System i

Programming
DB2 Query Manager and SQL Development Kit for i5/OS commands

Version 6 Release 1

System i

Programming
DB2 Query Manager and SQL Development Kit for i5/OS commands

Version 6 Release 1

Note

Before using this information and the product it supports, be sure to read the information in "Notices," on page 175.

This edition applies to version 6, release 1, modification 0 of IBM DB2 Query Manager and SQL Development Kit for i5/OS (product number 5761-ST1) and to all subsequent releases and modifications until otherwise indicated in new editions. This version does not run on all reduced instruction set computer (RISC) models nor does it run on CICS models.

Contents

Create SQL COBOL Program (CRTSQLCBL)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Create SQL COBOL Program (CRTSQLCBL) command calls the Structured Query Language (SQL) precompiler which precompiles COBOL source containing SQL statements, produces a temporary source member, and then optionally calls the COBOL for System i compiler to compile the program.

If the **Relational database (RDB)** parameter is specified and a program is created, an SQL package will be created at the specified relational database.

Top

Parameters

Keyword	Description	Choices	Notes
PGM	Program	*Qualified object name*	Required, Positional 1
	Qualifier 1: Program	*Name*	
	Qualifier 2: Library	*Name*, ***CURLIB**	
SRCFILE	Source file	*Qualified object name*	Optional, Positional 2
	Qualifier 1: Source file	*Name*, **QLBLSRC**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SRCMBR	Source member	*Name*, ***PGM**	Optional, Positional 3
COMMIT	Commitment control	***CHG**, *ALL, *CS, *NONE, *RR, *UR, *RS, *NC	Optional
RDB	Relational database	*Simple name*, ***LOCAL**, *NONE	Optional
TEXT	Text 'description'	*Character value*, ***SRCMBRTXT**, *BLANK	Optional
USER	RDB user	*Name*, ***CURRENT**	Optional
PASSWORD	RDB user password	*Character value*, ***NONE**, ' '	Optional
OPTION	Precompiler options	Values (up to 18 repetitions): ***NOSRC**, *NOSOURCE, *SRC, *SOURCE, *XREF, *NOXREF, *GEN, *NOGEN, *COMMA, *PERIOD, *JOB, *SYSVAL, *QUOTESQL, *APOSTSQL, *QUOTE, *APOST, *SECLVL, *NOSECLVL, *LSTDBG, *NOLSTDBG, *SQL, *SYS, *NOEXTIND, *EXTIND	Optional
TGTRLS	Target release	*Simple name*, ***CURRENT**, *PRV	Optional
INCFILE	INCLUDE file	*Qualified object name*	Optional
	Qualifier 1: INCLUDE file	*Name*, ***SRCFILE**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
ALWCPYDTA	Allow copy of data	***OPTIMIZE**, *YES, *NO	Optional
CLOSQLCSR	Close SQL cursor	***ENDPGM**, *ENDSQL, *ENDJOB	Optional
ALWBLK	Allow blocking	***ALLREAD**, *NONE, *READ	Optional
DLYPRP	Delay PREPARE	***NO**, *YES	Optional
GENLVL	Severity level	0-40, **10**	Optional
DATFMT	Date format	***JOB**, *USA, *ISO, *EUR, *JIS, *MDY, *DMY, *YMD, *JUL	Optional
DATSEP	Date separator character	***JOB**, '/', '.', ',', '-', ' ', *BLANK	Optional

Keyword	Description	Choices	Notes
TIMFMT	Time format	***HMS**, *USA, *ISO, *EUR, *JIS	Optional
TIMSEP	Time separator character	***JOB**, ':', '.', ',', ' ', *BLANK	Optional
REPLACE	Replace	***YES**, *NO	Optional
RDBCNNMTH	RDB connect method	***DUW**, *RUW	Optional
DFTRDBCOL	Default collection	*Name*, ***NONE**	Optional
DYNDFTCOL	Dynamic default collection	***NO**, *YES	Optional
SQLPKG	Package	*Qualified object name*	Optional
	Qualifier 1: Package	*Name*, ***PGM**	
	Qualifier 2: Library	*Name*, ***PGMLIB**	
SQLPATH	SQL path	Single values: ***NAMING**, *LIBL Other values (up to 268 repetitions): *Name*	Optional
SQLCURRULE	SQL rules	***DB2**, *STD	Optional
SAAFLAG	IBM SQL flagging	***NOFLAG**, *FLAG	Optional
FLAGSTD	ANS flagging	***NONE**, *ANS	Optional
PRTFILE	Print file	*Qualified object name*	Optional
	Qualifier 1: Print file	*Name*, **QSYSPRT**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
USRPRF	User profile	***NAMING**, *USER, *OWNER	Optional
DYNUSRPRF	Dynamic user profile	***USER**, *OWNER	Optional
SRTSEQ	Sort sequence	Single values: ***JOB**, *HEX, *JOBRUN, *LANGIDUNQ, *LANGIDSHR Other values: *Qualified object name*	Optional
	Qualifier 1: Sort sequence	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
LANGID	Language id	*Character value*, ***JOB**, *JOBRUN	Optional
TOSRCFILE	To source file	*Qualified object name*	Optional
	Qualifier 1: To source file	*Name*, **QSQLTEMP**	
	Qualifier 2: Library	*Name*, **QTEMP**, *LIBL, *CURLIB	
DECRESULT	Decimal result options	*Element list*	Optional
	Element 1: Maximum precision	**31**, 63	
	Element 2: Maximum scale	0-63, **31**	
	Element 3: Minimum divide scale	0-9, **0**	
DECFLTRND	Decimal float rounding mode	***HALFEVEN**, *HALFUP, *DOWN, *CEILING, *FLOOR, *HALFDOWN, *UP	Optional
COMPILEOPT	Compiler options	*Character value*, ***NONE**	Optional

Top

Program (PGM)

Specifies the program to be created.

