolution transferred with a tile size of 256 x 256 requires a RASTERBUFSIZE of (256 * 256 * 8) 524,288 bytes. If memory is at a premium, advise users to specify a smaller tile size rather than raise the RASTERBUFSIZE.

Maximum time difference

The maximum time difference that a client's system clock can deviate from a host's system clock can be set with the giomgr.defs MAXTIMEDIFF parameter. The parameter is specified in seconds. It prevents an unauthorized entry by individuals who may have captured a network packet with a sniffer software that contains an ArcSDE connection string. Because the encrypted password is time stamped, the packet cannot be resent at a later time if MAXTIMEDIFF is set low enough (for example, 60 seconds).

If legitimate connections receive a "-99 password received was sent 7 MAXTIMEDIFF seconds before" error, reset the client machine's system time to the host's system time. Disable MAXTIMEDIFF by setting it to -1.

Displaying the ArcSDE initialization parameters

You can list the current ArcSDE initialization parameters using the sdemon command with the "-o info -I config" options.

```
$ sdemon -o info -I config
ArcSDE I/O Manager Configuration Parameters at
Fri Apr 16 10:52:48 1999

------------------------------------

ArcSDE Version                    8.1
ArcSDE Server Build               for Oracle Build
945 Mon Jul 24 22:46:20 PDT 2000
Underlying DBMS                   Oracle
Max. Server Connections           64
Max. Number ArcSDE Layer Locks    10000
Max. Number ArcSDE State Locks    10000
Max. Number ArcSDE Table Locks    10000
Max. Numver ArcSDE Object Locks   10000
Max. Layer Number                 1500
Max. Registration Number          3000
Root Path                         /abbey1/sdeexe81
Temp Path                         /tmp
Min. Transmission Buffer Size     16384 Bytes
Max. Transmission Buffer Size     65536 Bytes
Max. Transmission Buffer Count    512 Objects
Max. Initial Features             10000 Objects
Max. stream pool size             3 streams
Server Timeout                    16 Seconds
Max. BLOB Size                    1000000 Bytes
Max. in Memory BLOB Size          1024 Bytes
Max. Distincts                    512
Autocommit Frequency              5000
Max. Streams                      50
Shape Point Buffer Size           400000 Bytes
Attribute Buffer Size             50000 Bytes
Blob Buffer Size                  30000 Bytes
Max. Array Size                   100
Max. Array Bytes                  550000 Bytes
Max. Client/Server Time Diff.     Unlimited
State Caching                     On
Instance name                     esri_sde
Port Number                       5151/tcp
TCP/IP Keepalive on Connections   Off
Instance Type                     Read/Write

$
```

Managing ArcSDE services

4

IN THIS CHAPTER

- Before starting an ArcSDE service
- Starting a local ArcSDE service on Windows NT
- Starting a remote ArcSDE service on Windows NT
- Starting a local ArcSDE service on UNIX
- The sdemon command output
- Starting a remote ArcSDE service on UNIX
- Pausing, resuming, and shutting down an ArcSDE service
- Removing ArcSDE sessions (Windows NT and UNIX)

The administrator who manages the ArcSDE service can place the service in one of three states: running, paused, or shutdown. Although it is desirable to maintain the ArcSDE service in the most productive state (running), at times it is necessary to shut down the service, perhaps to perform essential maintenance.

The tools you use to manage an ArcSDE service depend on whether the service has been installed on a Windows NT or a UNIX system. On UNIX systems, the administrator uses the sdemon command. ArcSDE services installed on Windows NT are started and stopped from the service menu. In both cases, the sdemon command is used to pause and resume the service.

ArcSDE services installed on either Windows NT or UNIX can be managed from a remote computer using the sdemon command. However, only a remote UNIX system can manage an ArcSDE service installed on a UNIX system.

Before starting an ArcSDE service

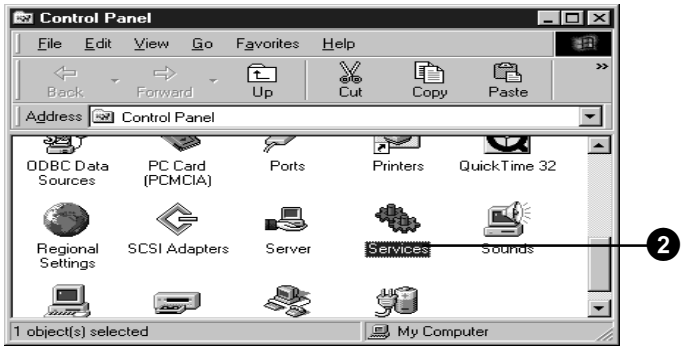
Before you can start the ArcSDE service, you must satisfy the following conditions:

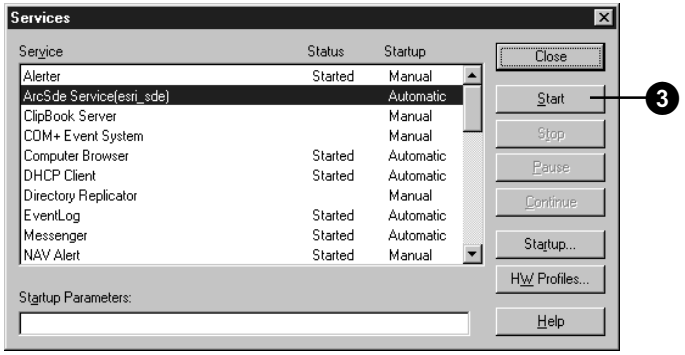
- The database management system (DBMS) instance must be started.
- The DBMS sde user account must exist.
- The system environment must be set such that the ArcSDE server can connect to the DBMS instance as the sde user.
- The ArcSDE home directory must exist.
- An ArcSdeServer license must be available from a license server accessible through the network.

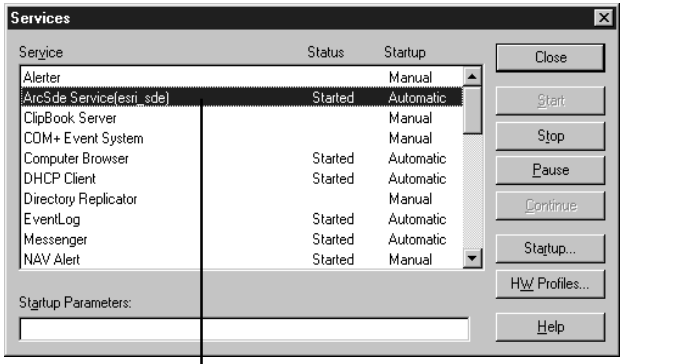
Starting a local ArcSDE service on Windows NT

You can start an ArcSDE service on Windows NT from the Services menu. The name of the service always begins with "ArcSDE service", and the ArcSDE service name itself is enclosed in parentheses—for example:
ArcSDE service(arcsde8).
If the service fails to start, make a note of the Windows NT error number and see the 'Common ArcSDE startup problems on Windows NT servers' in Chapter 6, 'Troubleshooting the ArcSDE service'.

1. Open the Control Panel by clicking Start, pointing to Settings, then checking Control Panel.
2. Double-click Services to open the Services menu.
3. Click the ArcSDE service name and click Start. The status changes to Started.







The status changes to Started, and the Start button is unavailable.

Starting a remote ArcSDE service on Windows NT

You can start a remote Windows NT ArcSDE service from another Windows NT machine.

The remote computer must be accessible over the network. The ping command can be used from an MS–DOS prompt to determine whether the remote ArcSDE service host can be reached.

Remote startup is initiated using the sdemon command from an MS–DOS command prompt. The addition of the -s <server> and -i <service> options identifies the remote host computer and the name of the remote ArcSDE service.

Tip

ArcSDE administrator Windows NT user group

The ArcSDE administrator must belong to the Windows NT administrator or power user group on the remote machine and have access via the system's environment variables to the sdemon command.

See Also

For more information on the various sdemon command options, see 'Appendix D: ArcSDE service command references'.

Using the ping command to verify a remote network connection

1. At the MS–DOS command prompt, type the command "ping" followed by the name or TCP/IP address of the remote computer.

```
C:\> ping bruno

Pinging bruno.esri.com [46.1.1.92] with 32
bytes of data:

Reply from 46.1.1.92: bytes=32 time<10ms
TTL=128
Reply from 46.1.1.92: bytes=32 time<10ms
TTL=128
Reply from 46.1.1.92: bytes=32 time<10ms
TTL=128
Reply from 46.1.1.92: bytes=32 time<10ms
TTL=128

C:\>
```

Starting a remote Windows NT ArcSDE service

1. To start the remote ArcSDE service, "arcsde_8", on host computer "Bruno", type the sdemon command and include the "-s bruno -i arcsde_8" options.

```
C:\>sdemon -o start -p my_password
   -s bruno -i  arcsde_8
```

Starting a local ArcSDE service on UNIX

The sdemon command manages ArcSDE services configured on UNIX systems.

Tip

Configuring the ArcSDE service on UNIX

Because the root account can run "sdemon -o start", you may configure the ArcSDE service to start automatically when booting up the UNIX system. For Windows NT systems, set the ArcSDE service to automatic.

Tip

Starting the ArcSDE service on UNIX

You must be logged in as either the owner of the ArcSDE service's home directory, $SDEHOME, or as the user "root" to start an ArcSDE service.

Tip

The sdemon command with -p option

You may enter the password as part of the sdemon command—for example, "$ sdemon -o start -p my_password"*—but the password will be displayed on the screen.*

1. Type the command sdemon with the "-o start" option to start the ArcSDE service.
2. You will be prompted to type in a password. This will not be displayed on the screen for system security.

❶ `$ sdemon -o start`

❷ `Please enter the ArcSDE DBA password:`

The sdemon command output

The output of the sdemon command during startup looks like this:

```
_____________________________________

ESRI ArcSDE I/O Manager - Release 8.1 - Mon Oct
2  13:59:40 PST 2000

_____________________________________
Instance initialized for SDE . . .
Connected to instance . . .

DBMS Connection established...
DBMS:            “Oracle”
Service Name:    “esri_sde”

ArcSDE Instance esri_sde started  Mon Oct 16
14:44:37 2000
```

Starting a remote ArcSDE service on UNIX

Before an ArcSDE service on a UNIX system can be started from a remote UNIX or a Windows NT machine, you must complete five configuration steps.

The dbinit.sde file must contain the database connection, the license manager variables, and the library path to the ArcSDE and DBMS dynamic libraries. By default, this file does not exist on the UNIX platform. You will need to create the file before you update it.

You must also add additional one-line entries to the /etc/services and the /etc/inetd.conf files and then reinitialize the inetd daemon.

After you have completed the five configuration steps, you can test the remote startup procedure from either a UNIX or a Windows NT computer using the sdemon command with the "-s" and "-i" options.

Tip

/etc/inetd.conf file entry

This must be a single-line entry with no carriage returns or new lines.

1. Create the $SDEHOME/etc/dbinit.sde file. This is an example of the variables in the dbinit.sde file.

```
set    ORACLE_HOME=/ultra1/oracle
set    ORACLE_SID=ora8
set    LD_LIBRARY_PATH=/usr/lib:/ultra1/
       oracle/lib:/ultra1/oraexe/sdeexe81/lib
set    ESRI_LICENSE_FILE=27005@ultra
unset  TWO_TASK
```

2. As the root user, duplicate the ArcSDE service name in the /etc/services file as a user datagram protocol (UDP) entry that uses the same port number.

```
# \etc\services

esri_sde     5151/tcp
esri_sde     5151/udp
```

3. Again as the root user, update the /etc/inetd.conf file. Add this line to the bottom of the file.

   This is an example of an /etc/inetd.conf file.

```
<ArcSDE instance> dgram udp wait <ArcSDE home
owner> <$SDEHOME>/bin/sderemote iomgr_inetd
<$SDEHOME>

# SDE remote start-up entries.

esri_sde dgram udp wait sde /ultra1/oraexe/
sdeexe80/bin/sderemote iomgr_inetd /ultra1/
oraexe/sdeexe80
```

4. As the root user, identify the relevant process using the UNIX command "ps -" piped through "grep". Reinitialize the inetd daemon by sending it a signal hang-up or SIGHUP.

   As the ArcSDE administrator, make sure the service is not started.

```
$ ps -u root | grep inetd
root  112  1 0  Aug 30 ? 0:08 /usr/sbin/inetd -s

$ kill -HUP 112


$ sdemon -o status

ArcSDE Instance esri_sde Status on ultra at
Thu Sep 30 11:32:56 2000
_____________________________________

ArcSDE instance esri_sde is not available on
ultra.
```

5. From either a UNIX or Windows NT computer, type the sdemon command with the start, server, and service name options to remotely start an ArcSDE service.

```
$ sdemon -o start -p my_password  -s ultra -i
esri_sde

ArcSDE Instance esri_sde started Thu Jan 15
20:31:28 2000
```

Pausing, resuming, and shutting down an ArcSDE service

The ArcSDE service has three modes—running, paused, and shutdown. While running mode is the productive state of the ArcSDE service, there will be times when you need to pause the service or shut it down—for example, to do routine maintenance work.

When the ArcSDE service is running, client applications can login and access the database through the ArcSDE service.

When the service is paused, current application connections continue, but additional application requests to connect are refused. This allows current users to complete work before the ArcSDE service is shut down. You can return a paused ArcSDE service to running mode by executing the sdemon with the resume option.

When the ArcSDE server is shut down, the sdemon command notes any ArcSDE processes still running and prompts to confirm that these tasks should be terminated before continuing to shut down. Any user who knows the ArcSDE DBMS user's password can shut down the ▸

Pausing the ArcSDE service (Windows NT and UNIX)

1. To pause an ArcSDE service, type the sdemon command and specify the pause option.

```
❶ $ sdemon -o pause -p my_password
  ArcSDE I/O Manager is paused, no further
  connections will be allowed
```

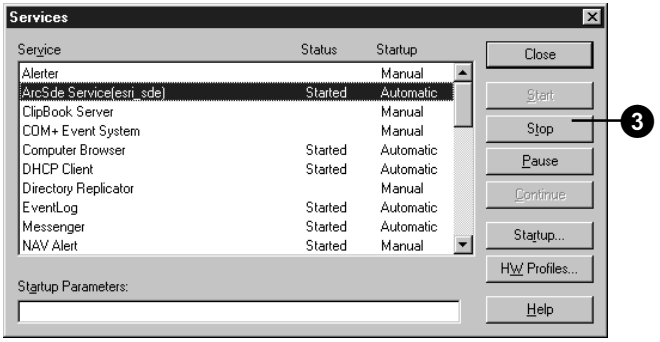
Resuming operation (Windows NT and UNIX)

1. To resume a paused ArcSDE service, type the sdemon command and specify the resume option.

```
❶ $ sdemon -o resume -p my_password
  ArcSDE I/O Manager is Resuming, new
  connections will now be allowed
```

Shutting down a local Windows NT ArcSDE service

1. Click the Start menu, click Settings, and click Control Panel.
2. Click on Services and scroll through the list of Windows NT services to find the ArcSDE service to shut down.
3. Click Stop to shut down the service.



ArcSDE service. Shutting down the ArcSDE service relinquishes all ArcSDE service processes and operating system resources.

If the giomgr process stalls on a Windows NT machine and it cannot be stopped using the methods discussed, it may be necessary to terminate the process with the "killp" executable file found under %SDEHOME%\tools.

If the ArcSDE service or giomgr process stalls on a UNIX platform and it cannot be stopped, it may be necessary to terminate the process with the UNIX "kill" command.

Tip

User account permissions

Windows NT users must have power user or administrator group permissions to pause, resume, or shut down a local or remote ArcSDE service.

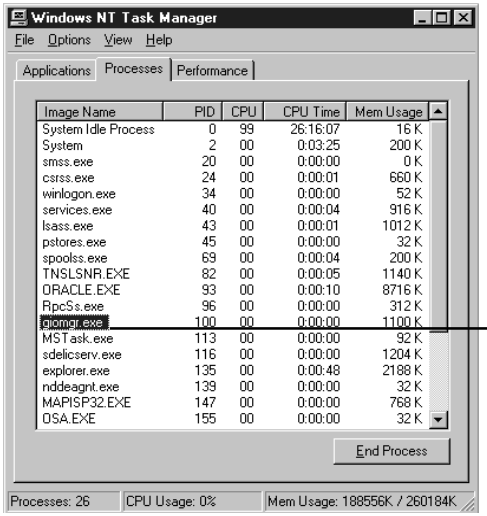
Tip

Remotely pausing and resuming an ArcSDE service (UNIX and Windows NT)

You can use the sdemon command to remotely pause, resume, and shut down an ArcSDE service by including the remote ArcSDE server and service name with the command options—for example, "$ sdemon -o pause -p my_password -s ultra -i esri_sde".

Shutting down a stalled giomgr process on Windows NT

1. Right-click the Windows NT taskbar and click Task Manager. Identify the process ID (PID) of the stalled giomgr process.
2. Type "killp" at the MS–DOS command line. Include the PID obtained from the Task Manager window.



2
```
C:\> cd %SDEHOME%\tools
C:\%SDEHOME%\tools> killp 100
Do you really want to kill process with pid
100(y/n)
y
```

Shutting down a local UNIX ArcSDE service

1. Type the sdemon command with the shutdown option.
2. Type the sdemon command with the status option to confirm the ArcSDE service has been shut down.

1
```
$ sdemon -o shutdown -p my_password
ArcSDE I/O Manager is Shutdown
```

2
```
$ sdemon -o status
ArcSDE I/O Manager is not available
```

Shutting down a stalled giomgr process on UNIX

1. Identify the PID of the stalled ArcSDE service or giomgr process using the UNIX “ps -ef” command that was piped through “grep” to isolate the “giomgr” process.
2. Type the UNIX “kill” command. Include the PID to terminate the giomgr process.

❶
```
$ ps -ef | grep giomgr
sde804  3403    1  0 06:00:03 ?      0:03
/luke1/sdeexe81/bin/giomgr /luke1/sdeexe81
```

❷
```
$ kill -9 3403
```

Removing ArcSDE sessions (Windows NT and UNIX)

To terminate an ArcSDE user process, list all ArcSDE user processes with sdemon, locate the process identifier, and use the sdemon command with the kill option to remove the process.

Before terminating a user process, be aware that the sdemon command disconnects user processes immediately. If the termination occurs before users issue a commit on a large transaction, the changes are rolled back. This operation should be used for emergencies only such as terminating a user process that is no longer responding to the application or when a user abnormally aborts an operation while it was processing a request and the process has hung.

Removing a single ArcSDE user session

1. Locate the PID for the ArcSDE session to be terminated with the sdemon command.
2. Issue the sdemon command with the "-o kill" option and the PID.
3. Verify that the process has been terminated.

❶
```
$ sdemon -o info -I users  (list the user processes)
ArcSDE Instance esri_sde Registered Server Tasks  on luke at Thu Sep 16 11:23:04 2000
_____________________________________________

PID   User  Host:OS         Started
_____________________________________________

90    bob buru:Win32:XDR  Thu Sep 16 09:29:52
10627 bob zanzi:Win32:XDR Thu Sep 16 11:12:31
```

❷
```
$ sdemon -o kill -t 10627
Please enter ArcSDE DBA password:
ArcSDE Instance esri_sde Process Management on luke at Thu Sep 16 11:23:46 2000
_____________________________________________

Kill Server Task 10627?   ARE YOU SURE (Y/N)?:
y
```

❸
```
$ sdemon -o info -I users  (list the user processes)
ArcSDE Instance esri_sde Registered Server Tasks  on luke at Thu Sep 16 11:25:04 2000
_____________________________________________

PID User  Host:OS        Started
_____________________________________________

90  bob  buru:Win32:XDR Thu Sep 16 09:29:52
$
```

Tip

The sdemon command with the -t option

The -t option specifies the PID of the process to be terminated. It can also be used to terminate all current user processes by typing the "all" keyword in place of a PID.

Tip

Stalled ArcSDE user sessions (UNIX)

To remove stalled ArcSDE user connections or gsrvr processes that cannot be terminated with the "sdemon -o kill" command, you must use the UNIX "kill" command.

Tip

Stalled ArcSDE user sessions (Windows NT)

The Task Manager cannot be used to terminate gsrvr processes for Windows NT since the process was started from the admin account. Instead, you must use the killp command under %ARCHOME%\tools.

Removing multiple ArcSDE user sessions

1. Type the sdemon -o command with the kill and -t all options.
2. Verify that all user processes have been terminated.

❶
```
$ sdemon -o kill -t  all
```

❷
```
$ sdemon -o info -I users   (list the user processes)
ArcSDE Instance esri_sde Registered Server
Tasks  on luke at Thu Sep 16 11:25:04 2000
_____________________________________________

PID User  Host:OS        Started
_____________________________________________

There are no ArcSDE users logged in.
$
```

Monitoring ArcSDE services

5

IN THIS CHAPTER

- **Displaying ArcSDE service status and lock table information**
- **How ArcInfo uses ArcSDE locking**
- **Displaying ArcSDE service statistics**
- **Displaying ArcSDE user session information**

To display the status of the ArcSDE service, use the sdemon command. This command provides a variety of administrative information, including:

- The current mode of the ArcSDE service
- The number of clients using the service
- Information about each client/server connection
- Current ArcSDE service configurations

Displaying ArcSDE service status and lock table information

You can check the status of an ArcSDE service by using the sdemon command with the status option. You will see the current status of the service—the current connection mode and the number of active service processes—reported on-screen.

```
$ sdemon -o status

ArcSDE Instance as8 Status on luke at Thu Sep 16
10:18:15 2000
-----------------------------------------------

Server Connection Mode:  Accepting Connections

Active Server Processes:  57

$
```

The ArcSDE locking mechanisms manage concurrent user access and guarantee read consistency when a user queries the database.

