

Using SYSDBOPEN and SYSDBCLOSE Procedures

To set the initial environment for one or more sessions, create and install the **sysdbopen()** SPL procedure. The typical effect of this procedure is to initialize the properties of a session without requiring the properties to be explicitly defined within the session.

Setting the initial environment for one or more sessions is useful if users access databases through client applications that cannot modify application code or set environment options or environment variables.

The **sysdbopen** procedure is executed whenever users successfully issue the DATABASE or CONNECT statement to explicitly connect to a database where the procedures are installed. (But when a user who is connected to the local database calls a remote UDR or performs a distributed DML operation that references a remote database object by using the *database:object* or *database@server.object* notation, no **sysdbopen** procedure is invoked in the remote database.)

These procedures are exceptions to the general rule that Informix® ignores the name of the owner of a UDR when a routine is invoked in a database that is not ANSI-compliant. For UDRs other than **sysdbopen** and **sysdbclose**, multiple versions of UDRs that have the same SQL identifier but that have different *owner* names cannot be registered in the same database unless the CREATE DATABASE statement that created the database also included the WITH LOG MODE ANSI keywords.

You can also create the **sysdbclose** SPL procedure, which is executed when a user issues the CLOSE DATABASE or DISCONNECT statement to disconnect from the database. If a PUBLIC.**sysdbclose** procedure is registered in the database, and no *user.sysdbclose* procedure is registered for the current user, then the PUBLIC.**sysdbclose** procedure is executed automatically when that *user* disconnects from the database.

You can include valid SQL or SPL language statements that are appropriate when a database is opened or closed. The general restrictions on SQL statements that are valid in SPL procedures also apply to these routines. See the following sections for restrictions on SQL and SPL statements within SPL routines:

- [Subset of SPL Statements Valid in the Statement Block.](#)
- [SQL Statements Valid in SPL Statement Blocks.](#)
- [Restrictions on SPL Routines in Data-Manipulation Statements.](#)

Important: The **sysdbopen** and **sysdbclose** procedures are exceptions to the scope rule for stored procedures. In ordinary UDR procedures, the scope of variables and statements is local. SET PDQPRIORITY and SET ENVIRONMENT statement settings do not persist when these SPL procedures exit. In **sysdbopen** and **sysdbclose** procedures, however, statements that set the session environment remain in effect until another statement resets the options, or the session ends.

For example, the following procedure sets the transaction isolation level to Repeatable Read, and sets the **OPTCOMPIND** environment variable to instruct the query optimizer to prefer nested-loop joins. When a user who owns no *user.sysdbopen* procedure connects to the database, this routine will be executed:

```
CREATE PROCEDURE public.sysdbopen()
  SET ISOLATION TO REPEATABLE READ;
  SET ENVIRONMENT OPTCOMPIND '1';
END PROCEDURE;
```

Procedures do not accept arguments or return values. The **sysdbopen** and **sysdbclose** procedures must be registered in each database in which you want to execute them. The DBA can create the following four categories of **sysdbopen** and **sysdbclose** procedures.

Procedure Name

Description

user.sysdbopen

This procedure is executed when the specified ***user*** opens the database as the current database.

public.sysdbopen

If no ***user.sysdbopen*** procedure applies, this procedure is executed when any user opens the database as the current database. To avoid duplicating SPL code, you can call this procedure from a user-specific procedure.

user.sysdbclose

This procedure is executed when the specified ***user*** closes the database, disconnects from the database server, or the user session ends. If ***user.sysdbclose*** did not exist when the session opened the database, however, the procedure is not executed when the session closes the database.

public.sysdbclose

If no ***user.sysdbclose*** procedure applies, this procedure is executed when the user closes or disconnects from the database server, or when the session ends. If ***public.sysdbopen*** did not exist when the session opened the database, however, the procedure is not executed when the session closes the database.

The database server calls ***user.sysdbclose*** procedure, if it exists in the database, or ***public.sysdbclose*** if this exists and no version owned by ***user*** exists, when the CLOSE DATABASE or DISCONNECT statement explicitly terminates the connection. If the application terminates without issuing the CLOSE DATABASE or DISCONNECT statement, the database server forces an implicit close of the database and executes the ***sysdbclose*** procedure, if a UDR with that name is owned by the user or by PUBLIC.

Make sure that you set file access permissions appropriately to allow intended users to execute the SPL procedure statements. For example, if the SPL procedure executes a command that writes output to a local directory, permissions must be set to allow users to write to this directory. If you want the procedure to continue if permission failures occur, include an ON EXCEPTION error handler for this condition.

For more information about the SQL statements that can appear in SPL routines, and about SPL support for transactions and for roles, see the section [Statement Block](#).

Warning: If a ***sysdbclose*** procedure fails, the failure is ignored. If a ***sysdbopen*** procedure fails, however, the database cannot be opened.

To avoid situations in which a database cannot be opened, take the following precaution while you are writing and debugging a ***sysdbopen*** procedure:

- Set the **IFX_NODBPROC** environment variable before you connect to the database. When **IFX_NODBPROC** is set, the procedure is not executed, and failures cannot prevent the database from opening.

Failures from these procedures can be generated by the system or simulated within the procedures by the RAISE EXCEPTION statement of SPL. If the ***sysdbopen*** routine that is invoked for a user at connection time includes this statement, that user cannot connect to the database. For more information, refer to the description of [RAISE EXCEPTION](#).

For security reasons, non-DBAs cannot prevent execution of these procedures. For some applications, however, such as *ad hoc* query applications, users can execute commands and SQL statements that subsequently change the environment.

A default role defined in the ***sysdbopen*** procedure take precedence over any other role that the user holds when a user establishes a connection to a database in which ***sysdbopen*** successfully specifies a default role for that user.

Any database objects that are created by DDL statements in a ***user.sysdbopen*** or ***user.sysdbclose*** procedure are owned by the connected ***user***, and any object created within **PUBLIC.sysdbopen** or within **PUBLIC.sysdbclose** is owned by the PUBLIC userid, unless the object name is fully qualified by some other owner name when the object name is declared in the DDL statement.

For ANSI-compliant databases, an explicit COMMIT WORK statement is required at the end of the **sysdbopen** or **sysdbclose** definition in the CREATE PROCEDURE statement, to prevent any implicit transactions of SQL statements that the **sysdbopen** or **sysdbclose** procedure executes from being rolled back when the procedure terminates. (Omitting the COMMIT WORK statement does not cause the connection to fail, but does waste resources in opening and then rolling back the transactions.)

For a list of SQL statements that are not valid in these procedures, see [SQL Statements Valid in SPL Statement Blocks](#). For a list of the SPL statements that are valid in these procedures, see [Subset of SPL Statements Valid in the Statement Block](#).


For general information about how to write and install SPL procedures, refer to the section about SPL routines in *IBM Informix Guide to SQL: Tutorial*.

Parent topic: [Session Configuration Procedures](#)

[Send feedback](#) | [Examples exchange](#) | [Troubleshooting](#)

To find the PDF, see [Publications for the IBM Informix 11.70 family of products](#).

For the release notes, documentation notes, and/or machine notes, see the [Release Notes](#) topic.

 Release date: October 2012

http://publib.boulder.ibm.com/infocenter/idshelp/v117/topic/com.ibm.sqls.doc/ids_sq_1799.htm