This is a required parameter.

Qualifier 1: Program

name Specify the name of the program to be created.

Qualifier 2: Library

***CURLIB**
The current library for the job is used to locate the compiled program. If no current library entry exists in the library list, QGPL is used.

name Specify the name of the library where the compiled program is located.

Top

Source file (SRCFILE)

Specifies the source file that contains the COBOL source statements and SQL statements.

Qualifier 1: Source file

QLBLSRC
Source file QLBLSRC contains the COBOL source.

name Specify the name of the source file that contains the COBOL source. This source file should have a record length of 92. The source file can be a database file, device file, or an inline data file.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library to be searched.

Top

Source member (SRCMBR)

Specifies the source file member that contains the input source. This parameter is used only if the source file specified for the **Source file (SRCFILE)** parameter is a database file.

***PGM** The source file member that has the same name as the program name specified for the **Program (PGM)** parameter contains the input source.

name Specify the name of the source file member that contains the input source.

Top

Commitment control (COMMIT)

Specifies whether SQL statements in the compiled program are run under commitment control. Files referred to in the host language source are not affected by this option. Only SQL tables, SQL views, SQL packages, SQL sequences, SQL aliases, SQL Types, SQL procedures, SQL functions, SQL indexes, SQL schemas, SQL triggers, and SQL views referred to in SQL statements are affected.

***CHG** *or* ***UR**
Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs can be seen.

***CS** Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT,

LABEL ON, RENAME, and REVOKE statements and the rows updated, deleted, and inserted are locked until the end of the unit of work (transaction). A row that is selected, but not updated, is locked until the next row is selected. Uncommitted changes in other jobs cannot be seen.

ALL** ***or ***RS**
Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows selected, updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs cannot be seen.

NONE** ***or ***NC**
Specifies that commitment control is not used. Uncommitted changes in other jobs can be seen. If the SQL DROP SCHEMA statement is included in the program, *NONE or *NC must be used. If a relational database is specified for the **Relational database (RDB)** parameter, and the relational database is on a system that is not on a System i, *NONE or *NC cannot be specified.

***RR** Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows selected, updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs cannot be seen. All tables referred to in SELECT, UPDATE, DELETE, and INSERT statements are locked exclusively until the end of the unit of work (transaction).

Top

Relational database (RDB)

Specifies the name of the relational database where the SQL package is to be created.

***LOCAL**
The program is created as a distributed SQL program. The SQL statements will access the local database. An SQL package object is not created as part of the precompile process. The Create Structured Query Language Package (CRTSQLPKG) command can be used.

***NONE**
An SQL package object is not created. The program object is not a distributed program and the Create Structured Query Language Package (CRTSQLPKG) command cannot be used.

name Specify the name of the relational database where the new SQL package object is to be created. When the name of the local relational database is specified, the program created is still a distributed SQL program. The SQL statements will access the local database.

Top

Text 'description' (TEXT)

Specifies text that briefly describes the program and its function.

***SRCMBRTXT**
The text is taken from the source file member being used to create the program. If the source file is an inline file or a device file, the text is blank.

***BLANK**
No text is specified.

'description'
Specify no more than 50 characters, enclosed in apostrophes.

Top

RDB user (USER)

Specifies the user name sent to the remote system when starting the conversation. This parameter is valid only when **Relational database (RDB)** is specified.

***CURRENT**
The user name associated with the current job is used.

name Specify the user name to be used for the application server job.

Top

RDB user password (PASSWORD)

Specifies the password to be used on the remote system. This parameter is valid only when **Relational database (RDB)** is specified.

***NONE**
No password is sent. A user name cannot be specified for the **RDB user (USER)** parameter if this value is specified.

Note: Specifying a password of a blank is the same as specifying *NONE.

password
Specify the password of the user name specified for the **RDB user (USER)** parameter.

Top

Precompiler options (OPTION)

Specifies whether the following options are used when the COBOL source is precompiled. If an option is specified more than once, or if two options conflict, the last option specified is used.

Source listing options:

***NOSRC** *or* ***NOSOURCE**
The precompiler does not produce a source printout unless errors are detected during precompile or create package.

***SRC** *or* ***SOURCE**
The precompiler produces a source printout.

Cross reference options:

***NOXREF**
The precompiler does not produce a cross-reference of names.

***XREF** The precompiler produces a cross-reference between items declared in your program and the numbers of the statements in your program that refer to these items.

Host language compiler call options:

***GEN** The precompiler calls the COBOL compiler. If a relational database name is specified for the **Relational database (RDB)** parameter, and the compile is successful, an SQL package is also created.

***NOGEN**
The precompiler does not call the COBOL compiler. Neither a program nor an SQL package is created.

SQL string delimiter options:

***QUOTESQL**
The character used as the string delimiter in the SQL statements is the quote (″).

***APOSTSQL**
The character used as the string delimiter in the SQL statements is the apostrophe (′).

COBOL string delimiter options:

***QUOTE**
The character used for nonnumeric literals and Boolean literals in the COBOL statements is the quote (″).

***APOST**
The character used for nonnumeric literals and Boolean literals in the COBOL statements is the apostrophe (′).