You should configure the ArcSDE service to have enough locks for the applications. Use "sdemon -o info -I locks" to keep track of the number of locks. If you find that your users' applications are approaching a lock limit set by giomgr.defs parameters, you must increase the LOCKS parameter and restart the ArcSDE service. For more information on the LOCKS parameter, refer to 'Appendix E: ArcSDE initialization parameters'.

```
$ sdemon -o info -I locks

ArcSDE Instance as8 Lock Table Information on
luke at Thu Sep 16 10:13:48 2000
-----------------------------------------------

1 PID:1017,Map Layer: 3,Lock Type: Shared Area

2 PID:1017,Map Layer: 2,Lock Type: Update Layer

-----------------------------------------------

2 of 10000 ArcSDE Layer Lock(s) currently in use.

1  PID: 1017, Object Id:  1, Object Type: 1
Application: [ArcSDE Internal]  Lock Type: Shared
Object
2  PID: 1017, Object Id:  3, Object Type: 2
Application: [User]  Lock Type: Exclusive Object
3  PID: 1017, Object Id:  2, Object Type: 1
Application: [User]  Lock Type: Shared Object
-----------------------------------------------

3 of 10000 ArcSDE Object Lock(s) currently in
use.

1   PID: 28129, State: 25685, Lock Type: Auto
Exclusive State
2   PID: 28129, State: 25684, Lock Type: Auto
Exclusive State
3   PID: 28129, State: 25683, Lock Type: Auto
Exclusive State
4   PID: 28129, State: 25682, Lock Type: Auto
Exclusive State
5   PID: 28129, State: 25681, Lock Type: Auto
Exclusive State
6   PID: 28129, State: 25680, Lock Type: Auto
Exclusive State
7   PID: 28129, State: 25679, Lock Type: Auto
Exclusive State
8   PID: 28129, State: 25685, Lock Type: Shared
State
9   PID: 28129, State: 25684, Lock Type: Shared
State
10  PID: 28129, State: 25683, Lock Type: Shared
State
11  PID: 28129, State: 25682, Lock Type: Shared
State
12  PID:   90, State: 25670, Lock Type: Auto
Exclusive State
13  PID:   90, State: 25669, Lock Type: Auto
Exclusive State
14  PID:   90, State: 25668, Lock Type: Auto
Exclusive State
15  PID:   90, State: 25667, Lock Type: Auto
Exclusive State
```

```
16  PID:    90, State: 25666, Lock Type: Auto
Exclusive State
17  PID:    90, State: 25665, Lock Type: Auto
Exclusive State
18  PID:    90, State: 25664, Lock Type: Auto
Exclusive State
19  PID:    90, State: 25670, Lock Type: Shared
State
20  PID:    90, State: 25669, Lock Type: Shared
State
21  PID:    90, State: 25668, Lock Type: Shared
State
22  PID:    90, State: 25667, Lock Type: Shared
State
23  PID:    90, State: 25666, Lock Type: Shared
State
24  PID:    90, State: 25665, Lock Type: Shared
State
25  PID:    90, State: 25664, Lock Type: Shared
State
-------------------------------------------------

25 of  10000 ArcSDE State Lock(s) currently in
use.

1  PID: 90, table: 162, Lock Type: Shared Table
2  PID: 26732, table: 122, Lock Type: Shared
Table
3  PID: 28129, table: 162, Lock Type: Shared
Table

-------------------------------------------------

3 of  10000 ArcSDE Table Lock(s) currently in
use.

$
```

The following lock table data appears for each lock in use:

`PID`	`Identifier of the process that owns the lock`
`MAP LAYER`	`Map layer number to which the lock applies`
`LOCK TYPE`	`The type of lock (update/ shared,layer/area, auto)`

If the lock type is an area lock or an automatic lock on a feature, the locked area also appears.

ArcSDE provides applications with four kinds of locks:

Object locks—used for versioning and geodatabase activities

Table locks—used to lock the rows of a table

Area locks—used to lock a spatial extent of a feature class

State locks—used to lock a versioned state of a feature class or table

How ArcInfo uses ArcSDE locking

Whenever an ArcInfo application connects to an ArcSDE service, it obtains the default state of the version it is connecting to and requests a shared lock on the state. It acquires the shared lock on the state if no other session currently holds an exclusive lock on all states. Sessions release their shared state locks when they disconnect.

When an edit occurs, the session acquires a new shared state lock for each new state created.

A session that attempts to perform a compress operation must first acquire an exclusive lock on all states. If it cannot acquire this lock it will fail immediately. To acquire this lock no other sessions can be connected.

When an ArcInfo application starts to edit a version, a shared object lock is acquired. The object lock is released when editing is stopped.

When an ArcInfo session references an object class, it acquires a shared table lock on all tables of that object class. For instance, if the session references a feature class, then a table lock is acquired on the business table, the feature table, and the spatial index table.

Before an ArcInfo session can change the properties of the object class or the schema of any of the tables of the object class, an exclusive table lock must be acquired. The exclusive lock cannot be acquired if other sessions have acquired shared locks on the tables.

When an ArcInfo session reconciles a version, the shared object locks are promoted to exclusive locks. The promotion and therefore the reconciliation will fail if other sessions are editing the object.

ArcInfo does not acquire layer locks, row locks, or area locks.

Displaying ArcSDE service statistics

You can use the sdemon command with the stats option to display statistical information about each current ArcSDE service using the "-o info -I stats" option:

```
$ sdemon -o info -I stats
ArcSDE Instance as8 Server Process Statistics
on luke at Thu Sep 16 10:37:14 2000

_____________________________________________
PID    OPS    READS  WRITES BUFFERS PARTIAL
_____________________________________________
90     17831  2557   958    1815      0
26732  3941   5386   3321   355       0
28129  17890  3964   969    1834      0
6796   363    59     0      37        0
_____________________________________________
F/BUF BUF AVG  TOT Kbytes
_____________________________________________
1     14K     26732K
24    79K     28129K
2     3K       6796K
1     21K       792K

$
```

The output includes:

PID	Process identifier
OPS	Number of client/server operations
READS	Number of features/identifiers read from disk
WRITES	Number of features written to disk
BUFFERS	Total number of buffers sent to client task
PARTIAL	Number of features sent to client that were larger than the buffer size
F/BUF	Average number of features/identifiers per buffer
BUF AVG	Average buffer size in bytes
TOT Kbytes	Total kilobytes of data sent to client

If no processes are connected to the ArcSDE service, a message to that effect will appear.

```
$ sdemon -o info -I stats
ArcSDE Instance as8 Server Process Statistics on
luke at Thu Sep 16 10:37:14 2000

_____________________________________________

There are no ArcSDE users logged in.
$
```

Displaying ArcSDE user session information

You can list all the server process information relating to current user connections using sdemon with the "-o info -I users" option.

```
$ sdemon -o info -I users

ArcSDE Instance esri_sde Registered Server Tasks
on luke at Thu Sep 16 11:10:51 2000

_____________________________________________

PID  User  Host:OS         Started

_____________________________________________

90   bob    buru:Win32:XDR  Thu Sep 16 09:29:52

9526 bob    zanzi:Win32:XDR Thu Sep 16 11:02:46

10406 vtest kayak:Win32:XDR Thu Sep 16 11:10:21

$
```

The following process information appears for each registered ArcSDE client:

```
PID     Process identifier

USER    User name of client

HOST:OS Name of the host computer and operating
        system

STARTED Name and time the process started
```

When a client terminates its ArcSDE connection, all process statistics are written to the SDEHOME/etc/giomgr.log file. The output from that file would look similar to the following:

```
Thu Sep 16 10:24:26 2000 - ArcSDE Server Pid 5429
Stopped, User: sdetest.

Thu Sep 16 10:23:30 2000 - ArcSDE Server Pid 5429
Registered, User: sdetest.

Thu Sep 16 10:24:26 2000 - Process 5429, R/T
Calls 22, Features read 0, wrote 35

51, Locks 0, Buffers 5, Partial 0 , Buffered
Features 3551, Buffered Data 3320K

Thu Sep 16 10:24:26 2000 - ArcSDE Server 5429
exit'd with status 0

$
```

Troubleshooting the ArcSDE service 6

IN THIS CHAPTER

- **What happens when you start an ArcSDE service**
- **ArcSDE Windows NT license manager**
- **What happens when an ArcSDE application connects**
- **Common ArcSDE startup problems on UNIX servers**
- **Common ArcSDE startup problems on Windows NT servers**
- **The Windows NT Event Viewer**
- **Examining the ArcSDE error log files**
- **ArcSDE intercept**

Most problems associated with starting an ArcSDE service occur because of a problem with the system environment. Often, a critical step was missed during the installation or configuration of the software.

What happens when you start an ArcSDE service

This section describes the startup of an ArcSDE service, the problems that may occur, and their probable causes.

The ArcSDE service starts the giomgr process

The giomgr executable file must be accessible. On UNIX systems, make sure that $SDEHOME\bin is in the system path and $SDEHOME\lib is in the system library path. On Windows NT, %SDEHOME%\bin must be in the system path if the sdemon command is used to start the service. Although on Windows NT the service is normally started from the services menu, sometimes it is appropriate to use the sdemon command to debug a failed startup.

The giomgr reads the system environment variables from the dbinit.sde file

The SDEHOME\dbinit.sde file contains settings for system environment variables that override those set in the system environment for either UNIX or Windows NT systems. On UNIX systems, if the dbinit.sde file does not exist, the sdemon command displays a warning message during startup. For more information, see 'The dbinit.sde file format' in Chapter 3.

On UNIX systems (and on Windows NT systems if the service is started from an MS–DOS command prompt using the sdemon command), the contents of the dbinit.sde file can be displayed during startup by setting the SDEDBECHO system environment variable to TRUE.

Be sure the database managment system (DBMS) connection variables and the license manager variables are set correctly.

You may list the current ArcSDE environment variables by using the sdemon command with "-o info -I vars" options.

```
$ sdemon -o info -I vars

ArcSDE Instance esri_sde's environment variables
on nendrum at Mon Oct 18 10:42:48 2000

-----------------------------------------------
ARCHOME=\\install\daily\arcexe81
ARCHOME_USER=C:\Program Files\ESRI\ArcInfo
ARCINFOFONTNAME=Courier New
ARCINFOFONTSIZE=8
ATHOME=\\install\daily\arcexe81\arctools
COMPUTERNAME=NENDRUM
ComSpec=C:\WINNT\system32\cmd.exe
NUMBER_OF_PROCESSORS=1
OS=Windows_NT
Os2LibPath=C:\WINNT\system32\os2\dll;
Path=d:\oracle\bin;C:\Program
Files\Oracle\jre\1.1.7\bin;C:\WINNT\system32;C:\WINNT;C:\Program
Files\ESRI\ArcInfo\bin;D:\ESRI\ArcInfo\arcsde\bin;D:\ESRI\ArcInfo\arcsde\lib
PROCESSOR_ARCHITECTURE=x86
PROCESSOR_IDENTIFIER=x86 Family 6 Model 3
Stepping 4, GenuineIntel
PROCESSOR_LEVEL=6
PROCESSOR_REVISION=0304
SDEHOME=D:\ESRI\ArcInfo\arcsde
SystemDrive=C:
SystemRoot=C:\WINNT
USERPROFILE=C:\WINNT\Profiles\Default User
windir=C:\WINNT
SDENOEQUIV=true

$
```

The giomgr determines if an ArcsdeServer license is available

Be sure the license manager is running and that an ArcsdeServer license is available.

The giomgr reads the Transmission Control Protocol/Internet Protocol (TCP/IP) service name

On Windows NT platforms, the service name is read from the registry if the ArcSDE service started from the Services menu. However, when debugging a startup problem by using the "sdemon -o start" command, the service name is read from the services.sde file.

On UNIX platforms, the service name is always read from the $SDEHOME\etc\services.sde file.

The giomgr attaches to the TCP/IP port assigned to service name

The service name must be in the system's services file. On Windows NT, the ArcSDE installation program adds the service name to the system service file automatically. On UNIX platforms, the services file must be updated manually.

On Windows NT, the system service file is located at <drive>:\winnt\system32\drivers\etc\services, while on UNIX systems the file is located at /etc/services.

The giomgr connects to the DBMS using connection information from dbinit.sde and operating system environment variables

Make sure the DBMS is up and running and that the sde user can connect to the database. The DBMS connection parameters set in the dbinit.sde file have precedence over any parameters that are set in the system environment.

If the dbinit.sde file does not exist, a warning message is sent to standard output; however, this will not prevent the ArcSDE service from starting since the DBMS connection parameters of the system environment are used.

For the ArcSDE for Oracle8, Informix, and DB2 service, the giomgr locks the version table

The service locks the VERSION table if the READONLY parameter in the giomgr.defs file is set to FALSE. If another service has already locked the VERSION table, the second giomgr process will fail to start. If this happens, you must choose between shutting down the first service or setting the READONLY parameter to TRUE for the second service.

For the ArcSDE for Oracle8*i* and SQL Server service, the giomgr flushes the lock tables

The giomgr process flushes the locks of any processes orphaned by a chaotic shutdown of the service. The locks of direct connections are not flushed when the service starts.

The giomgr listens for connections on its TCP/IP port

At this point, the ArcSDE service has started. Applications can now connect and use the service.

ArcSDE Windows NT license manager

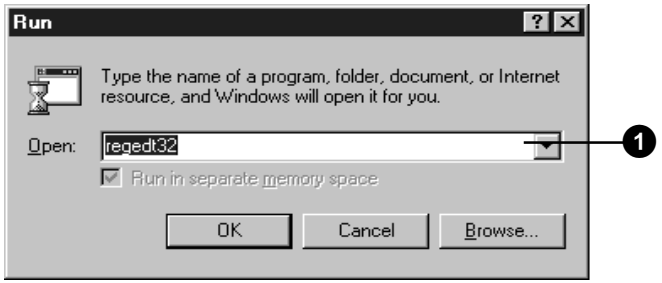
On the Windows NT platform, the ArcSDE administrator can determine which license manager the ArcSDE service is using by examining the LICENSE_SERVER registry parameter.

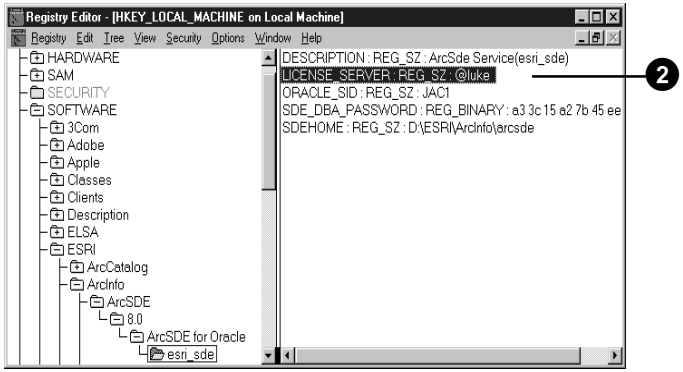
Once you have identified which license manager is being used, you can check its current status by logging on to the computer that hosts the ArcSDE license manager and displaying the license manager status window.

The display must include an entry for the ArcsdeServer license. ArcsdeServer licenses are also listed. Make sure there is at least one license available.

Examining the Windows NT LICENSE_SERVER registry parameter

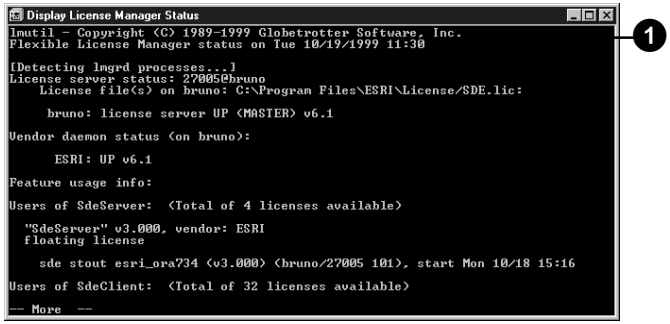
1. To invoke the registry editor, click the Start menu and click Run. Type regedt32 on the input line and click OK.
2. When the Registry Editor appears, click the HKEY_LOCAL_MACHINE window and expand SOFTWARE, ESRI, ArcInfo, ArcSDE, and 8.0. Expand your ArcSDE product folder and open the folder bearing the name of the ArcSDE service you are trying to start. Examine the value of the LICENSE_SERVER parameter—this will identify your license manager.





Checking the status of a Windows NT license manager

1. Log on to the computer that hosts the license manager. Click the Start menu, point to Programs, click ESRI, and click on License Manager.
2. Click on Display License Manager Status to bring up the status menu.



Listing ArcSDE license manager settings and the license manager status on UNIX

To determine which license manager the ArcSDE service is using, examine the contents of the dbinit.sde file and list the system environment settings.

On a UNIX system, you can set either the ESRI_LICENSE_FILE or LM_LICENSE_FILE variable in the dbinit.sde file to specify a license manager.

Either the ESRI_LICENSE_FILE variable or the LM_LICENSE_FILE variable may contain multiple entries separated by a colon. The giomgr process accesses the license managers as they are listed from left to right. The first license manager the giomgr process is able to access with an available ArcsdeServer license becomes the ArcSDE service license manager. Any other entries are ignored.

1. Use the "$printenv" command if you are using a C shell environment and the "$env" command if you are using a Bourne shell to list the current system environment. ►

```
$ env
HOME=/ultra1/sde
PATH=/ultra1/sdeexe81/bin:/ultra1/app/ora816/
product/8.1.6/bin:/bin:/usr/bin:/usr/ucb:/
etc:.         LOGNAME=sde
   SHELL=/bin/csh
PWD=/ultra1/sde
USER=sde
ORACLE_HOME=/ultra1/app/ora816/product/8.1.6
ORACLE_SID=ora816
LM_LICENSE_FILE=27005@luke ———————— 1
SDEHOME=/ultra1/sdeexe81
SDEINSTANCE=esri_sde
LD_LIBRARY_PATH=/ultra1/sdeexe81/lib:/ultra1/
app/ora816/product/8.1.6/lib:/usr/dt/lib:/usr/
lib
$
```

Tip

License manager variables

If the dbinit.sde file contains an LM_LICENSE_FILE variable that seems to be ignored, check the system environment to ensure that the ESRI_LICENSE_FILE variable is not set. An ESRI_LICENSE_FILE variable set in the system environment takes precedence over an LM_LICENSE_FILE variable set in the dbinit.sde file.

2. Once you have determined which license manager the ArcSDE service uses, examine the license manager itself.

   Log in to the host where the license manager is located and use “lmutil lmstat -A” to determine if the license server is running and if an ArcSDE Server license is present.

```
$ lmutil lmstat -A

lmutil-Copyright (C) 1989-1998 Globetrotter Software,
Inc.

Flexible License Manager status on Tue 10/19/2000
11:20

License server status: 27005@luke
   License file(s) on luke: /luke1/sde81shadow/
oraexe/sdeexe81/sysgen/license.dat:

    luke: license server UP (MASTER) v6.1

Vendor daemon status (on luke):

  ESRI: UP v6.1

Feature usage info:

Users of ArcSdeServer: (Total of 2 licenses ——— 2
available)

  “ArcSdeServer” v3.000, vendor: ESRI

  floating license

sde8 luke /dev/tty (v3.000) (luke/27005 5536), start
Tue 10/19 6:47


$
```

What happens when an ArcSDE application connects (three-tiered)

This section describes the sequence of events that take place when an ArcSDE client application connects to an ArcSDE service.

The giomgr process listens for connections on its TCP/IP port

The giomgr must be in a listening state before it can process a connection request. Make sure the ArcSDE service is started and listening.

On UNIX, use "sdemon -o status" to determine the state of the giomgr process.

On Windows NT, examine the state of the ArcSDE service from the services menu. Click Start, point to Settings, then click Control Panel. Double-click the Services icon to invoke the Services menu. Scroll down until you find the ArcSDE service. The ArcSDE service should have a status of 'STARTED' under the status field.

Applications submit connection requests to the ArcSDE service

The giomgr process responds to connection requests serially. Depending on the underlying DBMS, the giomgr process may require anywhere from 1 to 5 seconds to validate a connection request. It's possible that if many applications are trying to obtain an ArcSDE connection at the same time, some may exceed the standard TCP/IP 75-second time-out. This may be prevented from happening by setting the SDEATTEMPTS environment. SDEATTEMPTS specifies the number of times an application should retry the connection.

The giomgr compares the application computer's clock time with its host's clock time

If the application computer's clock time is more than MAXTIMEDIFF seconds from the ArcSDE server's clock time, the giomgr process does not allow the application to connect. MAXTIMEDIFF is set in the giomgr.defs file. For more information see Chapter 3,'Configuring ArcSDE services', and 'Appendix E: ArcSDE initialization parameters'.

The giomgr compares the application's client ArcSDE release with the ArcSDE service's release

If the application's release is greater than the ArcSDE service's release, the connection is refused.

ArcSDE applications are downward compatible. Applications developed with the ArcSDE 8 application programming interface (API) can connect to both ArcSDE 8 and SDE 3 services. Applications built with the SDE 3 API cannot connect to ArcSDE 8 services. Check the application's documentation for supported ArcSDE releases.

The giomgr process starts a gsrvr process that will serve the application

The giomgr process must be able to spawn a gsrvr process. If the maximum number of processes determined by current operating system restrictions has been reached, this operation will fail and no gsrvr process will be created.

The gsrvr process attaches to shared memory

Sufficient memory must be available on the ArcSDE service's host computer. Otherwise, the application connection will fail with a shared memory error. Should this happen, make more memory available to the gsrvr processes by reconfiguring either

the ArcSDE service or the DBMS server to use less memory. If possible, add more physical memory to the host computer.

The gsrvr process connects to the DBMS

The application must provide a valid user name, password, and database name (optional for some DBMSs) when it submits the connection request to the giomgr process. Invalid entries are rejected with a "-9 SE_INVALID_USER" error.

The gsrvr process opens the DBMS log file

ArcSDE applications often keep a log of table and feature class records that represent selected sets. For more information, see the sdelog command in 'Appendix D: ArcSDE service command references'. These logs are maintained in two tables, SDE_LOGFILES and SDE_LOGFILE_DATA, and are created when a user connects to the ArcSDE service for the first time. If the gsrvr process cannot create these tables, the connection will fail.

(ArcSDE for Informix prefixes the connected user's name to the log file tables.)

The giomgr process attaches the application to the gsrvr process

Once the giomgr process has attached the application to the gsrvr process, it resumes listening for new connections and performing other ArcSDE service management tasks. All application communication with the DBMS is conducted through the gsrvr process.

What happens when an ArcSDE application connects (two-tiered)

This section describes the sequence of events that occur when an ArcSDE client application connects directly to a DBMS server.

The DBMS server listens for local or remote connections

Each of the DBMSs supported by ArcSDE has its own method of accepting the connections of client applications. If you are not able to directly connect to the DBMS server, first make sure that this option is supported by your release of ArcSDE. At 8.1, only ArcSDE for Oracle8*i* and ArcSDE for SQL Server support direct connections. Also, make sure you are entering the connection information correctly if you are using an ESRI application such as ArcMap, ArcCatalog, or ArcView®. For information on how to perform a direct connection, consult the documention of these applications.

If you still cannot get connected, test the connection using the DBMS native SQL utility.

The ArcSDE client application detects the presence of the ArcSDEServer license

If the client application detects the presence of an ArcSDEServer license, it acquires a read/write connection to the database. If a valid ArcSDEServer license cannot be located, a READONLY connection is acquired.

The ArcSDE client application opens the DBMS log file

ArcSDE applications often keep a log of table and feature class records that represent selected sets. For more information, see the sdelog command in 'Appendix D: ArcSDE service command references'. These logs are maintained in two tables, SDE_LOGFILES and SDE_LOGFILE_DATA, and are created when the user connects to the DBMS for the first time. If these tables cannot be created, the connection will fail. The tables may have already been created if the user connected to the database through a gsrvr process.

Common ArcSDE startup problems on UNIX servers

The following section lists some of the ArcSDE service startup problems that are likely to be encountered in a UNIX environment.

System path variable issues

- If the PATH environment variable does not include the $SDEHOME/bin directory, the following error message is reported:

  ```
  sdemon: Command not found
  ```

- If the library path environment variable does not include the $SDEHOME/lib directory, the following error message is reported:

  ```
  ld.so.1: sdemon: fatal: libsde81.so: open failed: No such file or directory
  Killed
  ```

- If the library path environment does not include the necessary DBMS library directory, an error message similar to the following is reported:

  ```
  ld.so.1: /ultra1/ora8iexe/bin/giomgr: fatal: libclntsh.so.8.1: open failed: No such file or directory
  Killed
  Could not start ArcSDE – Check Network, $SDEHOME disk, DBMS settings and dbinit.sde.
  ```

  For more information on how to set the library environment variable for your ArcSDE product, see the *install_guide* under the SDEHOME documentation folder.

ArcSDE service already started

- If the I/O manager is already running, the following message appears:

  ```
  SDE Already Running
  ```

ArcSDE License Manager is not available/running

- If the license manager is not started, the following error message is reported:

```
ultra% sdemon -o start -p sde
-----------------------------------------------
ArcSDE 8.1 for Oracle8i  Tue Oct 31 22:41:36 PDT 2000
-----------------------------------------------
DBMS Connection established...
SDE servers do not appear to be licensed.
Error (-39) getting ArcSde server license.
 Could not start ArcSDE – Check Network, $SDEHOME disk, DBMS settings and dbini
t.sde.
ultra%
```

Temporary file permission problems

- If any ArcSDE temporary files exist and they are not owned by the ArcSDE administrator, the following error message is returned:

  ```
  ERROR: Cannot Initialize Shared Memory (-79)
  Delete /tmp/<service name> and /tmp<service name>.lock if present.
  Could not start ArcSDE – Check Network, $SDEHOME disk, DBMS settings and dbinit.sde.
  ```

  Delete the temporary files /tmp/<service name> and /tmp/<service name>.lock. For example, if the service name is esri_sde, you would delete the files /tmp/esri_sde and /tmp/esri_sde.lock. You may have to login as the root user to delete these files.

Problems relating to the DBMS

- If the DBMS is not started, you will receive an error message similar to the following:

```
init_DB DB_instance_open_as_dba: -51
DBMS error code: 1034
ORA-01034: ORACLE not available

Could not start ArcSDE – Check Network,
$SDEHOME disk, DBMS settings and dbinit.sde.
```

- If the sde DBMS user password is not correct, you will receive an error message similar to the following:

```
init_DB DB_instance_open_as_dba: -93
DBMS error code: 1017
ORA-01017: invalid username/password; login
denied

Could not start ArcSDE – Check Network,
$SDEHOME disk, DBMS settings and dbinit.sde.
```

- If the sde DBMS user does not exist, you will receive an error message similar to the following:

```
init_DB DB_instance_open_as_dba: -93
DBMS error code: 1017
ORA-01017: invalid username/password; login
denied

Could not start ArcSDE – Check Network,
$SDEHOME disk, DBMS settings and dbinit.sde.
```

Common ArcSDE startup problems on Windows NT servers

Normally the ArcSDE service is started as a Windows NT service from the Services control panel. If an error appears after clicking the Start button for the ArcSDE service, try to determine the nature of the problem. The error message contains an error number.

The error number generally relates to a specific type of error. Listed below are the error numbers you often encounter when starting an ArcSDE service and their likely cause.

1068 Dependency failure

The DBMS that the ArcSDE service is trying to connect to could not be found. The most likely causes of this problem are:

- The DBMS service is not started.
- The DBMS server has been removed.
- The DBMS connection information, entered when the ArcSDE server was created, is incorrect.

Make sure the DBMS service is started and independently confirm that this is not the source of the problem. If the error persists, use the sdeservice comand to delete the existing ArcSDE service and re-create it. For more information, see the discussion of the sdeservice command in ‘Appendix D: ArcSDE service command references’.

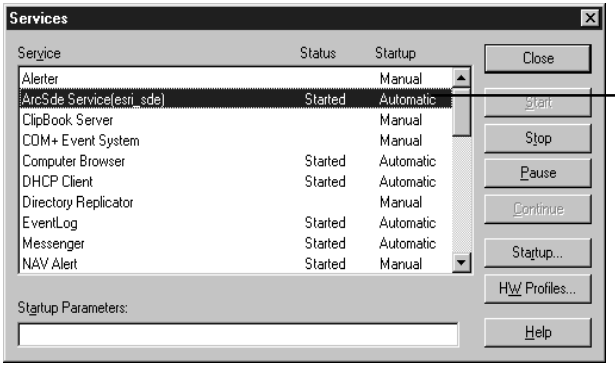
1069 Login failure

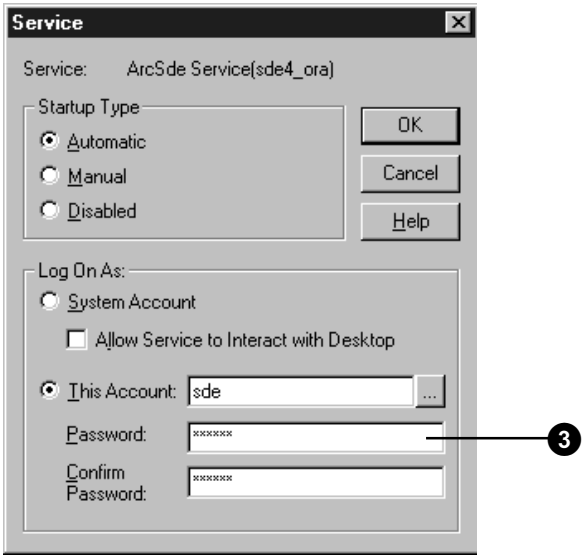
Generally, this error implies the Windows NT user who started the ArcSDE service is neither a Windows NT administrator nor a power user. An incorrect password is another possibility.

If the system administrator account is not being used to start the service, make sure the user account is a member of the administrator or power user group.

Verifying Windows NT user permissions

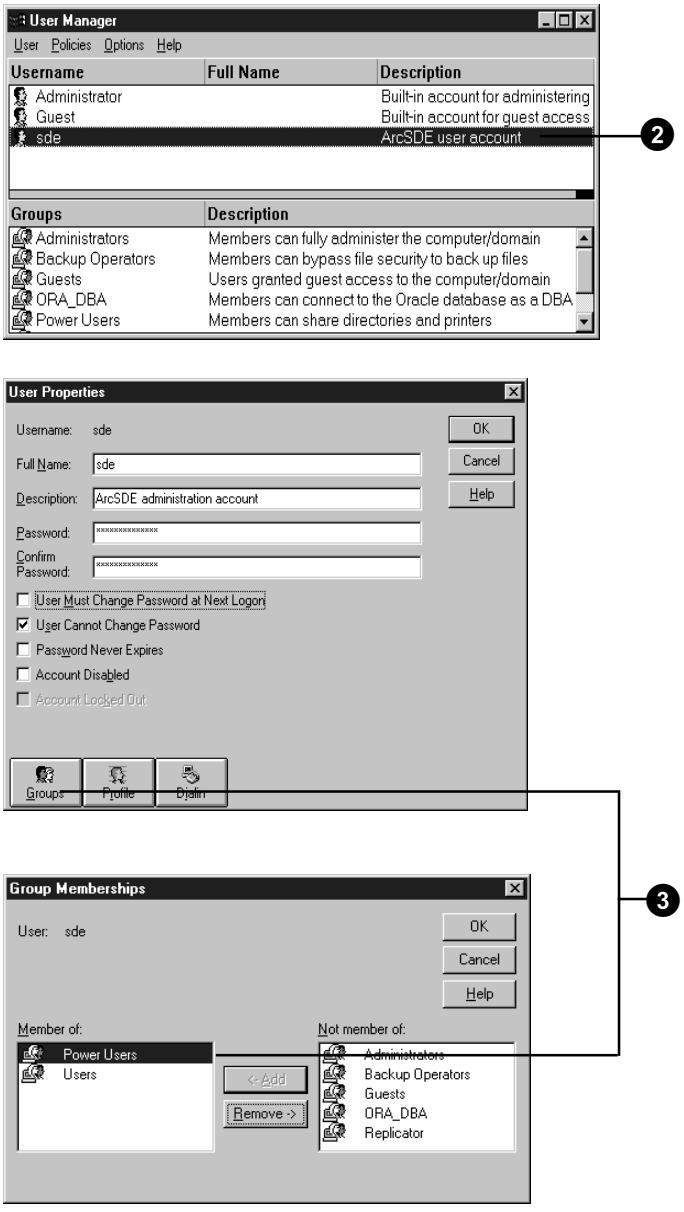
1. Click the Start menu, click Settings, click Control Panel, then double-click the Services icon.
2. On the Services menu, double-click the ArcSDE service entry to bring up the service menu.
3. Verify that the password is correct by reentering it.





Checking Windows NT user group permissions

1. Click the Start menu, click Programs, click AdministrativeTools, then click User Manager.
2. Double-click the user's name to bring up the User Properties menu.
3. Click the Groups button on the User Properties menu. Check to see if the user is in either the Administrators or Power Users group, as shown in the Member of list.



1072 Registry was busy

Something is happening in the registry regarding the ArcSDE service entry. Perhaps "sdeservice -o delete" was run or the service has been opened with the registry editor, regedt32. Alternatively, there may have been a problem with the OLE DB provider. Consult the installation guide for the correct version for the OLE DB provider.

1075 Service dependency deleted

The ArcSDE service is unable to locate the DBMS service that it will connect to.

Make sure the DBMS service exists and is started. If the problem persists, use the sdeservice command to delete and re-create the ArcSDE service. For more information, see 'Appendix D: ArcSDE service command references'.

2140 Internal Windows NT error

The ArcSDE service wasn't able to complete the startup process. Examine the %SDEHOME%\etc\sde.errlog file for possible clues as to why the ArcSDE service will not start.

Possible causes include:

- Can't connect to the DBMS server
- Can't create the ArcSDE data dictionary
- Can't locate an ArcSDE Server license in a license server

Possible solutions include:

- If the ArcSDE user's password was entered incorrectly, use sdeservice -o modify -r SDE_DBA_PASSWORD to correct it.
- If the DBMS connection information is incorrect, edit the %SDEHOME%\etc\dbinit.sde file.
- If an ArcSDE Server license cannot be located in a license server, make sure the license manager is running and its license file contains a valid ArcSDE Server license.

2186

This error is usually caused by a problem with the license manager.

Check the hardware key. It may not be seated correctly, or it may not be plugged into the host computer's parallel port. This error is also returned for general license manager problems. After checking the hardware key, refer to the tools provided with the license manager and to the license manager docmentation located in the SDEHOME\documentation directory.

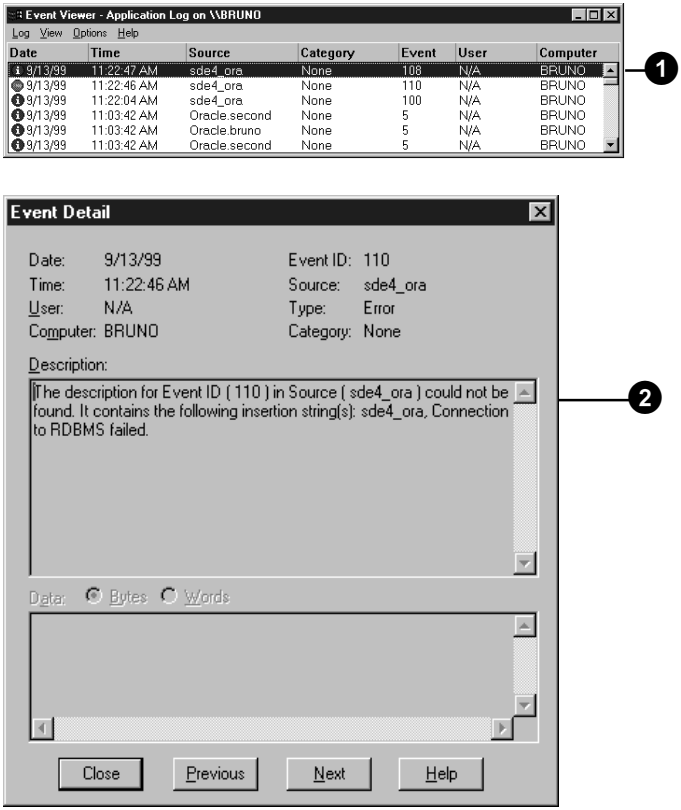
The Windows NT Event Viewer

The Windows NT Event Viewer provides diagnostic information that may also help explain ArcSDE startup problems.

Although the Event Viewer will often include a description of the problem, you can also check the %SDEHOME%\etc\sde.errlog file. This file will contain further information relating to the Windows NT startup problems.

Using the Event Viewer

1. Click the Start menu, click Programs, click Administrative Tools, and click the Event Viewer option. From the Event Viewer menu, click Application from the Log pulldown menu to list application events. Look for a red “stop” sign icon in the Date column and the corresponding name of the ArcSDE service in the Source column. Double-click the ArcSDE service entry to bring up the Event Detail menu.
2. The Event Detail menu includes a description of the problem. In this example, it is clear that a connection to the DBMS failed.



Examining the ArcSDE error log files

ArcSDE and all supported DBMSs track their activities by writing messages of events to log files. The log files may be examined to trace errors that have occurred. ArcSDE writes to two log files: giomgr.log file and sde.errlog file.

Viewing the giomgr.log file

The giomgr.log file is a text file that contains an entry for all giomgr process activity. Each time a user connects or attempts to connect to the ArcSDE service, a message is logged. When the user disconnects, another message is logged. The giomgr.log file also captures the startup and shutdown procedures of the ArcSDE service.

Viewing the sde.errlog file

Whenever a gsrvr process encounters a problem, the ArcSDE service records an entry in the sde.errlog. Sometimes the messages are warnings, while other times they point to ArcSDE service errors that should be addressed. When examining the sde.errlog file, keep in mind that the messages written to this file only occur on the server. Sometimes an ArcSDE application will report an ArcSDE-related problem, but this event will not appear in the sde.errlog.

The sde.errlog is truncated each time the ArcSDE service is started.

DBMS error log files

Each DBMS has its own way of logging errors. Consult the relevant DBMS administration guide to determine how your DBMS logs errors.

ArcSDE intercept

If you need to contact ESRI technical support, the analyst may ask you to intercept the ArcSDE client or server network broadcasts, depending on the nature of the problem. The ArcSDE intercept facility captures information that the client or server sends across the TCP/IP port to a file for examination.

If this information is required, set the relevant variables in the dbinit.sde file (see details below) and restart the ArcSDE service. The dbinit.sde file is located in the $SDEHOME/etc directory on UNIX systems and in the %SDEHOME%\etc directory on Windows NT systems.

To stop intercepting ArcSDE server network broadcasts, either comment out the variables by preceding the entry with the pound sign character "#" or delete them from the dbinit.sde file and restart the ArcSDE service.

To intercept ArcSDE client network broadcasts, set the variables in the client application user's system environment before connecting to the ArcSDE service. To stop intercepting ArcSDE client network broadcasts, disconnect the application from the ArcSDE service, unset the variables, and then reconnect to the ArcSDE service.

You may set the SDEINTERCEPT variable with the following flags to intercept network broadcasts:

c —intercept the API command name

r—intercept the Channel broadcasts read only

w—intercept the Channel broadcasts write only

t—intercept log time (minute:second)

T—intercept log time (hour:minute:second)

f—intercept flush immediate

For both client and server intercepts, set the SDEINTERCEPTLOC variable to the full pathname of the filename prefix that receives the information. Information is intercepted on a per-session basis. When intercept is enabled, a new file is created and written to each time an application connects to the ArcSDE service. The file is closed only after the application disconnects. ArcSDE generates a filename from the prefix provided in SDEINTERCEPTLOC by appending a numeric extension that begins at .001 and that increments sequentially for each new file created.

If the technical support analyst has asked for this intercept output from both the client and server, use distinct prefix names to distinguish between the client and server. For example, setting SDEINTERCEPTLOC to d:\tmp\sde_server in the dbinit.sde file captures server network broadcasts. Setting SDEINTERCEPTLOC to d:\tmp\sde_client in the applications environment captures client network broadcasts in the same directory but with a different prefix.

This is an example of the environment variables required to intercept server broadcasts from an ArcSDE service installed on Windows NT. These variables would be set in the %SDEHOME%\etc\dbinit.sde file but would not take effect until the ArcSDE service is restarted.

```
set SDEINTERCEPT=crwtf
set SDEINTERCEPTLOC=D:\tmp\sde_server
```

Following the ArcSDE service restart, any subsequent application connects to the ArcSDE service will each create a file in the D:\tmp directory with the prefix sde_server. These files will contain the server broadcasts generated during the application's ArcSDE session.

ArcSDE home directory

A

IN THIS APPENDIX

- **ArcSDE home directory (SDEHOME)**
- **ArcSDE client executable image (UNIX only)**

This appendix contains information about the ArcSDE home directory (SDEHOME) and the files installed in each subdirectory. It also includes details on switching the ArcSDE client executable image on a UNIX platform.

ArcSDE home directory (SDEHOME)

The ArcSDE home directory or SDEHOME, sdeexe81, contains the following subdirectories and files. For ArcSDE installed on Windows NT platforms, the executables and library files include the .exe and .dll extensions. For UNIX installations, the executables do not have extensions and the extensions on the shared library files vary depending on the UNIX operating system.

Directory	Contents
bin	Contains all ArcSDE 8.1 executable programs
documentation	Contains the ArcSDE 8.1 online documentation
etc	Contains all extra system files
geocode	Contains the geocoding support files and templates
include	Contains all ArcSDE client include files
lib	Contains libraries
locale	Contains National Language Support files
ssa	Directory for server-side applications
sysgen	Contains license manager files (UNIX only) (The default license manager files on Windows NT platforms are located in C:\Program Files\ESRI\License.)
tools	Contains database management system DBMS tools

bin	Description
cov2sde	Converts coverage to layers
gdbs	The Geodatabase Metadata Maintenance tool
giomgr	I/O manager process executable
gsrvr	On UNIX systems, it is a link to either gsrvr.shared or gsrvr.static. On Windows NT systems, it is the ArcSDE server executable
gsrvr.shared	Shared version of the ArcSDE server executable
gsrvr.static	Static version of the ArcSDE server executable
sde2cov	ArcSDE-to-coverage converter
sde2shp	ArcSDE-to-shape converter
sde2tbl	Converts ArcSDE tables to other DBMS table formats
sdedbtune	Imports and exports the contents of the dbtune table to and from a file
sdeexport	Creates export files of ArcSDE layers
sdegroup	Combines shapes into multipart shapes
sdeimport	Imports ArcSDE export files
sdelayer	Manages layers
sdelocator	Manages locator files
sdelog	Manages log files
sdemon	Manages the ArcSDE service and processes
sdeservice	Creates, deletes, and modifies the ArcSDE service (Windows NT only)
sdesetup*	Setup program for ArcSDE database

bin	Description (continued)
sdetable	Manages business tables.
sdeversion	Manages versioned tables.
sdexinfo	Lists ArcSDE export file information.
shp2sde	Shape-to-ArcSDE converter.
shpinfo	Displays shapefile statistics.
tbl2sde	Converts database management system (DBMS) table formats to ArcSDE tables.
AfCore.dll	License manager support library (Windows NT only).
AfLockMgr.dll	Provides file locking for all applications running on the same host PC.
dforrt.dll	FORTRAN runtime libraries used with data loaders (Windows NT only).
edge.dll	Shared ArcInfo library (Windows NT only).
gpservice.exe	ArcToolBox geoprocessing executable (Windows NT only).
gsrvr*81.dll	Direct connect driver (Windows NT only).
loceng.dll	Shared ArcInfo library (Windows NT only).
mtchloc.dll	Shared ArcInfo library (Windows NT only).
xyloc.dll	Shared ArcInfo library (Windows NT only).
mtch.dll	Shared ArcInfo library (Windows NT only).
sdbase.dll	Shared ArcInfo library (Windows NT only).
sdfeat.dll	Shared ArcInfo library (Windows NT only).
sdgridio.dll	Shared ArcInfo library (Windows NT only).

bin	Description (continued)
sdshape.dll	Shared ArcInfo library (Windows NT only).
sdelicserv	Used with the license manager.
sderelease	Lists release and upgrades server version.
sderemote	Remote startup executable (UNIX only).
jsg81.dll	(Windows NT only).
jsde81.dll	Shared Java™ library (Windows NT only).
pe81.dll	Shared projection run-time library (Windows NT only).
ras81.dll	Shared raster run-time library (Windows NT only).
sdeora8isrvr81.dll	ArcSDE server library (Windows NT only).
sde81.dll	Shared run-time library (Windows NT only).
sde81_trace.dll	Shared function trace library (Windows NT only).
sg81.dll	Shared shape geometry run-time library (Windows NT only).