Decimal point options:

***JOB** The representation for the decimal point specified for the job at precompile time is used.

Note: If QDECFMT specifies that the value used as the decimal point is a comma, any numeric constants in lists (such as in the SELECT clause and VALUES clause) must be separated by a comma followed by a blank. For example, VALUES(1,1, 2,23, 4,1) is equivalent to VALUES(1.1,2.23,4.1) where the decimal point is the period.

***SYSVAL**
The value used as the decimal point in numeric constants is from the QDECFMT system value. This value is also used as the decimal point character when casting a numeric value to character.

Note: If QDECFMT specifies that the value used as the decimal point is a comma; any numeric constants in lists (such as in the SELECT clause and VALUES clause) must be separated by a comma followed by a blank. For example, VALUES(1,1, 2,23, 4,1) is equivalent to VALUES(1.1,2.23,4.1) where the decimal point is the period.

***PERIOD**
The value used as the decimal point for numeric constants used in SQL statements is a period. This value is also used as the decimal point character when casting a numeric value to character.

***COMMA**
The value used as the decimal point in numeric constants is a comma. Any numeric constants in lists (such as in the SELECT clause and VALUES clause) must be separated by a comma followed by a blank. For example, VALUES(1,1, 2,23, 4,1) is equivalent to VALUES(1.1,2.23,4.1) where the decimal point is the period.

Naming convention options:

***SYS** Specifies that the system naming convention is used (library-name/file-name).

***SQL** Specifies that the SQL naming convention is used (schema-name.table-name).

If a relational database is specified for the **Relational database (RDB)** parameter, and the database is on a system that is not a System i, *SQL must be specified as the naming convention.

Second-level message text options:

***NOSECLVL**
Second-level text descriptions are not added to the listing.

***SECLVL**
Second-level text with replacement data is added for all messages on the listing.

Debug listing view options:

***NOLSTDBG**
Error and debug information is not generated.

***LSTDBG**
The SQL precompiler generates a listing view and error and debug information required for this view.

Note: You can only use *LSTDBG if you are using the CODE product to compile your program.

Extended indicators options:

***NOEXTIND**
Extended indicator support is not enabled.

***EXTIND**
Extended indicator support is enabled.

Top

Target release (TGTRLS)

Specifies the release of the operating system on which you intend to use the object being created.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

***CURRENT**
The object is to be used on the release of the operating system currently running on your system. The object can also be used on a system with any subsequent release of the operating system installed.

***PRV** The object is to be used on the previous release with modification level 0 of the operating system. The object can also be used on a system with any subsequent release of the operating system installed.

target-release
Specify the release in the format VxRxMx. The object can be used on a system with the specified release or with any subsequent release of the operating system installed.

Top

INCLUDE file (INCFILE)

Specifies the source file that contains members to be included in the program with the SQL INCLUDE statement.

Single values

***SRCFILE**
The qualified source file you specify for the **Source file (SRCFILE)** parameter contains the source file members specified on any SQL INCLUDE statements.

Qualifier 1: INCLUDE file

name Specify the name of the source file that contains the source file members specified on any SQL INCLUDE statements.

The record length of the source file you specify here must be no less than the record length of the source file you specify for the **Source file (SRCFILE)** parameter.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the source file is located.

Top

Allow copy of data (ALWCPYDTA)

Specifies whether a copy of the data can be used in a SELECT statement.

***OPTIMIZE**
The system determines whether to use the data retrieved directly from the database or to use a copy of the data. The decision is based on which method provides the best performance. If the **Commitment control (COMMIT)** parameter is not *NONE, the **Allow blocking (ALWBLK)** parameter should be set to *ALLREAD, when possible, for best performance.

***YES** A copy of the data is used only when necessary.

***NO** A copy of the data is not used. If a temporary copy of the data is required to perform the query, an error message is returned.

Top

Close SQL cursor (CLOSQLCSR)

Specifies when SQL cursors are implicitly closed, SQL prepared statements are implicitly discarded, and LOCK TABLE locks released. SQL cursors are explicitly closed by issuing the CLOSE, COMMIT (without HOLD), ROLLBACK (without HOLD), or CONNECT (Type 1) SQL statements.

***ENDPGM**
The SQL cursors are closed and SQL prepared statements are discarded when the program ends. LOCK TABLE locks are released when the first SQL program on the call stack ends.

***ENDSQL**
The SQL cursors remain open between calls and rows can be fetched without running another SQL OPEN statement. One of the programs higher on the call stack must have run at least one SQL statement. The SQL cursors are closed, SQL prepared statements are discarded, and LOCK TABLE locks are released when the first SQL program on the call stack ends. If you specify *ENDSQL for a program that is the first SQL program called (the first SQL program on the call stack), the program is treated as if *ENDPGM was specified.

***ENDJOB**
SQL cursors remain open between calls and can be fetched without running another SQL OPEN statement. The programs higher on the call stack do not need to have run SQL statements. SQL cursors are left open, SQL prepared statements are preserved, and LOCK TABLE locks are held

when the first SQL program on the call stack ends. SQL cursors are closed, SQL prepared statements are discarded, and LOCK TABLE locks are released when the job ends.