documentation	Description
ArcSDE Developer Help	
ArcSDE Developer Help/LayerSchema.pdf	
ArcSDE Developer Help/SDEHelp.chm	
ArcSDE Developer Help/SdeSchema.pdf	

etc	Description
dbinit.sde	Contains the DBMS connection information (not present by default on UNIX installations)
dbtune.sde	The default configuration keyword file
dbtune.proto	The prototype configuration keyword file
giomgr.defs	Text file containing the ArcSDE services configuration parameters
giomgr.log	Messages from giomgr process (created after ArcSDE service is started)
services.sde	ArcSDE Transmission Control Protocol/ Internet Protocol (TCP/IP) service name
sde.errlog	Contains error messages from giomgr and gsrvr processes (created after ArcSDE service is started)
sde.outlog	Contains output messages from gsrvr processes (created after ArcSDE service is started)
sdelic.log	Contains any error messages from the license manager (created after ArcSDE service is started)

include	Description
pe.h	Projection Engine data structures/types and functions
pe_coordsys_from_prj.h	
	Projection Engine predefined objects
pedef.h	Projection Engine predefined objects
pef.h	Projection Engine functions for predefined objects
sdeerno.h	Error codes include file
sderaster.h	Raster codes include file
sdetype.h	ArcSDE data structures/types
sg.h	Shape data structures/types
sgerr.h	Shape error include file

geocode	Description
template	Directory of geocoding templates

lib	Description
libmtch.so	Shared library for geolocation (all UNIX except HP®)
libmtch.sl	Shared library for geolocation (HP only)
libpe81.a	Static library for projection engine library
libpe81.sl	Shared library for projection engine (HP only)
libpe81.so	Shared library for projection engine (all UNIX except HP)
libras81.a	Static library for rasters library (UNIX)
libras81.sl	Shared library for rasters (HP only)
libras81.so	Shared library for rasters (all UNIX except HP)
libsde81.a	Static library (UNIX)
libsde81.sl	Shared run-time library (HP only)
libsde81.so	Shared run-time library (all UNIX except HP)
libsde81_trace.a	Static library for function tracing library (UNIX)
libsde81_trace.sl	Shared run-time library for function tracing (HP only)
libsde81_trace.so	Shared run-time library for function tracing (all UNIX except HP)
libsdesrvr81.sl	Shared run-time library (HP only)
libsdesrvr81.so	Shared run-time library (all UNIX except HP)
sdetable	Manages business tables
libsg81.a	Static shape geometry library (UNIX)
libsg81.sl	Shared shape geometry library (HP only)
libsg81.so	Shared shape geometry library (all UNIX)
libxlf.a	Shared library (IBM only)

lib	Description (continued)
libxlf90.a	Shared library (IBM only)
make.include	UNIX platform-dependent file for compiler/link options
metadata_util.spb	Stored procedure for user metadata maintenance
metadata_util.sps	Stored procedure for user metadata maintenance
pe81.lib	Export library for pe40.dll (Windows NT only)
ras81.lib	Export library for ras40.dll (Windows NT only)
sde81.lib	Export library for sde40.dll (Windows NT only)
sde81_trace.lib	Export library for sde40_trace.dll (Windows NT only)
sg81.lib	Export library for sg40.dll (Windows NT only)
xdr81.lib	Export library for the xdr40.dll
dbtune_util.spb	Stored procedure for dbtune table maintenance
dbtune_util.sps	Stored procedure for dbtune table maintenance
layers_util.spb	Stored procedure for layer maintenance
layers_util.sps	Stored procedure for layer maintenance
locator_util.spb	Stored procedure for locator maintenance
locator_util.sps	Stored procedure for locator maintenance
lock_util.spb	Stored procedure for shared lock maintenance

lib	Description (continued)
lock_util.sps	Stored procedure for shared lock maintenance
pinfo_util.spb	Stored procedure for process information maintenance
pinfo_util.sps	Stored procedure for process information maintenance
rastercolumns_util.spb	Stored procedure for raster data
rastercolumns_util.sps	Stored procedure for raster data
registry_util.spb	Stored procedure for registry maintenance
registry_util.sps	Stored procedure for registry maintenance
sde_util.sps	Stored procedure for ArcSDE maintenance
sref_util.spb	Stored procedure for spatial reference maintenance
sref_util.sps	Stored procedure for spatial reference maintenance
varlong_util.spb	Stored procedure for “varlong” maintenance
varlong_util.sps	Stored procedure for “varlong” maintenance
version_user_ddl.spb	Stored procedure for version DDL
version_user_ddl.sps	Stored procedure for version DDL
version_util.spb	Stored procedure for version maintenance
version_util.sps	Stored procedure for version maintenance

locale (UNIX)	Description
codepage	Various codepage files
msg	Error message files

ssa	Description
libloceng.so	Shared library for geocoding operations
liblocssa.so	Shared library for geocoding operations
libmtchloc.so	Shared library for geocoding operations
libxyloc.so	Shared library for geocoding operations
liblocssa.dll	Plug-in file for geocoding operations

sysgen (UNIX)	Description
ESRI	License daemon
adminlicense	Utility to help verify the license.dat file (UNIX only)
license.boot	Scripts to automatically start the license manager during a reboot (UNIX only)
license.boot_HP11	Scripts to automatically start the license manager during a reboot (HP only)
license.dat	License file used by the ESRI daemon to control licenses (UNIX only) (received following installation from ESRI customer service)
lmgrd	Starts the ESRI daemon

sysgen (UNIX)	Description (continued)
lmutil	FLEXlm utility program to manage the license file
sample.dat	Sample license.dat file
SDE.lic	License file used by the ESRI daemon to control licenses (Windows NT only)

tools	Description
generic	Directory that contains layer maintenance tools
generic/load_to_normal	Sample script that converts a layer from load_only_io mode to normal_io mode
generic/make_database	Sample script that creates an ArcSDE database
<rdbms>	Directory that contains RDBMS-specific maintenance and monitoring tools
gdbs	Executable file that creates metadata tables required by ArcInfo 8.1
killp	Executable file to terminate stalled ArcSDE processes (Windows NT only)

ArcSDE client executable image (UNIX only)

The ArcSDE service creates a gsrvr process for each client application connected to the service. These processes respond to requests from the client applications. Each process runs until the application program terminates or the program makes a call to terminate the ArcSDE session.

The UNIX ArcSDE software has two types of gsrvr executables: a static image, gsrvr.static, and a shared image, gsrvr.shared, both of which are found in the $SDEHOME/bin directory.

All code sections of the static image are statically linked. The resulting image is relatively large and requires more memory to execute, but it is fast.

The shared image uses shared libraries from the operating system, the underlying RDBMS, and the ArcSDE service. As a result, the shared image is substantially smaller and uses fewer resources.

If you wish to switch to a shared client executable image, execute the following commands (remember to log in to the system as the ArcSDE administrator):

```
$ cd $SDEHOME/bin
$ rm gsrvr
$ ln -s gsrvr.shared gsrvr
```

To change to a static executable image, execute the following commands:

```
$ cd $SDEHOME/bin
$ rm gsrvr
$ ln -s gsrvr.static gsrvr
```

By default, the installation on UNIX sets up the ArcSDE service to use the shared image. Do not attempt to switch executable images until the ArcSDE service has been shut down.

ArcSDE data dictionary

B

IN THIS APPENDIX

- ArcSDE system tables
- Geodatabase system tables

The ArcSDE data dictionary includes tables that maintain information about the feature classes and feature datasets. The feature class and feature dataset spatial references, states, and versions are also maintained within the ArcSDE data dictionary. The ArcSDE data dictionary combines both the ArcSDE system tables, created by the giomgr process when the ArcSDE service is started for the first time, and the geodatabase system tables, created by the gdbs executable file (located in the SDEHOME\tools directory).

ArcSDE creates and maintains these tables and views. They were not designed to be accessed by external programs. Changing the data within the ArcSDE data dictionary, either by an external program or manually, is not supported.

The purpose of illustrating the structure of the data dictionary here is to provide an inventory of database objects that must be backed up for later restoration in the event of system failure.

ArcSDE system tables

For each ArcSDE service, the ArcSDE software creates several system tables within the ArcSDE user's schema to manage spatial data.

The ArcSDE for SQLSERVER product prefixes each of these tables with SDE_.

VERSION table

The VERSION table maintains information about the ArcSDE version with which the database expects to operate. The table contains the specific release identification for the most recent version of ArcSDE that executed a version update. The ArcSDE giomgr process checks this table to ensure proper version compatibility.

The VERSION table and other ArcSDE system tables are updated by the sdesetup<DBMS> program after a new version of ArcSDE is installed.

VERSION

Name	Data_Type	Null?
major	SE_INTEGER	NOT NULL
minor	SE_INTEGER	NOT NULL
bugfix	SE_INTEGER	NOT NULL
description	SE_STRING_TYPE(96)	NOT NULL
release	SE_INTEGER	NOT NULL

LAYERS table

The LAYERS table maintains data about each feature class in the database. The information helps to build and maintain spatial indexes, ensure proper shape types, maintain data integrity, and store the spatial reference for the coordinate data.

It includes the following information:

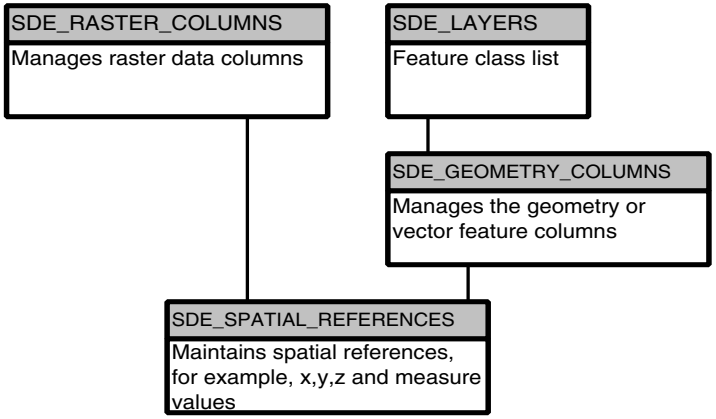
- Owner, table, and column name for the shape column
- Name of the table containing the actual shape data
- Spatial index grid cell sizes
- Envelope (minx, miny, maxx, maxy)
- ArcSDE assigned layer ID for internal use
- Feature class description
- Statistical information on the shapes in the feature class for data transfer buffer configurations

LAYERS

Name	Data_Type	Null?
Layer_id	SE_INTEGER_TYPE	NOT NULL
Description	SE_STRING_TYPE(65)	NULL
Database_name	SE_STRING_TYPE(32)	NULL
Owner	SE_STRING_TYPE(32)	NOT NULL
Table_name	SE_STRING_TYPE(160)	NOT NULL
Spatial_column	SE_STRING_TYPE(32)	NOT NULL
eflags	SE_INTEGER_TYPE	NOT NULL
Layer_mask	SE_INTEGER_TYPE	NOT NULL
gsize1	SE_FLOAT_TYPE	NOT NULL
gsize2	SE_FLOAT_TYPE	NOT NULL
gsize3	SE_FLOAT_TYPE	NOT NULL

LAYERS (continued)

Name	Data_Type	Null?
minx	SE_FLOAT_TYPE	NULL
miny	SE_FLOAT_TYPE	NULL
maxx	SE_FLOAT_TYPE	NULL
maxy	SE_FLOAT_TYPE	NULL
cdate	SE_INTEGER_TYPE	NOT NULL
layer_config	SE_STRING_TYPE(32)	NULL
optimal_array_size	SE_INTEGER_TYPE	NULL
stats_date	SE_INTEGER_TYPE	NULL
minimum_id	SE_INTEGER_TYPE	NULL
srid	SE_INTEGER_TYPE	NOT NULL
base_layer_ID	SE_INTEGER_TYPE	NOT NULL



The LAYERS, RASTER_COLUMNS, and SPATIAL_REFERENCES tables

GEOMETRY_COLUMNS table

The GEOMETRY_COLUMNS table contains the feature class names, their geometry's storage type, and coordinate dimension.

GEOMETRY_COLUMNS

Name	Data_Type	Null?
F_table_catalog	SE_STRING_TYPE(32)	NULL
F_table_schema	SE_STRING_TYPE(32)	NOT NULL
F_table_name	SE_STRING_TYPE(160)	NOT NULL
F_geometry_column	SE_STRING_TYPE(32)	NOT NULL
G_table_catalog	SE_STRING_TYPE(32)	NULL
G_table_schema	SE_STRING_TYPE(32)	NOT NULL
G_table_name	SE_STRING_TYPE(160)	NOT NULL
storage_type	SE_INTEGER_TYPE	NULL
geometry_type	SE_INTEGER_TYPE	NULL
coordinate_dimension	SE_INTEGER_TYPE	NULL
max_ppr	SE_INTEGER_TYPE	NULL
srid	SE_INTEGER_TYPE	NOT NULL

RASTER_COLUMNS table

The RASTER_COLUMNS table contains a list of raster columns stored in the database. The table and raster column names, the owner, creation date, description, database name, configuration keyword, and minimum ID are included. The database name is required for some database management systems (DBMSs).

RASTER_COLUMNS

Name	Data_Type	Null?
rastercolumn_id	SE_INTEGER_TYPE	NOT NULL
description	SE_STRING_TYPE(65)	NULL
database_name	SE_STRING_TYPE(32)	NULL
owner	SE_STRING_TYPE(32)	NOT NULL
table_name	SE_STRING_TYPE(160)	NOT NULL
raster_column	SE_STRING_TYPE(32)	NOT NULL
cdate	SE_INTEGER_TYPE	NOT NULL
config_keyword	SE_STRING_TYPE(32)	NULL
minimum_id	SE_INTEGER_TYPE	NULL
base_rastercolumn_id	SE_INTEGER_TYPE	NOT NULL
rastercolumn_mask	SE_INTEGER_TYPE	NOT NULL
srid	SE_INTEGER_TYPE	NULL

SPATIAL_REFERENCES table

The SPATIAL_REFERENCES table contains the coordinate system and floating point-to-integer transformation values. Internal functions use the parameters of a spatial reference system to translate and scale each floating point coordinate of the geometry into 32-bit positive integers prior to storage. Upon retrieval, the coordinates are restored to their original external floating point format.

SPATIAL_REFERENCES

Name	Data_Type	Null?
srid	SE_INTEGER_TYPE	NOT NULL
description	SE_STRING_TYPE(64)	NULL
auth_name	SE_STRING_TYPE(256)	NULL
auth_srid	SE_INTEGER_TYPE	NULL
falsex	SE_FLOAT_TYPE	NOT NULL
falsey	SE_FLOAT_TYPE	NOT NULL
xyunits	SE_FLOAT_TYPE	NOT NULL
falsez	SE_FLOAT_TYPE	NOT NULL
zunits	SE_FLOAT_TYPE	NOT NULL
falsem	SE_FLOAT_TYPE	NOT NULL
munits	SE_FLOAT_TYPE	NOT NULL
srtext	SE_STRING_TYPE(1024)	NOT NULL

TABLE_REGISTRY table

The TABLE_REGISTRY table manages all registered tables. The values include an ID, table name, owner, and description.

TABLE_REGISTRY

Name	Data_Type	Null?
registration_id	SE_INTEGER_TYPE	NOT NULL
table_name	SE_STRING_TYPE(160)	NOT NULL
owner	SE_STRING_TYPE(32)	NOT NULL
rowid_column	SE_STRING_TYPE(32)	NULL
description	SE_STRING_TYPE(65)	NULL
object_flags	SE_INTEGER_TYPE	NOT NULL
registration_date	SE_INTEGER_TYPE	NOT NULL
config_keyword	SE_STRING_TYPE(32)	NULL
minimum_id	SE_INTEGER_TYPE	NULL
imv_view_name	SE_STRING_TYPE(32)	NULL

VERSIONS table

The VERSIONS table contains the version metadata. The values include a name, owner, status (public or private), state ID, and description.

VERSIONS

Name	Data_Type	Null?
name	SE_STRING_TYPE(64)	NOT NULL
owner	SE_STRING_TYPE(32)	NOT NULL
status	SE_INTEGER_TYPE	NOT NULL
state_id	SE_INTEGER_TYPE	NOT NULL

VERSIONS (continued)

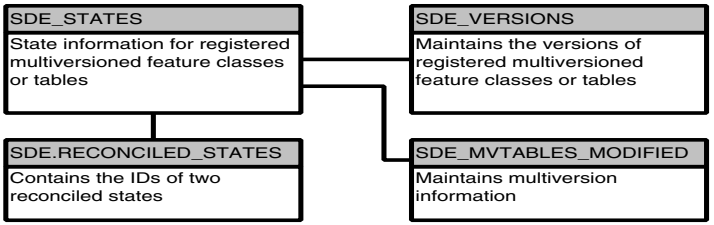
Name	Data_Type	Null?
description	SE_STRING_TYPE(65)	NULL
parent_name	SE_STRING_TYPE(64)	NULL
parent_owner	SE_STRING_TYPE(32)	NULL
parent_version_id	SE_INTEGER_TYPE	NULL
creation_time	SE_DATE_TYPE	NOT NULL

STATES table

The STATES table contains the state metadata. The values include a state ID, owner, creation and closing time, state ID of the parent state, and lineage information.

STATES

Name	Data_Type	Null?
state_id	SE_INTEGER_TYPE	NOT NULL
owner	SE_STRING_TYPE(32)	NOT NULL
creation_time	SE_DATE_TYPE	NOT NULL
closing_time	SE_DATE_TYPE	NULL
parent_state_id	SE_INTEGER_TYPE	NOT NULL
lineage_length	SE_INTEGER_TYPE	NOT NULL
lineage	SE_BLOB_TYPE	NULL



STATES and VERSIONS tables

MVTABLES_MODIFIED table

The MVTABLES_MODIFIED table contains the state and table IDs modified in a given state.

MVTABLES_MODIFIED

Name	Data_Type	Null?
state_id	SE_INTEGER_TYPE	NOT NULL
registration_id	SE_INTEGER_TYPE	NOT NULL

STATE_LINEAGES table

The STATE_LINEAGES table contains a state's ID and its ancestory lineage ID.

STATE_LINEAGES

Name	Data_Type	Null?
state_id	SE_INTEGER_TYPE	NOT NULL
lineage_id	SE_INTEGER_TYPE	NOT NULL

METADATA table

The METADATA table contains ArcSDE metadata.

METADATA

Name	Data_Type	Null?
record_id	SE_INTEGER_TYPE	NOT NULL
object_name	SE_STRING_TYPE(32)	NOT NULL
object_owner	SE_STRING_TYPE(32)	NOT NULL
object_type	SE_INTEGER_TYPE	NOT NULL
class_name	SE_STRING_TYPE(32)	NULL
property	SE_STRING_TYPE(32)	NULL
prop_value	SE_STRING_TYPE(255)	NULL
description	SE_STRING_TYPE(65)	NULL
creation_date	SE_DATE_TYPE	NOT NULL

LOCATORS table

The LOCATORS table stores information about locator objects.

LOCATORS

Name	Data_Type	Null?
locator_id	SE_INTEGER_TYPE	NOT NULL
name	SE_STRING_TYPE(32)	NOT NULL
owner	SE_STRING_TYPE(32)	NOT NULL
category	SE_STRING_TYPE(32)	NOT NULL
type	SE_INTEGER_TYPE	NOT NULL
description	SE_STRING_TYPE(64)	NULL

DBTUNE table

The DBTUNE table stores the configuration keywords for ArcSDE data objects.

DBTUNE

Name	Data_Type	Null?
keyword	SE_STRING_TYPE(32)	NOT NULL
parameter	SE_STRING_TYPE(32)	NOT NULL
config_string	SE_STRING_TYPE(2048)	NULL

LAYER_LOCKS table (DBS service only)

The LAYER_LOCKS table maintains the locks on feature classes.

LAYER_LOCKS

Name	Data_Type	Null?
sde_id	SE_INTEGER_TYPE	NOT NULL
layer_id	SE_INTEGER_TYPE	NOT NULL
autolock	SE_STRING_TYPE(1)	NOT NULL
lock_type	SE_STRING_TYPE(1)	NOT NULL
minx	SE_INTEGER_TYPE	NULL
miny	SE_INTEGER_TYPE	NULL
maxx	SE_INTEGER_TYPE	NULL
maxy	SE_INTEGER_TYPE	NULL

OBJECT_LOCKS table (DBS service only)

The OBJECT_LOCKS table maintains locks on Geodatabase objects.

OBJECT_LOCKS

Name	Data_Type	Null?
sde_id	SE_INTEGER_TYPE	NOT NULL
object_id	SE_INTEGER_TYPE	NOT NULL
object_type	SE_INTEGER_TYPE	NOT NULL
application_id	SE_INTEGER_TYPE	NOT NULL
autolock	SE_INTEGER_TYPE	NOT NULL
lock_type	SE_STRING_TYPE(1)	NOT NULL

STATE_LOCKS table (DBS service only)

The STATE_LOCKS table maintains the version state locks.

STATE_LOCKS

Name	Data_Type	Null?
sde_id	SE_INTEGER_TYPE	NOT NULL
state_id	SE_INTEGER_TYPE	NOT NULL
autolock	SE_STRING_TYPE(1)	NOT NULL
lock_type	SE_STRING_TYPE(1)	NOT NULL

TABLE_LOCKS table

The TABLE_LOCKS table maintains the locks on ArcSDE registered tables.

TABLE_LOCKS

Name	Data_Type	Null?
sde_id	SE_INTEGER_TYPE	NOT NULL
registration_id	SE_INTEGER_TYPE	NOT NULL
lock_type	SE_STRING_TYPE(1)	NOT NULL

SDE_TABLES_MODIFIED table

The SDE_TABLES_MODIFIED table maintains the list of modified tables.

SDE_TABLES_MODIFIED

Name	Data_Type	Null?
table_name	SE_STRING_TYPE(32)	NOT NULL
time_last_modified	SE_DATE_TYPE	NOT NULL

PROCESS_INFORMATION table

The PROCESS_INFORMATION table collects ArcSDE session statistics such as the number of records read and the number of records written while the session was active.

PROCESS_INFORMATION

Name	Data_Type	Null?
sde_id	SE_INTEGER_TYPE	NOT NULL
server_id	SE_INTEGER_TYPE	NOT NULL
start_time	SE_DATE_TYPE	NOT NULL
rcount	SE_INTEGER_TYPE	NOT NULL
wcount	SE_INTEGER_TYPE	NOT NULL
opcount	SE_INTEGER_TYPE	NOT NULL
numlocks	SE_INTEGER_TYPE	NOT NULL
fb_partial	SE_INTEGER_TYPE	NOT NULL
fb_count	SE_INTEGER_TYPE	NOT NULL
fb_fdcount	SE_INTEGER_TYPE	NOT NULL
fb_kbyteS	SE_INTEGER_TYPE	NOT NULL
owner	SE_STRING_TYPE(30)	NOT NULL
direct_connect	SE_STRING_TYPE(1)	NOT NULL

GDB_ANNOSYMBOLS table

The GDB_ANNOSYMBOLS table contains feature class annotation. The values include annosymbol ID and the annotation string.

GDB_ANNOSYMBOLS

Name	Data_Type	Null?
id	SE_INTEGER_TYPE	NOT NULL
symbol	SE_BLOB_TYPE	NULL

GDB_ATTRRULES table

The GDB_ATTRRULES table contains the rules for each attribute domain.

GDB_ATTRRULES

Name	Data_Type	Null?
ruleid	SE_INTEGER_TYPE	NOT NULL
subtype	SE_INTEGER_TYPE	NOT NULL
fieldname	SE_STRING_TYPE(32)	NOT NULL
domainname	SE_STRING_TYPE(160)	NOT NULL

GDB_CODEDDOMAINS table

The GDB_CODEDDOMAINS table contains coded values for each domain.