Top

Allow blocking (ALWBLK)

Specifies whether the database manager can use record blocking and the extent to which blocking can be used for read-only cursors.

***ALLREAD**

Rows are blocked for read-only cursors. All cursors in a program that are not explicitly able to be changed are opened for read-only processing even though there may be EXECUTE or EXECUTE IMMEDIATE statements in the program.

Specifying *ALLREAD:

- Allows record blocking for all read-only cursors.
- Can improve the performance of almost all read-only cursors in programs, but limits queries in the following ways:
  - The Rollback (ROLLBACK) command, a ROLLBACK statement in host languages, or the ROLLBACK HOLD SQL statement does not reposition a read-only cursor when *ALLREAD is specified.
  - Dynamic running of a positioned UPDATE or DELETE statement (for example, using EXECUTE IMMEDIATE), can not be used to update a row in a cursor unless the DECLARE statement for the cursor includes the FOR UPDATE clause.

***NONE**

Rows are not blocked for retrieval of data for cursors.

Specifying *NONE:

- Guarantees that the data retrieved is current.
- May reduce the amount of time required to retrieve the first row of data for a query.
- Stops the database manager from retrieving a block of data rows that is not used by the program when only the first few rows of a query are retrieved before the query is closed.
- Can degrade the overall performance of a query that retrieves a large number of rows.

***READ**

Records are blocked for read-only retrieval of data for cursors when:

- *NONE is specified for the **Commitment control (COMMIT)** parameter, which indicates that commitment control is not used.
- The cursor is declared with a FOR READ ONLY clause or there are no dynamic statements that could run a positioned UPDATE or DELETE statement for the cursor.

Top

Delay PREPARE (DLYPRP)

Specifies whether the dynamic statement validation for a PREPARE statement is delayed until an OPEN, EXECUTE, or DESCRIBE statement is run. Delaying validation improves performance by eliminating redundant validation.

***NO** Dynamic statement validation is not delayed. When the dynamic statement is prepared, the access plan is validated. When the dynamic statement is used in an OPEN or EXECUTE statement, the access plan is revalidated. Because the authority or the existence of objects referred

to by the dynamic statement may change, you must still check the SQLCODE or SQLSTATE after issuing the OPEN or EXECUTE statement to ensure that the dynamic statement is still valid.

***YES** Dynamic statement validation is delayed until the dynamic statement is used in an OPEN, EXECUTE, or DESCRIBE SQL statement. When the dynamic statement is used, the validation is completed and an access plan is built. If you specify *YES for this parameter for precompiled programs, you should check the SQLCODE and SQLSTATE after running an OPEN, EXECUTE, or DESCRIBE statement to ensure that the dynamic statement is valid.

Note: If you specify *YES, performance is not improved if the INTO clause is used on the PREPARE statement or if a DESCRIBE statement uses the dynamic statement before an OPEN is issued for the statement.

Top

Severity level (GENLVL)

Specifies whether the compiler is called, depending on the severity of messages generated as a result of errors found by the SQL precompiler. If precompiler errors are generated that have a message severity level greater than the value specified for this parameter, the compiler is not called.

If the **Relational database (RDB)** parameter is specified and the severity of the messages generated as a result of package creation is greater than the severity level specified for this parameter, the SQL package is not created.

10 Do not call the compiler if SQL precompiler messages with a message severity greater than 10 are generated.

0-40 Specify the maximum SQL precompiler message severity level to be used to control whether the compiler is called.

Top

Date format (DATFMT)

Specifies the format used when accessing date result columns. All output date fields are returned in the format you specify. For input date strings, the value you specify is used to determine whether the date is a valid format.

Note: An input date string that uses the format *USA, *ISO, *EUR, or *JIS is always valid.

If you connect to a relational database that is on a system that is not a System i, the format must be *USA, *ISO, *EUR, or *JIS.

***JOB** The format specified for the job at precompile time or when a new interactive SQL session is created is used.

Use the Display Job (DSPJOB) command to determine the current date format for the job.

***USA** The United States date format **mm/dd/yyyy** is used.

***ISO** The International Organization for Standardization (ISO) date format **yyyy-mm-dd** is used.

***EUR** The European date format **dd.mm.yyyy** is used.

***JIS** The Japanese Industrial Standard date format **yyyy-mm-dd** is used.

***MDY** The date format **mm/dd/yy** is used.

***DMY** The date format **dd/mm/yy** is used.

***YMD** The date format **yy/mm/dd** is used.

***JUL** The Julian date format **yy/ddd** is used.

Top

Date separator character (DATSEP)

Specifies the separator to be used when accessing date result columns.

Note: This parameter applies only when *JOB, *MDY, *DMY, *YMD, or *JUL is specified for the **Date format (DATFMT)** parameter.

***JOB** The date separator specified for the job at precompile time, when a new interactive SQL session is created, or when Run SQL Statement (RUNSQLSTM) command is run.

Use the Display Job (DSPJOB) command to determine the current date separator value for the job.

'/' A slash is used as the date separator.

'.' A period is used as the date separator.

'-' A dash is used as the date separator.

',' A comma is used as the date separator.

' ' *or* *BLANK
A blank is used as the date separator.

Top

Time format (TIMFMT)

Specifies the format used when accessing time result columns. All output time fields are returned in the format you specify. For input time strings, the value you specify is used to determine whether the time is specified in a valid format.

Note: An input time string that uses the format *USA, *ISO, *EUR, or *JIS is always valid.