GDB_CODEDDOMAINS

Name	Data_Type	Null?
domainid	SE_INTEGER_TYPE	NOT NULL
codedvalues	SE_BLOB_TYPE	NOT NULL

GDB_DEFAULTVALUES table

The GDB_DEFAULTVALUES table contains the default values for the subtypes of each object class.

GDB_DEFAULTVALUES

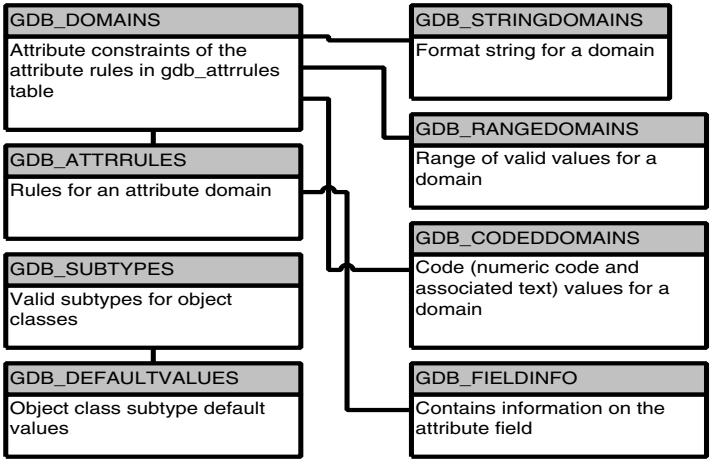
Name	Data_Type	Null?
classid	SE_INTEGER_TYPE	NOT NULL
fieldname	SE_STRING_TYPE(32)	NOT NULL
subtype	SE_INTEGER_TYPE	NOT NULL
defaultstring	SE_STRING_TYPE(160)	NULL
defaultnumber	SE_DOUBLE_TYPE(38,8)	NULL

GDB_DOMAINS table

The GDB_DOMAINS table contains the attribute constraints associated with attribute rules of the GDB_ATTRRULES table.

GDB_DOMAINS

Name	Data_Type	Null?
id	SE_INTEGER_TYPE	NOT NULL
owner	SE_STRING_TYPE(32)	NOT NULL
domainname	SE_STRING_TYPE(160)	NOT NULL
description	SE_STRING_TYPE(160)	NULL
domaintype	SE_INTEGER_TYPE	NOT NULL
fieldtype	SE_INTEGER_TYPE	NOT NULL
mergepolicy	SE_INTEGER_TYPE	NOT NULL
splitpolicy	SE_INTEGER_TYPE	NOT NULL



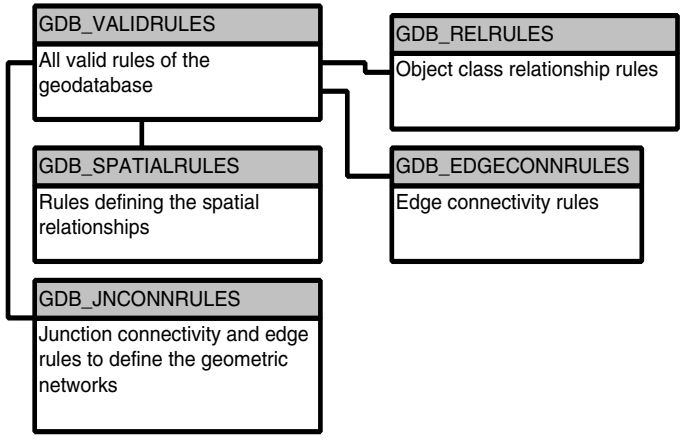
Attribute domain tables

GDB_EDGECONNRULES table

The GDB_EDGECONNRULES table contains the edge connectivity rules. Edge connectivity rules, together with junction rules, function to define the geometric networks stored in the gdb_geonetwork table.

GDB_EDGECONNRULES

Name	Data_Type	Null?
Ruleid	SE_INTEGER_TYPE	NOT NULL
Fromclassid	SE_INTEGER_TYPE	NOT NULL
Fromsubtype	SE_INTEGER_TYPE	NOT NULL
Toclassid	SE_INTEGER_TYPE	NOT NULL
Tosubtype	SE_INTEGER_TYPE	NOT NULL
Junctions	SE_BLOB_TYPE	NOT NULL



The geometric network edge/junction rules

GDB_FEATURECLASSES table

The GDB_FEATURECLASSES table contains the feature classes.

GDB_FEATURECLASSES

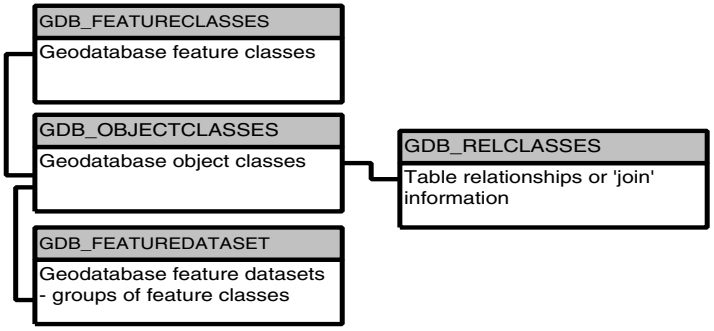
Name	Data_Type	Null?
objectclassid	SE_INTEGER_TYPE	NOT NULL
featuretype	SE_INTEGER_TYPE	NOT NULL
geometrytype	SE_INTEGER_TYPE	NOT NULL
shapefield	SE_STRING_TYPE(32)	NOT NULL
geomnetworkid	SE_INTEGER_TYPE	NULL
graphid	SE_INTEGER_TYPE	NULL

GDB_FEATUREDATASET table

The GDB_FEATUREDATASET table contains the feature datasets. A *feature dataset* is a grouping of feature classes.

GDB_FEATUREDATASET

Name	Data_Type	Null?
id	SE_INTEGER_TYPE	NOT NULL
databasename	SE_STRING_TYPE(32)	NULL
owner	SE_STRING_TYPE(32)	NOT NULL
name	SE_STRING_TYPE(160)	NOT NULL
srid	SE_INTEGER_TYPE	NOT NULL



The feature/object class tables

GDB_FIELDINFO table

The GDB_FIELDINFO table contains the field name, default domain names values, default string, and number values for each attribute field associated with a feature class.

GDB_FIELDINFO

Name	Data_Type	Null?
classid	SE_INTEGER_TYPE	NOT NULL
fieldname	SE_STRING_TYPE(160)	NOT NULL
aliasname	SE_STRING_TYPE(160)	NULL
modelname	SE_STRING_TYPE(160)	NULL
defaultdomainname	SE_STRING_TYPE(160)	NULL
defaultvaluestring	SE_STRING_TYPE(160)	NULL
defaultvaluenumber	SE_DOUBLE_TYPE(38,8)	NULL
isrequired	SE_INTEGER_TYPE	NOT NULL
issubtypefixed	SE_INTEGER_TYPE	NOT NULL
iseditable	SE_INTEGER_TYPE	NOT NULL

GDB_GEOMNETWORKS table

The GDB_GEOMNETWORKS table contains the geometric networks of a feature dataset.

GDB_GEOMNETWORKS

Name	Data_Type	Null?
id	SE_INTEGER_TYPE	NOT NULL
databasename	SE_STRING_TYPE(32)	NULL
owner	SE_STRING_TYPE(32)	NOT NULL
name	SE_STRING_TYPE(160)	NOT NULL
networktype	SE_INTEGER_TYPE	NOT NULL
datasetid	SE_INTEGER_TYPE	NOT NULL

GDB_EXTENSIONS table

The GDB_EXTENSIONS table.

GDB_EXTENSIONS

Name	Data_Type	Null?
id	SE_INTEGER_TYPE	NOT NULL
name	SE_STRING_TYPE(160)	NOT NULL
clsid	SE_STRING_TYPE(38)	NOT NULL

GDB_JNCONNRULES table

The GDB_JNCONNRULES table contains the junction connectivity rules. Junction connectivity rules, together with edges rules, function to define the geometric networks stored in the GDB_GEOMNETWORKS table.

GDB_JNCONNRULES

Name	Data_Type	Null?
ruleid	SE_INTEGER_TYPE	NOT NULL
edgeclassid	SE_INTEGER_TYPE	NOT NULL
edgesubtype	SE_INTEGER_TYPE	NOT NULL
edgemincard	SE_INTEGER_TYPE	NOT NULL
edgemaxcard	SE_INTEGER_TYPE	NOT NULL
junctionclassid	SE_INTEGER_TYPE	NOT NULL
junctionsubtype	SE_INTEGER_TYPE	NOT NULL
junctionmincard	SE_INTEGER_TYPE	NOT NULL
junctionmaxcard	SE_INTEGER_TYPE	NOT NULL
isdefault	SE_INTEGER_TYPE	NULL

GDB_NETCLASSES table

The GDB_NETCLASSES table contains the network classes of the geometric networks.

GDB_NETCLASSES

Name	Data_Type	Null?
classid	SE_INTEGER_TYPE	NOT NULL
networkid	SE_INTEGER_TYPE	NOT NULL
enabledfield	SE_STRING_TYPE(32)	NULL
ancillaryrole	SE_INTEGER_TYPE	NULL
ancillaryfield	SE_STRING_TYPE(32)	NULL

GDB_NETWEIGHTS table

The GDB_NETWEIGHTS table contains the network weights of the geometric networks.

GDB_NETWEIGHTS

Name	Data_Type	Null?
oid	SE_INTEGER_TYPE	NOT NULL
networkid	SE_INTEGER_TYPE	NOT NULL
name	SE_STRING_TYPE(160)	NOT NULL
weightid	SE_INTEGER_TYPE	NOT NULL
weighttype	SE_INTEGER_TYPE	NOT NULL
bitgatesize	SE_INTEGER_TYPE	NULL

GDB_NETWEIGHTASOCS table

The GDB_NETWEIGHTASOCS table contains the association between the network classes and the network weights of the geometric networks.

GDB_NETWEIGHTASOCS

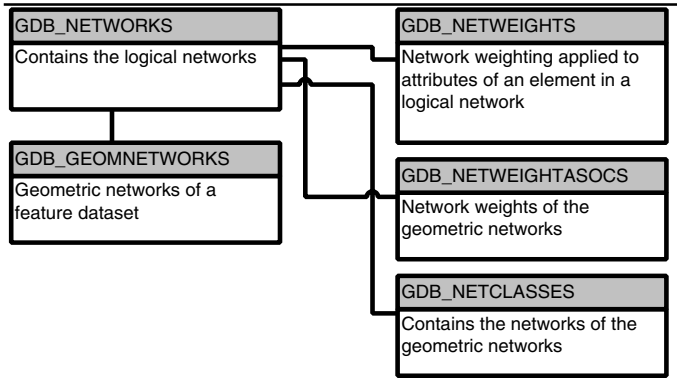
Name	Data_Type	Null?
networkid	SE_INTEGER_TYPE	NOT NULL
weightid	SE_INTEGER_TYPE	NOT NULL
tablename	SE_STRING_TYPE(160)	NOT NULL
fieldname	SE_STRING_TYPE(32)	NULL

GDB_NETWORKS table

The GDB_NETWORKS table contains the logical networks.

GDB_NETWORKS

Name	Data_Type	Null?
id	SE_INTEGER_TYPE	NOT NULL
databasename	SE_STRING_TYPE(32)	NULL
owner	SE_STRING_TYPE(32)	NOT NULL
name	SE_STRING_TYPE(160)	NOT NULL
networktype	SE_INTEGER_TYPE	NOT NULL
indextype	SE_INTEGER_TYPE	NOT NULL
normalized	SE_INTEGER_TYPE	NOT NULL



The logical network tables

GDB_OBJECTCLASSES table

The GDB_OBJECTCLASSES table contains all of the object classes in the geodatabase, which includes the feature classes, relationship classes, business tables, and columns.

GDB_OBJECTCLASSES

Name	Data_Type	Null?
id	SE_INTEGER_TYPE	NOT NULL
databasename	SE_STRING_TYPE(32)	NULL
owner	SE_STRING_TYPE(32)	NOT NULL
name	SE_STRING_TYPE(160)	NOT NULL
aliasname	SE_STRING_TYPE(160)	NULL
modelname	SE_STRING_TYPE(160)	NULL
clsid	SE_STRING_TYPE(38)	NOT NULL
extclsid	SE_STRING_TYPE(38)	NULL
extprops	SE_BLOB_TYPE	NULL
subtypefield	SE_STRING_TYPE(32)	NULL
datasetid	SE_INTEGER_TYPE	NULL

GDB_RANGEDOMAINS table

The GDB_RANGEDOMAINS table contains the range of possible values allowed in a domain.

GDB_RANGEDOMAINS

Name	Data_Type	Null?
domainid	SE_INTEGER_TYPE	NOT NULL
minvalue	SE_DOUBLE_TYPE(38,8)	NOT NULL
maxvalue	SE_DOUBLE_TYPE(38,8)	NOT NULL

GDB_RELCLASSES table

The GDB_RELCLASSES table contains the table relationships required by the geodatabase.

GDB_RELCLASSES

Name	Data_Type	Null?
id	SE_INTEGER_TYPE	NOT NULL
databasename	SE_STRING_TYPE(32)	NULL
owner	SE_STRING_TYPE(32)	NOT NULL
name	SE_STRING_TYPE(160)	NOT NULL
originclassid	SE_INTEGER_TYPE	NOT NULL
destclassid	SE_INTEGER_TYPE	NOT NULL
forwardlabel	SE_STRING_TYPE(32)	NULL
backwardlabel	SE_STRING_TYPE(32)	NULL
cardinality	SE_INTEGER_TYPE	NOT NULL
notification	SE_INTEGER_TYPE	NOT NULL
iscomposite	SE_INTEGER_TYPE	NOT NULL
isattributed	SE_INTEGER_TYPE	NOT NULL
originprimarykey	SE_STRING_TYPE(32)	NOT NULL
destprimarykey	SE_STRING_TYPE(32)	NOT NULL
originforeignkey	SE_STRING_TYPE(32)	NOT NULL
destforeignkey	SE_STRING_TYPE(32)	NOT NULL
datasetid	SE_INTEGER_TYPE	NULL

GDB_RELEASE table

The GDB_RELEASE table stores geodatabase version release information as a single record.

GDB_RELEASE

Name	Data_Type	Null?
major	SE_INTEGER_TYPE	NOT NULL
minor	SE_INTEGER_TYPE	NOT NULL
bugfix	SE_INTEGER_TYPE	NOT NULL

GDB_RELRULES table

The GDB_RELRULES table contains the object class relationship rules.

GDB_RELRULES

Name	Data_Type	Null?
ruleid	SE_INTEGER_TYPE	NOT NULL
originsubtype	SE_INTEGER_TYPE	NOT NULL
originmincard	SE_INTEGER_TYPE	NOT NULL
originmaxcard	SE_INTEGER_TYPE	NOT NULL
destsubtype	SE_INTEGER_TYPE	NOT NULL
destmincard	SE_INTEGER_TYPE	NOT NULL
destmaxcard	SE_INTEGER_TYPE	NOT NULL

GDB_SPATIALRULES table

The GDB_SPATIALRULES table contains the spatial rules of the geodatabase. Such rules determine which spatial relationships are permitted.

GDB_SPATIALRULES

Name	Data_Type	Null?
rruleid	SE_INTEGER_TYPE	NOT NULL
subtype	SE_INTEGER_TYPE	NOT NULL
spatialrel	SE_INTEGER_TYPE	NOT NULL
distance	SE_DOUBLE_TYPE(38,8)	NOT NULL
relclassid	SE_INTEGER_TYPE	NOT NULL
relsubtype	SE_INTEGER_TYPE	NOT NULL

GDB_STRINGDOMAINS table

The GDB_STRINGDOMAINS table stores a domain's format string.

GDB_STRINGDOMAINS

Name	Data_Type	Null?
domainid	SE_INTEGER_TYPE	NOT NULL
format	SE_STRING_TYPE(32)	NOT NULL

GDB_SUBTYPES table

The GDB_SUBTYPES table contains the valid subtypes of the geodatabase object classes.

GDB_SUBTYPES

Name	Data_Type	Null?
id	SE_INTEGER_TYPE	NOT NULL
classid	SE_INTEGER_TYPE	NOT NULL
subtypecode	SE_INTEGER_TYPE	NOT NULL
subtypename	SE_STRING_TYPE(160)	NOT NULL

GDB_USERMETADATA table

The GDB_USERMETADATA table stores user metadata for all parts of the geodatabase including object classes, feature classes, feature datasets, logical networks, and relationship classes.

GDB_USERMETADATA

Name	Data_Type	Null?
id	SE_INTEGER_TYPE	NOT NULL
databasename	SE_STRING_TYPE(32)	NULL
owner	SE_STRING_TYPE(32)	NOT NULL
name	SE_STRING_TYPE(160)	NOT NULL
datasettype	SE_INTEGER_TYPE	NOT NULL
xml	SE_BLOB_TYPE	NOT NULL

GDB_VALIDRULES table

The GDB_VALIDRULES table contains all the valid rules of the geodatabase, which includes the attribute rules, edge connectivity rules, junction connectivity rules, relationship rules, and spatial rules.

GDB_VALIDRULES

Name	Data_Type	Null?
id	SE_INTEGER_TYPE	NOT NULL
ruletype	SE_INTEGER_TYPE	NOT NULL
classid	SE_INTEGER_TYPE	NOT NULL
rulecategory	SE_INTEGER_TYPE	NOT NULL
helpstring	SE_STRING_TYPE(160)	NULL

ArcSDE table definitions

C

IN THIS APPENDIX

- **Business tables**
- **Binary schema implementation**
- **Feature table**
- **Spatial index table**
- **Normalized schema**
- **Spatial types and functions schema**
- **Logical network tables**
- **Log file tables**
- **Version tables**
- **Raster tables**

This appendix lists the ArcSDE tables that a user can create. These tables should be included within your existing backup arrangements. With the exception of the business table, these tables should only be accessed through the application interface provided either by ArcInfo or an ArcSDE C application programming interface (API) program. Direct access to the nonbusiness tables via the Structured Query Language (SQL) interface is not supported.

Business tables

The business table is an existing database management system (DBMS) table that ArcSDE spatially enables by adding a spatial column. A business table with a spatial column is a *feature class,* and information about each feature class is maintained in the LAYERS table.

The data type of the spatial column varies depending on the implementation of ArcSDE.

Binary schema implementation

With ArcSDE 8.1, most implementations employ the binary schema. Under this implementation, the integer spatial column contains feature identifiers that uniquely reference the spatial data. The feature ID joins the business table with the associated ArcSDE-managed feature and spatial index tables.

A database trigger is defined on the spatially enabled business table to maintain the relationship between records in the business table and the feature table. The trigger is:

```
TRIGGER SPCOL_DEL_CASCADE_<layer>
AFTER DELETE OR UPDATE OF <SPATIAL_COL> ON
business_table
   IF DELETING THEN
       DELETE FROM F<layer>
       WHERE F<layer>.fid = old.<SPATIAL_COL>

       DELETE FROM S<layer>
       WHERE S<layer>.fid = old.<SPATIAL_COL>


   IF UPDATING AND new.<SPATIAL_COL> IS NULL THEN
      DELETE FROM F<layer>
         WHERE F<layer>.fid = old.<SPATIAL_COL>

      DELETE FROM S<layer>
         WHERE S<layer>.fid = old.<SPATIAL_COL>


   IF UPDATING AND new.<SPATIAL_COL> !=
                             old.<SPATIAL_COL> THEN

     raise_application_error

     (-20013,'Invalid update of SDE Spatial
     Column.')
```

Feature table

Under the binary schema implementation, the feature table stores the geometric shapes for each feature. This table is identified by the spatial column layer number using the name F<layer_id>.

The relationship between the business table and the feature table is managed through the Feature ID, or FID. This key, which is maintained by ArcSDE, is unique for the spatial column.

F<layer_id>

NAME	DATA_TYPE	NULL?
fid	SE_INTEGER_TYPE	NOT NULL
numofpts	SE_INTEGER_TYPE	NOT NULL
entity	SE_SMALLINT_TYPE	NOT NULL
eminx	SE_FLOAT_TYPE(64)	NOT NULL
eminy	SE_FLOAT_TYPE(64)	NOT NULL
emaxx	SE_FLOAT_TYPE(64)	NOT NULL
emaxy	SE_FLOAT_TYPE(64)	NOT NULL
eminz	SE_FLOAT_TYPE(64)	NULL
emaxz	SE_FLOAT_TYPE(64)	NULL
min_measure	SE_FLOAT_TYPE(64)	NULL
max_measure	SE_FLOAT_TYPE(64)	NULL
area	SE_FLOAT_TYPE(64)	NOT NULL
len	SE_FLOAT_TYPE(64)	NOT NULL
points	SE_BLOB_TYPE	NULL

The feature table stores the geometry and has several additional columns to support ArcSDE query processing.

For storing the geometry:

- Points (SE_BLOB)—contains the byte stream of point coordinates that define the geometry

For query processing:

- Area (SE_FLOAT_TYPE)—the area of the geometry
- Len (SE_FLOAT_TYPE)—the length or perimeter of the geometry
- Eminx, eminy, emaxx, emaxy (SE_FLOAT_TYPE)—the envelope of the geometry
- Eminz (SE_FLOAT_TYPE)—the minimum z-value in the geometry
- Emaxz (SE_FLOAT_TYPE)—the maximum z-value in the geometry
- Min_measure (SE_FLOAT_TYPE)—the minimum measure value in the geometry
- Max_measure (SE_FLOAT_TYPE)—the maximum measure value in the geometry

For internal ArcSDE use:

- Fid (SE_INTEGER_TYPE)—contains the unique ID that joins the feature table to the business table
- Entity (SE_INTEGER_TYPE)—the type geometry stored in the spatial column (for example, point, line string)
- Numofpts (SE_INTEGER_TYPE)—the number of points defining the geometry

As geometries are inserted or updated, the extents, numofpts, and so on, are recalculated automatically. The points column contains the coordinate array for the geometry in a compressed integer format. The binary layout of this data is discussed in the *ArcSDE Developer Help*, located in the SDEHOME\documentation directory.

Spatial index table

The spatial index of the binary implementation is the spatial index table. It stores references to shapes based on a simple, regular grid. This table is identified by the spatial column layer number using the name S<layer_id>.

The spatial index contains an entry for each shape and grid cell combination. A feature that crosses into three grid cells has three entries in the table. When a spatial query is performed, the grid cells within the search area are identified and used to return a list of candidate geometries.

- Sp_fid (SE_INTEGER_TYPE)—contains the ID that joins the spatial index table to the feature table
- Gx, gy (SE_INTEGER_TYPE)—the grid cell coordinate
- Eminx, eminy, emaxx, emaxy (SE_INTEGER_TYPE)—the envelope of the geometry

S<layer_id>

NAME	DATA TYPE	NULL?
sp_fid	SE_INTEGER_TYPE	NOT NULL
gx	SE_INTEGER_TYPE	NOT NULL
gy	SE_INTEGER_TYPE	NOT NULL
eminx	SE_INTEGER_TYPE	NOT NULL
eminy	SE_INTEGER_TYPE	NOT NULL
emaxx	SE_INTEGER_TYPE	NOT NULL
emaxy	SE_INTEGER_TYPE	NOT NULL

Normalized schema

In the normalized schema, the coordinate values for a geometry are stored as individual numeric data types in a separate geometry table. Access from a business table to a geometry is through a foreign key—the Geometry ID, or GID.

The geometry table has at least six columns: GID, ETYPE, ESEQ, SEQ, and two coordinate values. The geometry type (point, line string, or polygon) is stored as an ETYPE value. The ETYPE is either 1 (point), 2 (line string), or 3 (polygon). A normalized representation might not store all of a geometry's coordinates in a single row in the geometry table. To accommodate geometries with a large number of values, the geometry is represented by multiple rows. The element sequence value, ESEQ, identifies multiple parts within a geometry. Rows that represent a single part are listed by a sequence (SEQ) value so that rows are returned in the proper order.

The tables below illustrate the relationship between a business table and its geometry table in the normalized geometry representation. Any number of coordinate pairs are allowed for a given table. In this example, two coordinate pairs are stored for each row in the geometry table.

<BUSINESS TABLE>

Feature-ID	Column1	Column2	GeometryID
101			1
102			2
103			3
…			…

A business table with GIDs

<GEOMETRY TABLE>

GID	ETYPE	ESEQ	SEQ	X1	Y1	X2	Y2
1	3	1	1	10.00	10.00	10.00	15.00
1	3	1	2	10.00	15.00	15.00	15.00
1	3	1	3	15.00	15.00	10.00	10.00
1	3	1	4	15.00	10.00	10.00	10.00
2	3	1	1	100.00	100.00	50.00	50.00
2	3	1	2	100.00	150.00	50.00	50.00
2	3	1	3	150.00	150.00	00.00	00.00
2	3	1	4	150.00	100.00	00.00	00.00
2	3	2	1	70.00	90.00	75.00	90.00
…	…	…	…	…	…	…	…

A geometry table with two shapes. The second shape has more than one part.

Spatial types and functions schema

Some of the implementations of ArcSDE employ the latest object-relation technology. Instead of using a separate table to store the geometries, the spatial types and functions schema stores them directly in the spatial column. Under this implementation, the spatial column is a user-defined type (UDT).

The ArcSDE spatial types and functions implementation includes seven of the UDTs defined by OpenGIS® RFP-1. Those data types are point, line string, polygon, multipoint, multilinestring, multipolygon, and geometry.

The database administrator normally runs a program that adds these UDTs and a number of spatial user-defined functions (UDFs) to a database. After the UDTs and UDFs have been added, the database becomes “spatially enabled”. Users may now create tables that include columns whose types are of the seven spatial UDTs listed above. As a bonus, the spatial types and functions allow users to perform spatial queries using the SQL interface provided by the DBMS vendor.

Logical network tables

The logical network tables provide connectivity and flow direction information between elements, junctions, and edges in a network.

The N_*_DESC table maintains a description of each element in the network.

N_*_DESC

Name	Data_Type	Null?
oid	SE_INTEGER_TYPE	NOT NULL
userclassid	SE_SMALLINT_TYPE	NOT NULL
userid	SE_INTEGER_TYPE	NOT NULL
usersubid	SE_INTEGER_TYPE	NOT NULL
elementtype	SE_SMALLINT_TYPE	NOT NULL
eid	SE_INTEGER_TYPE	NOT NULL

The N_*_EDESC table contains a description of the edges within a network.

N_*_EDESC

Name	Data_Type	Null?
oid	SE_INTEGER_TYPE	NOT NULL
pagenumber	SE_INTEGER_TYPE	NOT NULL
pageblob	SE_BLOB_TYPE	NULL

The N_*_ESTATUS table contains the status of each element including its deleted state and disabled state.

N_*_ESTATUS

Name	Data_Type	Null?
oid	SE_INTEGER_TYPE	NOT NULL
pagenumber	SE_INTEGER_TYPE	NOT NULL
pageblob	SE_BLOB_TYPE	NULL

The N_*_ETOPO table contains the edge topology.

N_*_ETOPO

Name	Data_Type	Null?
oid	SE_INTEGER_TYPE	NOT NULL
pagenumber	SE_INTEGER_TYPE	NOT NULL
pageblob	SE_BLOB_TYPE	NULL

The N_*_FLODIR table contains the flow direction.

N_*_FLODIR

Name	Data_Type	Null?
oid	SE_INTEGER_TYPE	NOT NULL
pagenumber	SE_INTEGER_TYPE	NOT NULL
pageblob	SE_BLOB_TYPE	NULL

The N_*_JDESC table contains a description of the junctions of a network.

N_*_JDESC

Name	Data_Type	Null?
oid	SE_INTEGER_TYPE	NOT NULL
pagenumber	SE_INTEGER_TYPE	NOT NULL
pageblob	SE_BLOB_TYPE	NULL

The N_*_JSTATUS table contains the status of each element including its deleted state and disabled state.

N_*_JSTATUS

Name	Data_Type	Null?
oid	SE_INTEGER_TYPE	NOT NULL
pagenumber	SE_INTEGER_TYPE	NOT NULL
pageblob	SE_BLOB_TYPE	NULL

The N_*_JTOPO table contains the junction topology.

N_*_JTOPO

Name	Data_Type	Null?
oid	SE_INTEGER_TYPE	NOT NULL
pagenumber	SE_INTEGER_TYPE	NOT NULL
pageblob	SE_BLOB_TYPE	NULL

The N_*_JTOPO2 table contains the overflow junction topology.

N_*_JTOPO2

Name	Data_Type	Null?
oid	SE_INTEGER_TYPE	NOT NULL
pagenumber	SE_INTEGER_TYPE	NOT NULL
pageblob	SE_BLOB_TYPE	NULL

The N_*_PROPS table contains the network properties including the number of junctions, edges, and turns.

N_*_PROPS

Name	Data_Type	Null?
propertyid	SE_INTEGER_TYPE	NULL
propertyname	SE_STRING_TYPE(255)	NULL
propertyvalue	SE_INTEGER_TYPE	NULL

The N_*_E* table contains all of the network edge weights.

N_*_E*

Name	Data_Type	Null?
oid	SE_INTEGER_TYPE	NOT NULL
name	SE_STRING_TYPE	NULL
internalid	SE_INTEGER_TYPE	NULL
weighttype	SE_INTEGER_TYPE	NULL
maxvalue	SE_DOUBLE_TYPE(38,8)	NULL
minvalue	SE_DOUBLE_TYPE(38,8)	NULL

The N_*_J* table contains all of the network junction weights.

N_*_J*

Name	Data_Type	Null?
oid	SE_INTEGER_TYPE	NOT NULL
name	SE_STRING_TYPE	NULL
internalid	SE_INTEGER_TYPE	NULL
weighttype	SE_INTEGER_TYPE	NULL
maxvalue	SE_DOUBLE_TYPE(38,8)	NULL
minvalue	SE_DOUBLE_TYPE(38,8)	NULL

Log file tables

The SDE_LOGFILES table contains the log file metadata. The values include a log file name and ID, the registration ID, flags, and session information.

SDE_LOGFILES

Name	Data_Type	Null?
logfile_name	SE_STRING_TYPE(256)	NOT NULL
logfile_id	SE_INTEGER_TYPE	NOT NULL
logfile_data_id	SE_INTEGER_TYPE	NOT NULL
registration_id	SE_INTEGER_TYPE	NOT NULL
flags	SE_INTEGER_TYPE	NOT NULL
session_tag	SE_INTEGER_TYPE	NOT NULL

The SDE_LOGFILE_DATA table contains the list of business table records that are part of each log.

SDE_LOGFILE_DATA

Name	Data_Type	Null?
logfile_data_id	SE_INTEGER_TYPE	NOT NULL
sde_row_id	SE_INTEGER_TYPE	NOT NULL

Version tables

The version tables store information about the changes (additions or deletions) that are made to a versioned business table.

The A<registration_ID> table (or the “adds” table) stores the records added to each state in a versioned business table.

A<registration_ID>

Name	Data_Type	Null?
User-defined column names	User-defined data types	
.	.	
.	.	
.	.	
sde_state_id	SE_INTEGER_TYPE	NOT NULL

Conversely, the D<registration_ID> table (or the “deletes” table) stores the records deleted from each state in a versioned business table.

D<registration_ID>

Name	Data_Type	Null?
sde_state_id	SE_INTEGER_TYPE	NOT NULL
sde_row_id	SE_INTEGER_TYPE	NOT NULL
deleted_at	SE_INTEGER_TYPE	NOT NULL

ArcSDE stores images in three raster tables. The raster blocks table—SDE_BLK_<raster_column_ID>—stores the actual image data for each band of the image. The raster band table—SDE_BND_<raster_column_ID>— stores metadata about the bands of the image. The raster description table—SDE_RAS_<raster_column_ID>—stores the description of the images within a raster column. The raster column that is added to the business table stores the raster ID, which joins the business table to the three raster tables.

The raster blocks table— SDE_BLK_<raster_column_ID>— stores the image tiles.

SDE_BLK_<raster_column_ID>

Name	Data_Type	Null?
rasterband_id	SE_INTEGER_TYPE	NOT NULL
rrd_factor	SE_INTEGER_TYPE	NOT NULL
row_nbr	SE_INTEGER_TYPE	NOT NULL
col_nbr	SE_INTEGER_TYPE	NOT NULL
block_data	SE_BLOB_TYPE	NOT NULL

The raster description table—SDE_RAS_<raster_column_ID>—stores the description of the images stored in a raster column.

SDE_RAS_<raster_column_ID>

Name	Data_Type	Null?
raster_id	SE_INTEGER_TYPE	NOT NULL
description	SE_STRING_TYPE(65)	NULL

The raster band table—SDE_BND_<raster_column_ID>—stores information about the bands of the images. Among other things this table stores is the width and height of the band tiles, the pixel type, and pixel depth.

SDE_BND_<raster_column_ID>

Name	Data_Type	Null?
rasterband_id	SE_INTEGER_TYPE	NOT NULL
sequence_nbr	SE_INTEGER_TYPE	NOT NULL
raster_id	SE_INTEGER_TYPE	NOT NULL
name	SE_STRING_TYPE(65)	NULL
band_flags	SE_INTEGER_TYPE	NOT NULL
band_width	SE_INTEGER_TYPE	NOT NULL
band_height	SE_INTEGER_TYPE	NOT NULL
band_types	SE_INTEGER_TYPE	NOT NULL
block_width	SE_INTEGER_TYPE	NOT NULL
block_height	SE_INTEGER_TYPE	NOT NULL
eminx	SE_FLOAT_TYPE(64)	NOT NULL
eminy	SE_FLOAT_TYPE(64)	NOT NULL
emaxx	SE_FLOAT_TYPE(64)	NOT NULL
emaxy	SE_FLOAT_TYPE(64)	NOT NULL
cdate	SE_FLOAT_TYPE(64)	NOT NULL
mdate	SE_FLOAT_TYPE(64)	NOT NULL
statistics	SE_BLOB_TYPE	NULL

The raster auxiliary table—SDE_AUX_<raster_column_ID>—stores the image color map, image statistics, and the optional bit mask that is used for image overlays and mosaicking.

SDE_AUX_<raster_column_ID>

Name	Data_Type	Null?
rasterband_id	SE_INTEGER_TYPE	NOT NULL
type	SE_INTEGER_TYPE	NOT NULL
object	SE_BLOB_TYPE	NOT NULL

ArcSDE service command references

D

IN THIS APPENDIX

- Command listing
- Command syntax
- Getting help
- ArcSDE administration commands: sdemon
- ArcSDE administration commands: sdeservice
- ArcSDE administration commands: sdesetup

Some of the administration commands allow the ArcSDE administrator, usually a database administrator, to manage and monitor the use of an ArcSDE service. This appendix describes these commands in detail, providing both the syntax and example usage.

Other administration commands allow users to create and modify the schema of an ArcSDE database. Those commands are not documented here but can be found in the ArcSDE Developer's Help.

Command listing

The following commands are used to manage the service. Other commands, such as those used to convert data and create and maintain schema, are documented in the ArcSDE Developer Help.

ArcSDE command	Description
sdemon	Manages the ArcSDE server
sdeservice	Manages the ArcSDE service on Windows NT platforms
sdesetupdb2	Initial setup program for ArcSDE for DB2 UDB
sdesetupinfx	Initial setup program for ArcSDE for Informix
sdesetupora80	Initial setup program for ArcSDE for Oracle8
sdesetupora8i	Initial setup program for ArcSDE for Oracle8*i*
sdesetupssql	Initial setup program for ArcSDE for SQL Server

Command syntax

The administration commands use UNIX command syntax and notation according to the following conventions:

-letter	Specifies a command operation or option, for example, "-o" or "-a". Letters are both lowercase and uppercase.
<>	Required argument. Replace appropriately. For example, "-u <DB_User_name>" could become "-u av".
|	Mutually exclusive arguments. Pick one from the list.
{ }	Used with " | " to specify a list of choices for an argument.
[ ]	Optional parameter.

Each command has "operations" and "options". An operation does a specific task related to the command and is specified with "-o <task>". For example, nsome of the sdetable command operations are:

```
sdemon  -o status
sdemon  -o start
sdemon  -o shutdown
sdemon  -o pause
sdemon  -o resume
sdemon  -o info
sdemon  -o kill
```

Each operation has a set of options. Just like operations, options are specified by "-<letter>". The "-<letter>" used for a particular option is standard across all commands. For instance, the option to specify a service is always "-i". Sometimes a letter is used for two different types of options, but never in the same command.

For each command's operation, there are mandatory and nonmandatory options.

```
sdemon  -o status [[-i <service>] [-s
<server_name>] | [-H <sde_directory>]] [-q]
```

The example above has two options. Anything enclosed within "[ ]" isn't required. The status operation is required, while service "[i]" and quiet "[q]" aren't. Sometimes an option marked optional is not truly optional. The most common occurrence is "[-p <DB_User_password]". It is optional on the command line, but ArcSDE will query for the password if it is not given.

```
$ sdesetupora8i  -o  install

Password:
```

UNIX users should be careful with special characters such as "?". Depending on the UNIX shell version, you may need to use the appropriate quote character to use a special character. For example, you can use the "-?" operation with any administration command to get help. If you're using a C shell (rather than a Bourne shell), you must use "\", which tells the shell to use the next character directly rather than as a special character.

Therefore, in a C shell, you must use:

```
$ sdemon  -\?
```

In a Bourne shell, you can simply use:

```
$ sdemon  -?
```

Some commands include optional Structured Query Language (SQL) query statements, or "where clauses", to limit the features retrieved from a table or log file. If this option is included, the query must be quoted (for example, "area < 1000"). If your database management system (DBMS) encloses character literals with single quotes, enclose the entire expression with double quotes (for example, "state_code = 'CO' "). If your DBMS encloses character literals with double quotes, enclose the entire expression with single quotes, for example, 'state_code = "CO"'.

Getting help

You can list the usage, operations, and options for any ArcSDE administration command with the "-h" or "-?" options.

```
sdemon  -h

sdemon -o status [[-i <service>]
       [-s <server_name>] |
       [-H <sde_directory>]] [-q]

sdemon -o start [[-i <service>]
       [-s <server_name>] |
       [-H <sde_directory>]]
       [-p <DB_Admin_password>]

sdemon -o shutdown [[-i <service>]
       [-s <server_name>] | [-H <sde_directory>]]
       [-p <DB_Admin_password>]  [-N]

sdemon -o pause [[-i <service>]
       [-s <server_name>] | [-H <sde_directory>]]
       [-p <DB_Admin_password>]

sdemon -o resume [[-i <service>]
       [-s <server_name>] | [-H <sde_directory>]]
       [-p <DB_Admin_password>]

sdemon -o info -I <{users | config | stats |
       locks | vars }>  [-q]
       [[-i <service>] [-s <server_name>] |
       [-H <sde_directory>]]

sdemon -o kill [[-i <service>] [-s <server_name>]
       | [-H <sde_directory>]]
       [-p <DB_Admin_password>] -t <{ all | pid }>
       [-N]

Operations:

       status          Report instance status.
       start           Start the instance.
       shutdown        Shutdown the instance.
       pause           Stop the instance from
                       accepting further
                       connections.
       resume          Let the instance accept
                       connections again.
       info            Display requested
                       information about the
                       instance.
       kill            Kill all or a specified
                       connection to the instance.

Options:

       -o    Operation
       -s    Specify Server other than localhost
       -t    Kill target, either the pid of an
             ArcSDE server, or "all"
       -p    ArcSDE DBA DBMS password
       -H    ArcSDE home directory for instance to
             operate on
       -i    ArcSDE instance name to operate on
             (not allowed for start)
       -I    The requested information type
       -q    Quiet
       -N    No Verification
       -?    Print Options
       -h    Print Options
```

ArcSDE administration commands: sdemon

This command is used to monitor and manage the ArcSDE service. You can use sdemon to start up, pause, resume, and shut down all connection activity and display current configuration parameters and server task information. Individual server tasks can also be managed.

Security

Current execution privileges are granted to the root or the ArcSDE administrator for the start operation only. All other operations may be executed by any user who knows the password.

Usage syntax

```
sdemon    -o status [-i <service>]
          [-H <sde_directory>][-s <server_name>]
          [-q]
sdemon    -o start [-p <DB_Admin_password>]
          [-i <service>] [-H <sde_directory>]
          [-s <server_name>]
sdemon    -o shutdown [-p <DB_Admin_password>]
          [-i <service>] [-H <sde_directory>]
          [-s <server_name>] [-N]
sdemon    -o pause [-p <DB_Admin_password>]
          [-i <service>] [-H <sde_directory>]
          [-s <server_name>]
sdemon    -o resume [-p <DB_Admin_password>]
          [-i <service>] [-H <sde_directory>]
          [-s <server_name>]
sdemon    -o info -I <{users | config | stats |
          locks | vars}> [-i <service>]
          [-H <sde_directory>][-s <server_name>]
          [-q]
sdemon    -o kill [-p <DB_Admin_password>]
          -t <{ all | pid }> [-i <service>]
          [-H <sde_directory>] [-s <server_name>]
          [-N]
sdemon    -h
sdemon    -?
```

Operations

status	Reports the service status.
start	Starts the ArcSDE service if it's not running. Only the ArcSDE administrator or root can use this operation.
shutdown	Shuts down the ArcSDE service immediately if no server tasks are running. If server tasks are running, you're prompted to remove the running tasks before the shutdown takes place. If you use the "-N" option when shutting down, all server tasks stop and the system shuts down immediately.
pause	Disallows further client connection requests to be processed. No client tasks are allowed to connect to ArcSDE servers through this ArcSDE service.
resume	Allows client connection requests to be processed again. Client tasks are allowed to connect to ArcSDE servers through this ArcSDE service.
info	Displays information about users, configuration, statistics, locks, or environment variables.
kill	Kills all or a specified connection to the service.

Options

-h	Prints usage and options.
-H	ArcSDE home directory (SDEHOME).
-i	ArcSDE service name (not applicable for "start" option).
-I	Inquire about configuration, locks, statistics, users, or environment variables. config Displays current configuration variables. locks Displays lock information about processes that are holding locks. stats Displays process statistics for each ArcSDE client/server connection. users Lists users' connections to ArcSDE and associated process identifiers. vars Displays ArcSDE service environment variables.
-N	No verification is performed—the operation begins immediately after being invoked.
-o	Operation.
-p	ArcSDE administrator DBMS password.
-q	Quiet; all titles and warnings are suppressed.
-s	ArcSDE server hostname (default: localhost).
-t	Kills server tasks: all Forcefully removes all server tasks. pid Removes the task identified by the process identifier.
-?	Prints usage and options (use "-\?" in C shell).

ArcSDE administration commands: sdeservice (Windows NT only)

This command manages the ArcSDE service on Windows NT.

Usage syntax

```
sdeservice -h

sdeservice -o create -p <DB_Admin_password>
           -l <license_server_name> [-q]
           [-H <sde_directory>]
           [-d <{ORACLE,SID |ORACLE8I,SID |
           SQLSERVER | DB2 | JET}>]
           [-i <service>] [-u <service_user>]
           [-P <service_user_password>]

sdeservice -o delete [-i <service>] [-a] [-q]

sdeservice -o register -r <{SDEHOME |
           SDE_DBA_PASSWORD | NLS_LANG}>
           -v <value> -p <DB_Admin_password>
           [-i <service>] [-q]

sdeservice -o unregister -r <{SDEHOME |
           SDE_DBA_PASSWORD | NLS_LANG}>
           -p <DB_Admin_password>[-i <service>]
           [- q]

sdeservice -o modify -r <{SDEHOME |
           SDE_DBA_PASSWORD |
           LICENSE_SERVER | NLS_LANG}>
           -p <old_DB_Admin_password>
           -v <new_value> [-i <service>] [-q]

sdeservice -o list [-i <service>] [-a] [-q]
```

Operations

create	Creates a service.
delete	Deletes a service.
register	Registers a service.
unregister	Unregisters a service.
modify	Modifies a service.
list	Displays service information for all or a specified service.

Options

-a	For the delete operation, delete all services. For the list operation, list the information of all services.
-d	A DBMS whose service should start before ArcSDE. Optional if the DBMS is on a remote machine. This must be in uppercase.
-h	Prints usage and options.
-H	ArcSDE home directory (SDEHOME). Only needed if the SDEHOME variable isn't set or multiple services are in use.
-i	ArcSDE service name—only required if the service is not called "esri_sde".
-l	License server host computer name.
-N	No verification is performed—the operation begins immediately after being invoked.
-o	Operation.
-p	ArcSDE administrator DBMS password.
-P	ArcSDE service user password (Windows NT login password).
-q	Quiet; all titles and warnings are suppressed.
-r	Windows NT registry keyword.

Options	(continued)
-u	ArcSDE service account user—must be a Windows NT user who has "administrator" permissions on the server computer. Include the domain name if needed. For example, if you're logged into the "AVWORLD" domain and your username is 'joe', enter "AVWORLD\joe". You should be logged in as this user when you create the service.
-v	Registry value.
-?	Prints usage and options (use "-\?" on C shell).