If you connect to a relational database that is on a system that is not another System i, the time format must be *USA, *ISO, *EUR, *JIS, or *HMS with a time separator of a colon or period.

***HMS** The **hh:mm:ss** format is used.

***USA** The United States time format **hh:mmxx** is used, where **xx** is AM or PM.

***ISO** The International Organization for Standardization (ISO) time format **hh.mm.ss** is used.

***EUR** The European time format **hh.mm.ss** is used.

***JIS** The Japanese Industrial Standard time format **hh:mm:ss** is used.

Top

Time separator character (TIMSEP)

Specifies the separator used when accessing time result columns.

Note: This parameter applies only when *HMS is specified for the **Time format (TIMFMT)** parameter.

***JOB** The time separator specified for the job at precompile time, when a new interactive SQL session is created, or when RUNSQLSTM is run is used.

Use the Display Job (DSPJOB) command to determine the current time separator value for the job.

':' A colon is used as the time separator.

'.' A period is used as the time separator.

',' A comma is used as the time separator.

' ' *or* *BLANK
A blank is used as the time separator.

Top

Replace (REPLACE)

Specifies whether a SQL program or SQL package is created when there is an existing SQL program or SQL package of the same name in the same library. The value is passed to the CRTxxxPGM command (where xxx is the language of the program being created) and the Create SQL Package (CRTSQLPKG) command if the **Relational database (RDB)** parameter is specified.

***YES** An SQL program or SQL package is created and any existing SQL program or SQL package of the same name in the specified library is moved to QRPLOBJ. The authorities for the existing SQL package are kept for the new SQL package.

***NO** An SQL program or package is not created if an SQL program or package of the same name already exists in the specified library.

Top

RDB connect method (RDBCNNMTH)

Specifies the semantics used for CONNECT statements.

***DUW**
CONNECT (Type 2) semantics are used to support distributed unit of work. Consecutive CONNECT statements to additional relational databases do not result in disconnection of previous connections.

***RUW** CONNECT (Type 1) semantics are used to support remote unit of work. Consecutive CONNECT statements result in the previous connection being disconnected before a new connection is established.

Top

Default collection (DFTRDBCOL)

Specifies the name of the schema identifier used for the unqualified names of the tables, views, indexes, SQL packages, aliases, constraints, external programs, node groups, and triggers. This parameter applies only to static SQL statements.

***NONE**
The naming convention specified for the **Precompiler options (OPTION)** parameter is used.

name Specify the name of the schema identifier to be used instead of the naming convention specified for the **Precompiler options (OPTION)** parameter.

Top

Dynamic default collection (DYNDFTCOL)

Specifies whether the default schema name specified for the **Default collection (DFTRDBCOL)** parameter is also used for dynamic statements.

***NO** Do not use the value specified for the **Default collection (DFTRDBCOL)** parameter for unqualified names of tables, views, indexes, SQL packages, aliases, constraints, external programs, node groups, and triggers for dynamic SQL statements. The naming convention specified for the **Precompiler options (OPTION)** parameter is used.

***YES** The schema name specified for the **Default collection (DFTRDBCOL)** parameter will be used for the unqualified names of the tables, views, indexes, SQL packages, aliases, constraints, external programs, node groups, and triggers in dynamic SQL statements.

Top

Package (SQLPKG)

Specifies the name and library of the SQL package to be created on the remote relational database specified for the **Relational database (RDB)** parameter of this command.

Qualifier 1: Package

***PGM** The name of the SQL package is the same as the program name.

name Specify the name of the SQL package. If the remote system is not a System i, a maximum of 8 characters can be specified.

Qualifier 2: Library

***PGMLIB**
The SQL package is placed in the schema that has the same name as the library containing the program.

name Specify the name of the schema where the SQL package is to be placed. If the remote system is not a System i, a maximum of 8 characters can be specified.

Top

SQL path (SQLPATH)

Specifies the path to be used to find procedures, functions, and user defined types in static SQL statements.

***NAMING**
The path used depends on the naming convention specified for the **Precompiler options (OPTION)** parameter.

For *SYS naming, the path used is *LIBL, the current library list at runtime.

For *SQL naming, the path used is ″QSYS″, ″QSYS2″, ″userid″, where ″userid″ is the value of the USER special register. If a schema name is specified for the **Default collection (DFTRDBCOL)** parameter, the schema name takes the place of userid.

***LIBL** The path used is the library list at runtime.

name Specify one or more schema names. A maximum of 268 schema names may be specified.

Top

SQL rules (SQLCURRULE)

Specifies the semantics used for SQL statements.

***DB2** The semantics of all SQL statements will default to the rules established for DB2. The following semantics are controlled by this option:

Hexadecimal constants are treated as character data.

***STD** The semantics of all SQL statements will default to the rules established by the ISO and ANSI SQL standards. The following semantics are controlled by this option:

Hexadecimal constants are treated as binary data.

Top

IBM SQL flagging (SAAFLAG)

Specifies the IBM SQL flagging function. This parameter allows you to flag SQL statements to verify whether they conform to IBM SQL syntax.

***NOFLAG**
No checks are made to see whether SQL statements conform to IBM SQL syntax.

***FLAG**
Checks are made to see whether SQL statements conform to IBM SQL syntax.

Top

ANS flagging (FLAGSTD)

Specifies whether non-standard statements are flagged. This parameter allows you to flag SQL statements to verify whether they conform to the Core level of the ISO/IEC 9075-2003 standards.