Discussion

The sdeservice administration utility manages the ArcSDE services and registry entries on Windows NT platforms. Installing the ArcSDE software adds one service and several related registry entries, which include SDE_DBA_PASSWORD, SDEHOME, LICENSE_SERVER, and NLS_LANG.

The create and delete operations will add or delete the ArcSDE service entry, respectively. You can modify the registry values of SDEHOME, SDE_DBA_PASSWORD, LICENSE_SERVER, or NLS_LANG with the modify operation. You can also remove or re-create the SDEHOME or SDE_DBA_PASSWORD entries with the unregister and register operations. Normally, you'll only need to use the register operation after using unregister.

ArcSDE administration commands: sdesetup

This command administers business tables and their data.

Usage syntax

```
sdesetup* -h
sdesetup* -?
sdesetup* -0 upgrade [-H <sde_directory>]
          [-p <DB_Admin_password>] [-N] [-q]

sdesetup* -o list [-H <sde_directory>]
          [-p <DB_Admin_password>] [-q]

sdesetup* -o install [-H <sde_directory>]
          [-p <DB_Admin_password>] [-N] [-q]

sdesetup* -o remove [-H <sde_directory>]
          [-p <DB_Admin_password>] [-N] [-q]
```

Operations

install	Creates or updates the ArcSDE and Geodatabase system tables and stored procedures.
list	Lists the installed ArcSDE version.
remove	Removes the ArcSDE and Geodatabase system tables and stored procedures.
upgrade	Upgrades the ArcSDE and Geodatabase system tables and stored procedures.

Options

-h	Prints usage and options.
-H	ArcSDE home directory (SDEHOME). Only needed if the SDEHOME variable isn't set or multiple services are in use.
-N	No verification is performed—the operation begins immediately after being invoked.
-o	Operation.
-p	ArcSDE administrator DBMS password.
-q	Quiet—all titles and warnings are suppressed.
-?	Prints usage and options (use "-\?" on C shell).

Discussion

The sdesetup administration utility manages the creation and maintenance of the ArcSDE and geodatabase metadata, which primarily includes system tables, indexes, stored procedures, and some triggers. The actual name of the command varies according to the implementation of ArcSDE as follows:

ArcSDE for Oracle8*i*	sdesetupora81
ArcSDE for Oracle8	sdesetupora80
ArcSDE for Informix	sdesetupinfx
ArcSDE for SQL Server	sdesetupssql
ArcSDE for DB2 UDB	sdesetupdb2

The ArcSDE system tables and stored procedures can be created or updated using the install operation. For new installations of ArcSDE, the install operation will create the tables and stored procedures for the first time. For subsequent executions of the

install operation on an existing database, the metadata is checked to ensure that it is current. If it is not, the install operation will bring it up-to-date.

sdesetupora8i -o install -p bugaboo

Use the update operation to bring an existing database up-to-date with the latest additions or changes to a new ArcSDE installation.

sdesetupssql -o upgrade -p bugaboo

The list operation returns the current version of an ArcSDE installation.

sdesetupinfx -o list -p bugaboo

The remove operation will remove all tables and stored procedures from the database. As a safety precaution, this operation can only be run after the ArcSDE administrator has unregistered all tables referenced in the TABLE_REGISTRY table and deleted all feature classes referenced in the LAYERS table. The operation will not succeed while tables remain registered or feature classes exist.

sdesetupdb2 -o remove -p bugaboo

ArcSDE initialization parameters

E

IN THIS APPENDIX

- **ArcSDE service initialization parameters**

The ArcSDE service initialization parameters, set in the SDEHOME\etc\giomgr.defs file, provide the functionality for the ArcSDE administrator to customize individual components of the service initialization process. The operation of and default settings for each parameter are described in this appendix.

ArcSDE service initialization parameters

Parameter	Description
ATTRBUFSIZE	The size of the attribute array buffer. Defaults to 50,000.
AUTOCOMMIT	The implicit automatic commit rate within a transaction. If AUTOCOMMIT is set to zero, the transaction will commit only if the application issues an explicit commit. If it is set to a number greater than zero, the operation will commit after the number of updates specified by AUTOCOMMIT have occurred. This feature prevents transactions from becoming too large and exceeding the database management system (DBMS) logs. Defaults to 1,000.
BLOBBUFSIZE	Obsolete at ArcSDE 8.0.2.
BLOBMEM	When binary large objects (BLOBs) are stored, the server must accumulate the BLOB chunks the application sends over the network. If the BLOB size is greater than BLOBMEM, the server writes the BLOB data to a disk file before storing it in the database. If the BLOB size is less than BLOBMEM, the server accumulates the BLOB in memory. If BLOBMEM is a negative number, the server always uses memory, regardless of the BLOB size. Defaults to 1,000,000.
CONNECTIONS	The maximum number of simultaneous connections. Defaults to 64.

Parameter	Description (continued)
DISSMEM	Obsolete at ArcSDE 8.0.2.
LAYERS	The maximum number of feature classes. Defaults to 500.
LOCKS	The maximum number of feature class locks. Defaults to 10,000.
LOGS	Obsolete at ArcSDE 8.0.2.
MAXARRAYBYTES	The maximum number of array bytes allocated per stream. Defaults to 550,000.
MAXARRAYSIZE	The maximum array fetch size. Defaults to 100.
MAXBLOBSIZE	The maximum size of user-defined BLOBs in bytes. Defaults to 1,000,000.
MAXBUFSIZE	The maximum buffer threshold. The minimum value is 12,288. If the MAXBUFSIZE value is greater than the minimum value but less than the MINBUFSIZE, the two values are switched. Defaults to 65,536.
MAXDISTINCT	This parameter controls the maximum number of distinct values returned by an SE_DISTINCT_STATS statistic in a call to SE_table_calculate_stats or SE_stream_calculate_table_statistics. A value of zero means an unlimited number of distinct values can be returned. Defaults to 512.

Parameter	Description (continued)
MAXINITIALFEATS	The maximum number of features allowed in the initial features argument of the sdelayer administration tool and the SE_layer_create function. This parameter prevents the inadvertent creation of excessively large initial extents for the table of a feature class. This is an ArcSDE for Oracle parameter only. Defaults to 10,000.
MAXSTREAMS	The maximum number of streams allowed by the server. Defaults to eight.
MAXTABLELOCKS	The maximum number of table locks. Defaults to 10,000.
MAXTIMEDIFF	Specified in seconds, the maximum time difference allowed between the server machine and a client machine. Set this parameter to -1 to disable it. Defaults to 1,800.
MINBUFOBJECTS	The minimum number of buffer objects. Defaults to 512.
MINBUFSIZE	The minimum buffer threshold. The minimum value is 4,096. Defaults to 16,384.
OBJECTLOCKS	The maximum number of object locks. Defaults to 10,000.

Parameter	Description (continued)
RASTERBUFSIZE	The size of the raster buffer specified in bytes. This value must be large enough to store the largest raster tile accessed. Defaults to 102,400.
RASTERCOLUMNS	The maximum number of raster columns the ArcSDE service will maintain. Defaults to 500.
READONLY	When set to TRUE, the ArcSDE service will not allow edits to be performed by ArcSDE clients. The default setting of FALSE allows editing.
REGISTRATIONS	The maximum number of registered tables. Defaults to 1,000.
SHAPEPTSBUFSIZE	The size of the shape POINTS array buffer. The default value is 400,000, which is calculated for a 2D area feature with 500 points.
SHAPEBUFSIZE	Obsolete in ArcSDE 8.0.2.
SPINDEXBUFSIZE	Obsolete in ArcSDE 8.0.2.
STATECACHING	Set to TRUE, the current set state for each stream is maintained in the ArcSDE server's memory. Set to FALSE, the state is reread from disk for each stream operation. Defaults to TRUE.
STATELOCKS	The maximum number of state locks. Defaults to 10,000.

Parameter	Description (continued)
STREAMPOOLSIZE	The maximum number of allocated stream resources added to the stream pool. Until this value is exceeded, the resources of released streams are not deallocated, but are added to the stream pool. The resources of the stream pool are reused whenever new streams are created. If the stream pool is full when a stream is released, its resources are deallocated. Defaults to three.
TCPKEEPALIVE	Setting TCPKEEPALIVE to TRUE will allow the ArcSDE service to use the current system Transmission Control Protocol/Internet Protocol (TCP/IP) KEEPALIVE settings. ArcSDE servers will then be able to detect clients whose machines have crashed or have been deliberately terminated by the Windows NT Task Manager or the UNIX kill command. TCPKEEPALIVE set to TRUE turns on a form of probing, where, after two hours of idle time (usually), a packet is sent on the idle connection to see if there is anything on the other end. Be aware that if TCPKEEPALIVE is set to TRUE, a disconnection can be triggered by short-term network outages (~10 minutes). By default, TCPKEEPALIVE is set to FALSE.

Parameter	Description (continued)
TEMP	The temporary file directory. Defaults to /tmp.
TIMEOUT	Obsolete at ArcSDE 8.0.2.

Glossary

Abstract Data Types (ADT)

Spatial data types defined by the Open GIS Consortium and documented in the OpenGIS-RFP1 (SimpleFeatures) document. They include point, line string, polygon, multipoint, multilinestring, multipolygon, and geometry. This provides a standard feature-based abstraction of real-world phenomenon. This vector data consists of geometric and topological primitives used, separately or in combination, to construct objects that express the spatial characteristics of geographic features.

ANSI SQL 89

Industry standard language used in querying, updating, and managing relational databases. SQL can be used to retrieve, sort, and filter specific data to be extracted from the database. ANSI (American National Standards Institute) adopted SQL as the standard language for database management systems (DBMSs). See DBMS.

API

Application programming interface. Refers to a defined and documented set of tools or "functions" that application developers use to build or customize a program or set of programs. APIs can be built for programming languages such as C, COM, Java, and so on.

ArcCatalog

Spatial data browser application that facilitates data management and data access.

ArcIMS

Application for creating, designing, and managing Web sites with mapping and geographic information system (GIS) capabilities.

ArcMap

Spatial data display, editing, query, and analysis application.

ArcSDE

ArcSDE is a gateway to a multiuser commercial DBMS—for example, Oracle, Microsoft SQL Server, Informix, and DB2. ArcSDE is an open, high-performance spatial data server that employs client/server architecture to perform efficient spatial operations and manage large, shared geographic data. See also client/server.

ArcSDE data types

ArcSDE implementation of native DBMS data types: smallint (small integer), integer, float, double, string, BLOB, date.

ArcSDE for Coverages

An ArcSDE server that provides read-only access to ArcInfo coverages, shapefiles, ArcStorm™ library layers, and ArcInfo LIBRARIAN™ layers. Uses the same data transfer technology as ArcSDE for DBMS products.

ArcStorm

ArcStorm (Arc Storage Manager) is a data storage facility and transaction manager for ArcInfo file-based coverage data. ArcStorm manages a feature-oriented database that can be closely integrated with a DBMS. An ArcStorm database is a collection of libraries, layers, INFO tables, and external DBMS tables. Data stored in an ArcStorm database benefits from the transaction management and data archiving capabilities of ArcStorm.

ArcView GIS

Easy-to-use desktop GIS for exploring, analyzing, displaying, and querying geographic data.

attribute

1. A characteristic of a geographic feature described by numbers, characters, images, and CAD drawings, typically stored in tabular format and linked to the feature. For example, the attributes of a well might include depth and gallons per minute.
2. A column in a database table. See column.

BLOB

Binary large object. The binary data type of a column in a DBMS table that stores large image, text, or geometry data as attributes.

CAD

Computer-aided design. An automated system for the design, drafting, and display of graphically oriented information.

clean

An ArcInfo command that builds coverage polygon and arc–node topology by performing a geometric analysis on arcs and label points to identify coverage nodes and polygons.

client/server

A software system is said to have a client/server architecture when there is a central process (server) that accepts requests from multiple user processes (clients). The architecture enables the separation of local client processing from the server that manages the databases, access, and data integrity. ArcSDE is one example of a client/server architecture.

column

The vertical dimension of a table that holds attribute values. A column has a name and a data type applied to all values in the column. A table has rows and columns. See table and row.

coordinate

A set of numbers that designate location in a given reference system such as x,y in a planar coordinate system or x,y,z in a three-dimensional coordinate system. Coordinates represent locations on the earth's surface relative to other locations.

coordinate system

1. A reference system used to measure horizontal and vertical distances on a planimetric map. A coordinate system is usually defined by a map projection, a spheroid of reference, a datum, one or more standard parallels, a central meridian, and possible shifts in the x- and y-directions to locate x,y positions of point, line, and area features.

2. In ArcInfo, a system with units and characteristics defined by a map projection. A common coordinate system is used to spatially register geographic data for the same area. See spatial reference.

coverage

A file-based vector data storage format for storing the location, shape, and attributes of geographic features. A coverage stores geography as primary features (such as arcs, nodes, polygons, and label points) and secondary features (such as tics, map extent, links, and annotation). Associated feature attribute tables describe and store attributes of the geographic features.

data dictionary

A catalog of all data held in a database or a list of items giving data names and structures. Also referred to as DD/D for data dictionary/directory. Commercial DBMSs have online data dictionaries stored in special tables called system tables. ArcSDE and the geodatabase have data dictionary tables containing information about the spatial data in the database.

data integrity

Maintenance of data values according to data model and data type. For example, to maintain integrity, numeric columns will not accept character data.

data type

The characteristic of columns that defines what types of data values they can store. Examples include character, floating point, and integer.

database

1. A collection of related data organized for efficient retrieval of information.
2. A logical collection of interrelated information managed and stored as a unit, usually on some form of mass storage system such as magnetic tape or disk. A GIS database includes data about the spatial location and shape of geographic features recorded as points, lines, areas, pixels, grid cells, or TINs, as well as their attributes.

database administrator (DBA)

One who manages a database—sets up users, security, backup, and recovery procedures for all data and optimizes physical data storage for best performance.

database connection

A connection to a DBMS server, an ArcSDE application server, or an Object Linking and Embedding database (OLE DB).

database locking

A database management process for maintaining the consistency of the data while supporting simultaneous access by more than one user. A typical technique is to use a system of locking data to prevent data corruption caused by multiple users editing and reading it.

database trigger

Triggers are stored procedures associated with a specific operation on a specific database table. A trigger is automatically fired when the operation with which it is associated is performed on the table. For example, a trigger may be used in conjunction with a database sequence to enforce a primary key constraint. For every insert operation, the trigger ensures that a new sequence number is assigned to the table as a primary key.

dataset

A named collection of logically related data items arranged in a prescribed manner.

DB2 Spatial Extender

One of IBM's DB2 Universal Database Extenders that defines new data types and functions to support the storage and manipulation of spatial data.

DBMS

Database management system. A set of computer programs for organizing the information in a database. A DBMS supports the structuring of the database in a standard format and provides tools for data input, verification, storage, retrieval, query, and manipulation. See RDBMS.

domain

A named constraint in the database. This named constraint can be associated with a field for the subtype of a feature class or table to make an attribute validation rule. Types of attribute domains include range and coded value domains.

dynamic link library (DLL)

A precompiled library file that contains functions and data. A DLL is loaded at run time by its calling modules (EXE or DLL). Also referred to as an in-process server.

entity

A collection of objects (persons, places, things) described by the same attributes. Entities are identified during the conceptual design phase of database and application design.

executable file (.EXE)

A program file created from one or more source code files translated into machine code and linked together. Also referred to as an out-of-process server.

export

A platform-independent data transfer utility—generates a binary interchange file.

feature

1. A vector object in a geodatabase that has a geometry type of point, line, polygon, or annotation. Features are stored in feature classes.
2. A representation of a real-world object in a layer on a map.
3. A point, line, or polygon in a coverage or shapefile.

feature class

1. The conceptual representation of a geographic feature. When referring to geographic features, feature classes include point, line, area, and annotation. In a geodatabase, an object class that stores features and has a field of type geometry. See layer.
2. A classification describing the format of geographic features and supporting data in a coverage. Coverage feature classes for representing geographic features include point, arc, node, route-system, route, section, polygon, and region. One or more coverage features are used to model geographic features; for example, arcs and nodes can be used to model linear features such as street centerlines. The tic, annotation, link, and boundary feature classes provide supporting data for coverage data management and viewing.
3. The collection of all the point, line, or polygon features or annotation in a CAD dataset.

feature dataset

In geodatabases, a collection of feature classes that share the same spatial reference. Because the feature classes share the same spatial reference, they can participate in topological relationships with each other such as in a geometric network. Several feature classes with the same geometry may be stored in the same feature dataset. Object classes and relationship classes can also be stored in a feature dataset.

field

The intersection of a table row and a column—each field contains the values for a single attribute. See attribute.

firewall

A combination of filters and gateways that protect a site's computers from an external unauthorized access or outright attack.

geodatabase

An object-oriented geographic database, or collection of spatial data and related descriptive data, organized for efficient storage and retrieval by many users and hosted inside a DBMS.

Object behavior is implemented using validation rules, relationships, and topological associations.

geographic data

The locations and descriptions of geographic features—the composite of spatial data and descriptive data.

geographic database

A collection of spatial data and related descriptive data organized for efficient storage and retrieval by many users.

Geographic information system

An organized collection of computer hardware, software, geographic data, and personnel designed to efficiently capture, store, update, manipulate, analyze, and display all forms of geographically referenced information.

geolocation

The process of creating features from tabular data by matching the tabular data to a spatial location. One example of geolocation is creating point features from a table of x,y coordinates. Points can also be created by matching addresses to streets.

geometric network

1. A one-dimensional nonplanar graph, or logical network, that is composed of features. These features are constrained to exist within the network and can therefore be considered network features. ArcInfo 8 will automatically maintain the explicit topological relationships between network features in a geometric network.
2. Represents a one-dimensional linear network such as a road system, a utility network, or a hydrologic network. Geometric networks contain feature classes that play a topological role in the network. These feature classes are homogeneous collections of one of these four network feature types: simple junction feature, complex junction feature, simple edge feature, and complex edge feature. More than one feature class can have the same type of network feature.

geometry

The properties, measurement, and relationships of points, lines, angles, surfaces, and solids. In ArcInfo, geometry is used to represent the spatial component of geographic features.

grid

A geographic data model representing information as an array of equally sized square cells arranged in rows and columns. Each grid cell is referenced by its geographic x,y location.

image

Represents geographic features by dividing the world into discrete squares called cells. Examples include satellite and aerial photographs, scanned documents, and building photographs.

import

A platform-independent data transfer utility—reads data into an ArcSDE database from a binary interchange file.

index

Special data structure used in a database to facilitate searching for records in tables or spatial features in geographic datasets. Provides faster access to data than doing a full table scan. ArcInfo supports both spatial and attribute indexes.

Informix

A commercial DBMS supported by ArcSDE.

intercept

An ArcSDE facility to capture the TCP/IP-based communication that is passed to and from ArcSDE clients and ArcSDE servers.

layer

1. A collection of similar geographic features—such as rivers, lakes, counties, or cities—of a particular area or place for display on a map. A layer references geographic data stored in a data source, such as a coverage, and defines how to display it. You can create and manage layers as you would any other type of data in your database.
2. A feature class in a shared geodatabase managed with ArcSDE. See feature class.

License Manager

Software that allocates and monitors ArcInfo licenses and their usage.

line

A line connects two or more x,y coordinates. Rivers, roads, and electric and telecommunication networks are all linear features.

local area network (LAN)

A computer data communications technology that connects computers at the same site—for example, all computers in the same building. Computers and terminals on an LAN can freely share data and peripheral devices such as printers and plotters. LANs are composed of cabling hardware and software.

Map LIBRARIAN

A set of software tools to manage and access large geographic datasets in a map library. LIBRARIAN commands create and define a map library, move data in and out of a library, query the data in a map library, and display the results of a query.

MapObjects

A collection of software "building blocks" that can be used by developers to create applications that include GIS and mapping capabilities.

Microsoft Access

A commercial DBMS supported by ArcSDE.

Microsoft SQL Server

A commercial DBMS that is supported with ArcSDE.

network packet

Requests and results transferred between clients and servers are sent in fixed-size chunks of data known as "packets". If an application does bulk-copy operations, or sends or receives large amounts of text or image data, a packet size larger than the default may improve efficiency because it results in fewer network reads and writes. If an application sends and receives small amounts of information, setting the packet size to 512 bytes would be sufficient for most data transfers.

OLE DB provider

Object linking and embedding database provider. OLE DB is an open specification designed to build on the success of ODBC by providing an open standard for accessing all kinds of data. Each provider communicates with and retrieves data from a different database, but you can work with data retrieved by any provider the same way. Typically, they can only retrieve nonspatial data. However, if an OLE DB provider can retrieve geographic data in OpenGIS format, you can work with that data in ArcInfo.

OpenGIS

Open Geodata Interoperability Specification—a series of standards being developed by the Open GIS Consortium (OGC) to support interoperability of GIS systems in a heterogeneous computing environment.

Oracle

A commercial DBMS supported by ArcSDE.

Oracle Spatial Geometry Type

Geometries are stored using a table with a single column of type MDSYS.SDO_GEOMETRY and a single row per geometry instance.

Oracle Spatial Normalized Type

Uses a table with a predefined set of columns of type NUMBER and one or more rows for each geometry instance. In the Normalized Schema 5, tables are used to record all information about the layer.

paging

At the operating system level, this process transfers information from volatile system memory (RAM) to disk and back again. This enables the system to handle more information than it normally could handle in real memory.

point

A single x,y coordinate that represents a single geographic feature such as a telephone pole.

polygon

A two-dimensional feature representing an area such as a state or county.

port number

An entry/exit mechanism that controls and synchronizes the flow of data into and out of the central processing unit (CPU) to external devices. Ports, and their associated numbers, are used in TCP/IP to distinguish between different types of communication between TCP/IP addresses. Communication can be established at different ports to keep conversations separate. Some ports are assigned to particular types of communication. For example, port 80 is for HTTP, and port 443 is for Secure HTTP. The default port number for ArcSDE client/server communication is 5151.

RDBMS

Relational database management system. A database management system with the ability to access data organized in tabular files that can be related to each other by a common field (item). An RDBMS has the capability to recombine the data items from different files, providing powerful tools for data usage. ArcSDE supports several commercial RDBMSs. See DBMS.

regions

A coverage feature class used to represent a spatial feature as one or more polygons. Many regions can be defined in a single

coverage. Regions have attributes that describe the geographic feature they represent. See coverage.

relationship

An association or link between two objects. See also relationship class.

relationship class

Objects in a real-world system often have particular associations with other objects in the database. These kinds of associations between objects in the geodatabase are called relationships. Relationships can exist between spatial objects (features in feature classes), nonspatial objects (rows in a table), or spatial and nonspatial objects. While spatial objects are stored in the geodatabase in feature classes, and nonspatial objects are stored in object classes, relationships are stored in relationship classes.

rollback

Changes (updates, inserts, and deletes) made to the information stored in a database form part of a transaction. Until the user, or application, explicitly acts to make these changes permanent, the changes are pending. Issuing a ROLLBACK command will reverse all pending modifications to restore the database to the state it was prior to the beginning of the transaction. See transaction.

row

A record in an attribute table—the horizontal dimension of a table composed of a set of columns containing one data item each. Also referred to as a tuple.

A horizontal group of cells in a grid or pixels in an image.

schema

1. The structure or design of a database or database object such as a table.
2. The definition of the database. The schema can either be modeled in UML using a CASE tool or defined directly within ArcCatalog using wizard dialog boxes.

SDE

See ArcSDE.

server

A computer on which a service process runs. See service.

service

A computer program that receives a request from a client, processes it to generate results, and returns those results to the client.

shape

The characteristic appearance or visible form of a geographic object. Geographic objects can be represented on a map using one of three basic shapes: points, lines, or polygons.

shapefile

A vector data storage format for storing the location, shape, and attributes of geographic features.

Spatial DataBlade

A database enhancement module introduced by Informix to extend the functionality of the INFORMIX-Universal Server to provide support for spatial data management.

spatial reference

Describes both the projection and spatial domain extent for a feature dataset or feature class in a geodatabase. See coordinate system.

SQL

Structured Query Language. A syntax for defining and manipulating data from a relational database. Developed by IBM in the 1970s, it has become an industry standard for query languages in most DBMSs. See ANSI SQL 89.

table

Information formatted in rows and columns. A set of data elements that has a horizontal dimension (rows) and a vertical dimension (columns) in a DBMS. A table has a specified number of columns but can have any number of rows. See also attribute.

TCP/IP

The Transmission Control Protocol (TCP) is a communication protocol layered above the Internet Protocol (IP). These are low-level communication protocols that allow computers to send and receive data.

topology

1. In geodatabases, relationships between connected features in a geometric network or shared borders between features in a planar topology.
2. In coverages, the spatial relationships between connecting or adjacent features (for example, arcs, nodes, polygons, and points). The topology of an arc includes its from- and to-nodes and its left and right polygons. Topological relationships are built from simple elements into complex elements: points (connected arcs), routes (sets of sections, which are arcs' simplest elements), arcs (sets of connected points), and areas (sets of or portions of arcs). Redundant data (coordinates) is eliminated because an arc array represents a linear feature, part of the boundary of an area feature, or both.

transaction

A logical unit of work conducted in a database environment, comprising one or more SQL statements. Transactions can involve data definition (create an object), data manipulation (update an object), or reading data (select from an object).

UDP (User Datagram Protocol)

UDP is a connectionless datagram transport protocol. Connectionless protocols allow data to be exchanged without setting up a link between processes. Each unit of data, with all the necessary information to route it to the intended destination, is transferred independently of other data packets and can travel over different paths to reach the final destination. Some data packets might be lost in transmission or might arrive out of sequence to other data packets. It is known as a datagram protocol because it is analogous to sending a letter where you don't acknowledge receipt.

URL

Universal Resource Locator—convenient and succinct way to direct people and applications to a file or other electronic resource via a number of different Internet protocols.

version

A version is an alternative representation of the geodatabase that has an owner, a description, and a level of access (private, protected, and public). See version merging and version reconciliation.

version merging

The process of reconciling two versions of a feature dataset into a common version. If conflicting edits have been made in either of the merged versions, these conflicts are resolved, either automatically or by an interactive process.

version reconciliation

The process of updating a version of a dataset with changes made in another version. Using this technique, a version can

remain up to date with changes even if it is within a long transaction lasting many months.

wizard

A tool that leads a user step-by-step through an unusually long, difficult, or complex task such as software installation.

Index

A

Abstract Data Type (ADT)
  defined 127
Administrator 37
ADT (Abstract Data Type)
  defined 127
ANSI SQL 89
  defined 127
API (application programming interface)
  defined 127
  described 61
Application programming interface (API)
  defined 127
  described 61
Applications 17
ArcCatalog
  defined 127
  described 27
ArcIMS
  defined 127
  described 2
ArcMap
  defined 127
  described 27
ArcSDE 83, 99, 113
  C API program 99
  defined 127
  described 2
  properties 5
  SdeServer license 56
ArcSDE administration account 12
ArcSDE administration commands 113
  getting help 116
  mandatory options 115
  nonmandatory options 115
  operations 115
  options 115
  sdemon 117
  sdeservice 114
  sdeservice (Windows NT only) 119
ArcSDE administration commands (continued)
  sdesetup 121
  syntax 115
  UNIX 115
ArcSDE application locks 51
  area locks 51
  object locks 51
  state locks 51
  table locks 51
ArcSDE applications
  C API function 28
ArcSDE CAD Client 2
ArcSDE client 27
  transport buffer 27
ArcSDE client executable images (UNIX only) 81
ArcSDE data dictionary 1, 9, 83
ArcSDE data types
  defined 128
ArcSDE for Coverages
  defined 128
  described 9
ArcSDE home directory 1, 5, 9, 38
  binary executable files 5
  configuration files 5
  dynamic libraries 10
  dynamic link libraries 5
  internationalization code pages 5
ArcSDE intercept 72
  client 72
  SDEINTERCEPT variable 72
  SDEINTERCEPTLOC variable 72
  server 72
ArcSDE License Manager
  install wizard 15
ArcSDE server 27
  transport buffer 27
ArcSDE server process statistics
  listing 53
ArcSDE service 1, 5, 9, 21, 34, 37, 43, 44, 49, 50, 53, 55
  configurations 49
  data flow 6

ArcSDE service (continued)
database 5
dbinit.sde file 22
default name 18
disconnect 34
giomgr process 5. *See also* Giomgr process
initialization parameters 123
ATTRBUFSIZE 124
AUTOCOMMIT 124
BLOBMEM 124
CONNECTIONS 124
LAYERS 124
LOCKS 124
MAXARRAYBYTES 124
MAXARRAYSIZE 124
MAXBLOBSIZE 124
MAXBUFSIZE 124
MAXDISTINCT 124
MAXINITIALFEATS 125
MAXSTREAMS 125
MAXTABLELOCKS 125
MAXTIMEDIFF 125
MINBUFOBJECTS 125
MINBUFSIZE 125
RASTERBUFSIZE 125
RASTERCOLUMNS 125
REGISTRATIONS 125
SHAPEPTSBUFSIZE 125
STATECACHING 125
STATELOCKS 125
STREAMPOOLSIZE 126
TCPKEEPALIVE 126
License Manager 60
lock information
listing 25
lock limit 50
lock table information 50
LOCK TYPE 51
PID 51
MAP LAYER
lock table information 51
ArcSDE service (continued)
memory resources
shared image requirements (UNIX only) 81
static image requirements (UNIX only) 81
multiple 10
operating system resources 6
paused 37, 44
port number 9, 13
preventing new connections 44
problems 56
release 61
resume 44
running 37, 44
SE_layer_create function 125
shutdown 5, 37, 44, 45, 46
states 37
statistics 33, 53
BUF AVG 53
BUFFERS 53
F/BUF 53
memory 34
OPS 53
PARTIAL 53
PID 53
READS 53
SE_DISTINCT_STATS 124
SE_layer_create function 33
SE_stream_calculate_table_statistics function 33, 124
SE_table_calculate_stats function 33, 124
TOT Kbytes 53
WRITES 53
status 49, 50
client/server connection 49
current mode 49
listing 49
number of clients 49
system environment variables 22
TCP/IP service name 9, 13
UNIX 43
ArcSDE service (continued)
UNIX—remote 43
user session information 54
ArcSDE streams 27
array buffer 29
bytes per 32
feature classes 27
ArcSDE system tables 83
GEOMETRY_COLUMNS 85
LAYERS 84
LAYERS table 84
LOCATORS table 88
METADATA table 88
MVTABLES_MODIFIED table 88
RASTER_COLUMNS 86
RECONCILED_STATES 88
SPATIAL_REFERENCES 86
STATES 87
TABLE_REGISTRY 86
VERSION 84
VERSIONS 87
ArcSDE user database tables 99
business tables 100
feature table (F<Layer_ID>) 101
shape geometry 101
log file tables 108
SDE_LOGFILE_DATA table 108
SDE_LOGFILES table 108
logical network tables 105
N_*_DESC 105
N_*_E* 107
N_*_EDESC 105
N_*_ESTATUS 105
N_*_ETOPO 105
N_*_FLODIR 105
N_*_J* 107
N_*_JDESC 106
N_*_JSTATUS 106
N_*_JTOPO 106
n_*_JTOPO2 106
N_*_PROPS 106
raster tables 110

ArcSDE user database tables (continued)
spatial index table (S<Layer_ID>) 102
version tables 109
A<Registration_ID> 109
D<Registration_ID> 109
ArcSDE user process 47
large transaction 47
rolled back 47
multiple 48
removing 48
terminating 47
ArcStorm
defined 128
ArcView GIS
defined 128
described 2
Array buffer 29
I/O 29
Array fetch 29
ATTRBUFSIZE 29, 32. *See also* Giomgr.defs file: initialization parameters
Attribute
defined 128
described 32
AUTOCOMMIT. *See* Giomgr.defs file: initialization parameters

B

Binary large object (BLOB) 32, 124
defined 128
described 26
Binary schema implementation 100
BLOB (binary large object) 32, 124
defined 128
described 26
Business tables
trigger 100

C

CAD (computer-aided design)
defined 128
described 2
Clean
defined 128
Client/server 49
defined 128
described 5
Column
defined 128
described 101
feature tables 101
spatial index tables 102
Computer-aided design (CAD)
defined 128
described 2
Configuration files 10, 17
CONNECTIONS. *See* Session parameters
Coordinate
defined 128
described 101
shape storage format 101
Coordinate system
defined 128
Coverage
defined 129

D

Data dictionary 9, 12
defined 129
described 83
Data integrity
defined 129
described 84
Data type
defined 129
Database 2, 11
defined 129
described 5
Database administrator (DBA) 113
defined 129
described 33
Database connection
defined 129
described 5
Database locking
defined 129
described 33
Database management system (DBMS)
defined 130
Database trigger
defined 129
described 100
Dataset 83
defined 129
described 98
DB2 11
DB2 Spatial Extender
defined 130
DBA (database administrator) 113
defined 129
described 33
DBA password 18
changing 119
Dbinit.sde file 11, 23, 43, 56, 69
commands
set 24
unset 24
database connection 43
dynamic library path 43
License Manager 43
SDETEMP variable 23
SDEVERBOSE variable 23
DBMS (database management system)
defined 130
Direct connection
described 2
DLL (dynamic link library)
defined 130

Domain
defined 130
described 92
Dynamic link library (DLL)
defined 130

E

Entity
defined 130
Entity types 29, 103
Error log files 71
giomgr.log 71
sde.errlog 71
Error messages 33
-9 SE_INVALID_USER 62
SE_OUT_OF_LOCKS 33, 34
SE_RASTERBUFFER_TOO_SMALL 35
SE_TOO_MANY_DISTINCTS 33
SE_TOO_MANY_RASTERCOLUMNS 35
SE_TOO_MANY_STREAMS 34
Executable file (.EXE)
defined 130
Export
defined 130

F

Feature
defined 130
Feature class 83
defined 130
Feature dataset 83, 94
defined 130
described 93
Feature geometry 29
feature metadata 29
entity type 29
feature ID 29
number of points 29
Feature geometry (continued)
point data 29
x-value 30
y-value 30
Feature tables 101
Field
defined 131
described 93
Firewall
defined 131

G

Geodatabase
defined 131
described 83
Geodatabase system tables 83, 91
GDB_ANNOSYMBOLS 91
GDB_ATTRRULES 91
GDB_CODEDDOMAINS 91
GDB_DEFAULTVALUES 91
GDB_DOMAINS 92
GDB_EDGECONNRULES 92
GDB_FEATURECLASSES 93
GDB_FEATUREDATASET 93
GDB_FIELDINFO 93
GDB_GEOMNETWORKS 94
GDB_JNCONNRULES 94
GDB_NETCLASSES 94
GDB_NETWEIGHTASSOCS 95
GDB_NETWEIGHTS 95
GDB_NETWORKS 95
GDB_OBJECTCLASSES 96
GDB_RANGEDOMAINS 96
GDB_RELCLASSES 96
GDB_RELEASE 97
GDB_RELRULES 97
GDB_SPATIALRULES 97
GDB_STRINGDOMAINS 97
GDB_SUBTYPES 98
Geodatabase system tables (continued)
GDB_USERMETADATA 98
GDB_VALIDRULES 98
Geographic data
defined 131
Geographic database
defined 131
Geographic information system (GIS)
defined 131
described 2
Geolocation
defined 131
Geometric network
defined 131
described 94
Geometry
defined 131
described 103
Geometry table
columns 103
element sequence 103
GID (Geometry ID) 103
relationship to business table 103
sequence 103
shape types 103
storing shapes in 103
Giomgr executable file 56
$SDEHOME\bin 56
%SDEHOME%\lib 56
Giomgr process 5, 45, 57, 61
connection requests 61
MAXTIMEDIFF 61. *See also* Giomgr.defs file: initialization parameters
SDEATTEMPTS variable 61
data dictionary 5
gsrvr process 5, 61
listening state 61
port number 5
process ID (PID) 45
TCP/IP service name 5

Giomgr.defs file 23, 26, 32, 50, 123
initialization parameters 26
ATTRBUFSIZE 124
AUTOCOMMIT 124
BLOBMEM 26, 124
CONNECTIONS 124
LAYERS 124
LOCKS 124
MAXARRAYBYTES 124
MAXARRAYSIZE 124
MAXBLOBSIZE 124
MAXBUFSIZE 124
MAXDISTINCT 124
MAXINITIALFEATS 125
MAXSTREAMS 125
MAXTABLELOCKS 125
MAXTIMEDIFF 125
MINBUFOBJECTS 125
MINBUFSIZE 125
RASTERBUFSIZE 125
RASTERCOLUMNS 125
REGISTRATIONS 125
SHAPEPTSBUFSIZE 125
STATECACHING 125
STATELOCKS 125
STREAMPOOLSIZE 126
TCPKEEPALIVE 126
Giomgr.log file 71
GIS (geographic information system)
defined 131
described 2
Grid
defined 131
Gsrvr process 5, 62
application connection 5
RDBMS 62
log files 62. *See also* ArcSDE user database tables
SDE_LOGFILE_DATA table 62, 63
SDE_LOGFILES table 62, 63

I

IBM DB2 11
Image
defined 131
described 35
Images
RDBMS BLOB column 35
Import
defined 131
Index
defined 132
Informix
defined 132
described 11
Initialization parameters 123
Installing ArcSDE
VERSION table 84
Instances
missing name error 18
Intercept
defined 132
described 72

L

LAN (local area network)
defined 132
described 16
Layer 33
defined 132
feature classes 33
License Manager 58, 64
defined 132
described 15
LICENSE_SERVER 120
Line
defined 132
Local area network (LAN)
defined 132
described 16
Locking 33
LOCKS parameter 33. *See also* Giomgr.defs file: initialization parameters
Log file tables 108
Logical network tables 105

M

Map LIBRARIAN
defined 132
MapObjects
defined 132
described 2
MAXARRAYBYTES 29. *See also* Giomgr.defs file: initialization parameters
MAXARRAYSIZE 29, 30, 32. *See also* Giomgr.defs file: initialization parameters
MAXBUFSIZE 27. *See also* Giomgr.defs file: initialization parameters
MAXDISTINCT 33. *See also* Giomgr.defs file: initialization parameters
Maximum time difference 35
MAXINITIALFEATS 33. *See also* Giomgr.defs file: initialization parameters
MAXSTREAMS 34. *See also* Giomgr.defs file: initialization parameters
MAXTABLELOCKS 33. *See also* Giomgr.defs file: initialization parameters
MAXTIMEDIFF 61. *See also* Giomgr.defs file: initialization parameters
Microsoft Access
defined 132
described 29
Microsoft SQL Server
defined 132
described 11

MINBUFOBJECTS 28. *See also* Giomgr.defs file: initialization parameters
MINBUFSIZE 28. *See also* Giomgr.defs file: initialization parameters

N

Network packet
  ArcSDE connection string 35
  defined 132
  described 35
Network port number 18
Normalized schema 103
  business table 103
  geometry table 103

O

OLE DB provider
  defined 133
  described 69
OpenGIS
  defined 133
  described 104
OpenGIS RFP-1 104
  UDT (user-defined type) 104
Operating system services file 57
  UNIX 20
    /etc/services. 20
    NIS services file 19, 20
  Windows NT 20
    \winnt\system32\drivers\etc 20
Oracle 33
  defined 133
Oracle Spatial Geometry Type
  defined 133
Oracle Spatial Normalized Type
  defined 133

P

Paging
  defined 133
  described 29
Ping command 40
Point
  defined 133
Polygon
  defined 133
Port number 5, 16
  defined 133
  described 18

R

Raster parameters 35
Raster tables 110
RASTER_COLUMNS 35. *See also* Giomgr.defs file: initialization parameters
RASTERBUFSIZE 35. *See also* Giomgr.defs file: initialization parameters
  raster tile 35
RDBMS (relational database management system) 11, 17, 29, 30, 56
  ArcSDE user 38
  array fetch 29
  array inserts 29
  connection parameters 57
    dbinit.sde file 57
  database 11
  defined. *See* RDBMS (relational database management system): DBMS
  described 2
  error log files 71
  Microsoft Access 29
  server 38
Regions
  defined 133
Registered tables 32
  application logging 32
  versioned feature class or table 32
Relational database management system (RDBMS) 11, 17, 29, 30, 56
  ArcSDE user 38
  array fetch 29
  array inserts 29
  connection parameters 57
    dbinit.sde file 57
  database 11
  defined 133
  described 2
  error log files 71
  Microsoft Access 29
Relationship
  defined 134
  described 97
Relationship class
  defined 134
  described 96
Rollback
  defined 134. *See also* Transaction
Row
  defined 134
  described 32

S

Schema 104
  defined 134
  described 84
SDE
  defined 134
Sde.errlog file 69, 70, 71
  gsrvr process 71
SDE_LOGFILE_DATA table 62, 63
SDE_LOGFILES table 62, 63
SDEATTEMPTS 61. *See also* Dbinit.sde file
SDEDBECHO 56. *See also* Dbinit.sde file

SDEHOME 5, 10, 11, 41, 73, 74, 120. *See also* ArcSDE home directory
/bin 74
Description 74, 75
/etc 74
Description 76
/include 74
Description 77
/lib 74
Description 78
/locale 74
Description 79
/ssa 74
Description 79
/sysgen 74
Description 79, 80
/tools 74
Description 80
changing 119
dbinit.sde file 11
SDEHOME%\etc\services.sde 14
Sdelayer command 33
Sdemon command 14, 36, 37, 40, 44, 45, 47, 48, 50, 53, 117
operations 117
options 118
security 117
usage syntax 117
SdeServer license 38, 56, 58, 59
Sdeservice command (Windows NT only) 15, 21, 66, 69, 119
discussion 120, 121
operations 119
options 119, 121
usage syntax 119
Sdetable command 121
operations 121
usage syntax 121
Server
defined 134
described 16
Service
defined 134
described 2
Services.sde file 57
Session parameters 26
CONNECTIONS 26
TCPKEEPALIVE 26
TEMP 26
Shape
defined 134
storing multiple parts 103
Shapefile
defined 134
SHAPEPTSBUFSIZE 29, 30. *See also* Giomgr.defs file: initialization parameters
point data 29
Spatial data 17
described 2
Spatial DataBlade
defined 134
Spatial index table 102
Spatial reference 86
defined 134
Spatial types and functions schema 104
SQL (Structured Query Language)
defined 135
STATECACHING 125. *See also* Giomgr.defs file: initialization parameters
STATELOCKS 33, 34. *See also* Giomgr.defs file: initialization parameters
STREAMPOOLSIZE 34. *See also* Giomgr.defs file: Initialization parameters
Streams 34, 35. *See also* Array buffer
Structured Query Language (SQL)
defined 135
System clock 35. *See also* Giomgr.defs file: initialization parameters: MAXTIMEDIFF

T

Table
defined 135
described 32
RASTER_COLUMNS 35
TCP/IP 57
defined 135
described 18
port 57
port number
Information Sciences Institute 18
service name 57
TCP/IP port
number 18
TCPKEEPALIVE. *See* Session parameters
TEMP. *See* Session parameters
Three-tier architecture
described 2
Topology
defined 135
Transaction
defined 135
described 124
Transport buffer 27. *See also* Streams I/O 27
Two-tier architecture
described 2

U

UDP (User Datagram Protocol)
defined 135
described 43
UDT (user-defined type) 104
Universal Resource Locator (URL)
defined 135
described 16
UNIX 41, 43, 45, 59, 64
\etc\inetd.conf file 43
\etc\services 43
User Datagram Protocol (UDP) 43

ArcSDE client executable image 73, 81
gsrvr.shared 81
gsrvr.static 81
ArcSDE startup problems 64
error messages 64
Bourne shell 59
C Shell 59
inetd daemon 43
kill command 45
License Manager 59
License Manager variables
ESRI_LICENSE_FILE 59
LM_LICENSE_FILE 59
ps command
-ef option 46
-u root option 43
root account 41
SDEHOME 25
SIGHUP 43
URL (Universal Resource Locator)
defined 135
described 16
User Datagram Protocol (UDP)
defined 135
described 43
User-defined type (UDT) 104

V

Version 109
defined 135
described 34
save points 34
Version merging
defined 135
Version reconciliation
defined 135
Version tables 109
Versioned
database 33
states 33
feature class 33

W

Windows NT 13, 39, 40, 44, 57, 58, 66, 67, 68, 69
%SDEHOME%\etc\sde.errlog 69
%SDEHOME%\tools directory 45
killp command 45. *See also* ArcSDE user process
ArcSDE License Manager 58
ArcSDE service 40
remote startup 40
ArcSDE startup problems 66
error messages 66
Control Panel
services menu 39
dbinit.sde file 72
Event Viewer 70
Event Detail menu 70
MS–DOS 14
registry 13, 57, 58, 69
LICENSE_SERVER parameter 58
NLS_LANG parameter 120
SDE_DBA_PASSWORD parameter 120
SDEHOME 25
services 13
C:\winnt\system32\drivers\etc\services file 14
menu 14
user group permissions 68
user permissions 67
Wizard
defined 136
described 15