***NONE**
No checks are made to see whether SQL statements conform to ANSI standards.

***ANS** Checks are made to see whether SQL statements conform to standards.

Top

Print file (PRTFILE)

Specifies the printer device file to be used for the precompiler output listing.

Qualifier 1: Print file

QSYSPRT
The precompiler output file is directed to the IBM-supplied printer file, QSYSPRT. The file QSYSPRT has a record length of 132. If you specify a file whose record length is less than 132, information is lost.

name Specify the name of the printer device file to which the precompiler output is directed.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the printer file is located.

Top

User profile (USRPRF)

Specifies the user profile that is used when the compiled program object and SQL package object is run, including the authority that the program object or SQL package has for each object in static SQL statements. The profile of either the owner or the user is used to control access to objects.

***NAMING**
The user profile is determined by the naming convention. If the naming convention is *SQL, USRPRF(*OWNER) is used. If the naming convention is *SYS, USRPRF(*USER) is used.

***USER**
The profile of the user running the program or SQL package is used.

***OWNER**
The user profiles of both the owner and the user are used when the program or SQL package is run.

Top

Dynamic user profile (DYNUSRPRF)

Specifies the user profile used for dynamic SQL statements.

***USER**
Local dynamic SQL statements are run under the profile of the program's user. Distributed dynamic SQL statements are run under the profile of the application server job.

***OWNER**
Local dynamic SQL statements are run under the profile of the program's owner. Distributed dynamic SQL statements are run under the profile of the SQL package's owner.

Top

Sort sequence (SRTSEQ)

Specifies the sort sequence table to be used for string comparisons in SQL statements.

Note: *HEX must be specified for this parameter on distributed applications where the application server is not on a System i.

Single values

***JOB** The SRTSEQ value for the job is used.

***JOBRUN**
The SRTSEQ value for the job is retrieved when the program is run. For distributed applications, SRTSEQ(*JOBRUN) is valid only when LANGID(*JOBRUN) is also specified.

***LANGIDUNQ**
The unique-weight sort table for the language specified for the **Language id (LANGID)** parameter is used.

***LANGIDSHR**
The shared-weight sort table for the language specified for the LANGID parameter is used.

***HEX** A sort sequence table is not used. The hexadecimal values of the characters are used to determine the sort sequence.

Qualifier 1: Sort sequence

name Specify the name of the sort sequence table to be used with this program.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library to be searched.

Top

Language id (LANGID)

Specifies the language identifier to be used when SRTSEQ(*LANGIDUNQ) or SRTSEQ(*LANGIDSHR) is specified.

***JOB** The LANGID value for the job is retrieved during the precompile.

***JOBRUN**
The LANGID value for the job is retrieved when the program is run. For distributed applications, LANGID(*JOBRUN) is valid only when SRTSEQ(*JOBRUN) is also specified.

language-id
Specify the language identifier to be used by the program.

Top

To source file (TOSRCFILE)

Specifies the source file that is to contain the output source member that has been processed by the SQL precompiler. If the specified source file is not found, it will be created. The output member will have the same name as the name specified for the **Object (OBJ)** or **Program (PGM)** parameter.

Qualifier 1: To source file

QSQLTEMP
The source file QSQLTEMP will be used.

name Specify the name of the source file to contain the output source member.

Qualifier 2: Library

QTEMP
The library QTEMP will be used.

***LIBL** The job's library list is searched for the specified file. If the file is not found in any library in the library list, the file will be created in the current library.

***CURLIB**
The current library for the job will be used. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library that is to contain the output source file.

Top

Decimal result options (DECRESULT)

Specifies the maximum precision, maximum scale and minimum divide scale that should be returned for result data types. The specified limit only applies to numeric (zoned) and decimal (packed) data types used in arithmetic expressions and in SQL column functions AVG and SUM.

Element 1: Maximum precision

31 The maximum precision (length) that should be returned for the result data types is 31 digits.

63 The maximum precision (length) that should be returned for the result data types is 63 digits.

Element 2: Maximum scale

31 The maximum scale (number of decimal positions to the right of the decimal point) that should be returned for the result data types is 31 digits.

0-63 Specify the maximum scale (number of decimal positions to the right of the decimal point) that should be returned for the result data types. The value can range from 0 to the maximum precision.

Element 3: Minimum divide scale

0 The minimum divide scale (number of decimal positions to the right of the decimal point) that should be returned for both intermediate and result data types is 0.

0-9 Specify the minimum divide scale (number of decimal positions to the right of the decimal point) that should be returned for both intermediate and result data types. The value cannot to exceed the maximum scale. If 0 is specified for the maximum scale, minimum divide scale is not used.

Top

Decimal float rounding mode (DECFLTRND)

Specifies the decimal floating point rounding mode used for static SQL statements.

***HALFEVEN**
Round to nearest; if equidistant, round so that the final digit is even. If the discarded digits represent greater than half (0.5) the value of a one in the next left position then the result coefficient should be incremented by 1 (rounded up). If they represent less than half, then the result coefficient is not adjusted (that is, the discarded digits are ignored). Otherwise (they represent exactly half) the result coefficient is unaltered if its rightmost digit is even, or incremented by 1 (rounded up) if its rightmost digit is odd (to make an even digit).

***HALFUP**
Round to nearest; if equidistant, round up. If the discarded digits represent greater than or equal to half (0.5) of the value of a one in the next left position then the result coefficient should be incremented by 1 (rounded up). Otherwise the discarded digits are ignored.

***DOWN**
Round towards 0 (truncation). The discarded digits are ignored.

***CEILING**
Round towards +infinity. If all of the discarded digits are zero or if the sign is negative the result is unchanged other than the removal of discarded digits. Otherwise, the result coefficient should be incremented by 1 (rounded up).

***FLOOR**
Round towards -infinity. If all of the discarded digits are zero or if the sign is positive the result is unchanged other than the removal of discarded digits. Otherwise, the sign is negative and the result coefficient should be incremented by 1.

***HALFDOWN**
Round to nearest; if equidistant, round down. If the discarded digits represent greater than half (0.5) of the value of a one in the next left position then the result coefficient should be incremented by 1 (rounded up). Otherwise (the discarded digits are 0.5 or less) the discarded digits are ignored.

***UP** Round away from 0. Of all of the discarded digits are zero the result is unchanged other than the removal of discarded digits. Otherwise, the result coefficient should be incremented by 1 (rounded up).

Top

Compiler options (COMPILEOPT)

Specifies additional parameters to be used on the compiler command. The COMPILEOPT string is added to the compiler command built by the precompiler. There is no validation of the string. The compiler command will issue an error if any parameter is incorrect. Please refer to the DB2 for i5/OS SQL programming topic collection in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/ for a list of parameters that the precompiler generates for the compiler command. Using any of the keywords that the precompiler passes to the compiler will cause the compiler command to fail because of duplicate parameters.

***NONE**
No additional parameters will be used on the compiler command.

character-value
Specify no more than 5000 characters, enclosed in apostrophes.

Top

Examples

```
CRTSQLCBL   PGM(ACCTS/STATS)  SRCFILE(ACCTS/ACTIVE)
            TEXT('Statistical analysis of active accounts')
```

This command runs the SQL precompiler which precompiles the COBOL source and stores the changed source in member STATS of source file QSQLTEMP in library QTEMP. The COBOL compiler is called to create program STATS in library ACCTS using the source member created by the SQL precompiler.

Top

Error messages

***ESCAPE Messages**

SQL9001
SQL precompile failed.

SQL9002
Conflict in TGTRLS parameters for SQL precompile and &7 compile.

SQL9003
&7 Compile at wrong level for SQL source.

SQL9004
Create of SQL package failed.

SQL9006
DB2 Query Mgr and SQL DevKit not at same install level as the operating system.

Top

Create SQL ILE COBOL Object (CRTSQLCBLI)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Create SQL ILE COBOL Object (CRTSQLCBLI) command calls the Structured Query Language (SQL) precompiler which precompiles COBOL source containing SQL statements, produces a temporary source member, and then optionally calls the ILE COBOL for System i compiler to create a module, create a program, or create a service program.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Object	*Qualified object name*	Required, Positional 1
	Qualifier 1: Object	*Name*	
	Qualifier 2: Library	*Name*, ***CURLIB**	
SRCFILE	Source file	*Qualified object name*	Optional, Positional 2
	Qualifier 1: Source file	*Name*, **QCBLLESRC**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SRCMBR	Source member	*Name*, ***OBJ**	Optional, Positional 3
SRCSTMF	Source stream file	*Path name*	Optional
COMMIT	Commitment control	***CHG**, *ALL, *CS, *NONE, *RR, *UR, *RS, *NC	Optional
RDB	Relational database	*Simple name*, ***LOCAL**, *NONE	Optional
OBJTYPE	Compile type	***PGM**, *SRVPGM, *MODULE	Optional
OUTPUT	Listing output	***NONE**, *PRINT	Optional
TEXT	Text 'description'	*Character value*, ***SRCMBRTXT**, *BLANK	Optional
USER	RDB user	*Name*, ***CURRENT**	Optional
PASSWORD	RDB user password	*Character value*, ***NONE**, ' '	Optional
OPTION	Precompiler options	Values (up to 18 repetitions): ***XREF**, *NOXREF, *GEN, *NOGEN, *COMMA, *PERIOD, *JOB, *SYSVAL, *QUOTESQL, *APOSTSQL, *QUOTE, *APOST, *SECLVL, *NOSECLVL, *EVENTF, *NOEVENTF, *CVTDT, *NOCVTDT, *SQL, *SYS, *OPTLOB, *NOOPTLOB, *NOEXTIND, *EXTIND	Optional
TGTRLS	Target release	*Simple name*, ***CURRENT**, *PRV	Optional
INCFILE	INCLUDE file	*Qualified object name*	Optional
	Qualifier 1: INCLUDE file	*Name*, ***SRCFILE**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
INCDIR	SQL INCLUDE directory	*Path name*, ***NONE**	Optional
ALWCPYDTA	Allow copy of data	***OPTIMIZE**, *YES, *NO	Optional
CLOSQLCSR	Close SQL cursor	***ENDACTGRP**, *ENDMOD	Optional
ALWBLK	Allow blocking	***ALLREAD**, *NONE, *READ	Optional
DLYPRP	Delay PREPARE	***NO**, *YES	Optional
GENLVL	Severity level	0-40, **10**	Optional

Keyword	Description	Choices	Notes
DATFMT	Date format	***JOB**, *USA, *ISO, *EUR, *JIS, *MDY, *DMY, *YMD, *JUL	Optional
DATSEP	Date separator character	***JOB**, '/', '.', ',', '-', ' ', *BLANK	Optional
TIMFMT	Time format	***HMS**, *USA, *ISO, *EUR, *JIS	Optional
TIMSEP	Time separator character	***JOB**, ':', '.', ',', ' ', *BLANK	Optional
REPLACE	Replace	***YES**, *NO	Optional
RDBCNNMTH	RDB connect method	***DUW**, *RUW	Optional
DFTRDBCOL	Default collection	*Name*, ***NONE**	Optional
DYNDFTCOL	Dynamic default collection	***NO**, *YES	Optional
SQLPKG	Package	*Qualified object name*	Optional
	Qualifier 1: Package	*Name*, ***OBJ**	
	Qualifier 2: Library	*Name*, ***OBJLIB**	
SQLPATH	SQL path	Single values: ***NAMING**, *LIBL Other values (up to 268 repetitions): *Name*	Optional
SQLCURRULE	SQL rules	***DB2**, *STD	Optional
SAAFLAG	IBM SQL flagging	***NOFLAG**, *FLAG	Optional
FLAGSTD	ANS flagging	***NONE**, *ANS	Optional
PRTFILE	Print file	*Qualified object name*	Optional
	Qualifier 1: Print file	*Name*, **QSYSPRT**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
DBGVIEW	Debugging view	***NONE**, *SOURCE	Optional
USRPRF	User profile	***NAMING**, *USER, *OWNER	Optional
DYNUSRPRF	Dynamic user profile	***USER**, *OWNER	Optional
SRTSEQ	Sort sequence	Single values: ***JOB**, *HEX, *JOBRUN, *LANGIDUNQ, *LANGIDSHR Other values: *Qualified object name*	Optional
	Qualifier 1: Sort sequence	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
LANGID	Language id	*Character value*, ***JOB**, *JOBRUN	Optional
TOSRCFILE	To source file	*Qualified object name*	Optional
	Qualifier 1: To source file	*Name*, **QSQLTEMP**	
	Qualifier 2: Library	*Name*, **QTEMP**, *LIBL, *CURLIB	
DECRESULT	Decimal result options	*Element list*	Optional
	Element 1: Maximum precision	**31**, 63	
	Element 2: Maximum scale	0-63, **31**	
	Element 3: Minimum divide scale	0-9, **0**	
DECFLTRND	Decimal float rounding mode	***HALFEVEN**, *HALFUP, *DOWN, *CEILING, *FLOOR, *HALFDOWN, *UP	Optional
COMPILEOPT	Compiler options	*Character value*, ***NONE**	Optional

Top

Object (OBJ)

Specifies the object to be created.

This is a required parameter.

Qualifier 1: Object

name Specify the name of the object to be created.

Qualifier 2: Library

***CURLIB**
The new object is created in the current library for the job. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the object is created.

Top

Source file (SRCFILE)

Specifies the source file that contains the COBOL source statements and SQL statements.

Qualifier 1: Source file

QCBLLESRC
Source file QCBLLESRC contains the COBOL source.

name Specify the name of the source file that contains the COBOL source. This source file should have a record length of 92. The source file can be a database file, device file, or an inline data file.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library to be searched.

Top

Source member (SRCMBR)

Specifies the source file member that contains the input source. This parameter is used only if the source file specified for the **Source file (SRCFILE)** parameter is a database file.

***OBJ** The source file member that has the same name as the value specified for the **Object (OBJ)** parameter contains the input source.

name Specify the name of the source file member that contains the input source.

Top

Source stream file (SRCSTMF)

Specifies the path name to the file that contains the COBOL source statements and SQL statements. The path name can be either absolute or relative.

Source text must be in positions 1 to 80. Characters after position 80 will be ignored.

Top

Commitment control (COMMIT)

Specifies whether SQL statements in the compiled program are run under commitment control. Files referred to in the host language source are not affected by this option. Only SQL tables, SQL views, SQL packages, SQL sequences, SQL aliases, SQL Types, SQL procedures, SQL functions, SQL indexes, SQL schemas, SQL triggers, and SQL views referred to in SQL statements are affected.

***CHG** *or* ***UR**
Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs can be seen.

***CS** Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows updated, deleted, and inserted are locked until the end of the unit of work (transaction). A row that is selected, but not updated, is locked until the next row is selected. Uncommitted changes in other jobs cannot be seen.

***ALL** *or* ***RS**
Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows selected, updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs cannot be seen.

***NONE** *or* ***NC**
Specifies that commitment control is not used. Uncommitted changes in other jobs can be seen. If the SQL DROP SCHEMA statement is included in the program, *NONE or *NC must be used. If a relational database is specified for the **Relational database (RDB)** parameter, and the relational database is on a system that is not on a System i, *NONE or *NC cannot be specified.

***RR** Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows selected, updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs cannot be seen. All tables referred to in SELECT, UPDATE, DELETE, and INSERT statements are locked exclusively until the end of the unit of work (transaction).

Top

Relational database (RDB)

Specifies the name of the relational database where the SQL package is to be created.

***LOCAL**
The program is created as a distributed SQL program. The SQL statements will access the local database. An SQL package object is not created as part of the precompile process. The Create Structured Query Language Package (CRTSQLPKG) command can be used.

***NONE**
An SQL package object is not created. The program object is not a distributed program and the Create Structured Query Language Package (CRTSQLPKG) command cannot be used.

name Specify the name of the relational database where the new SQL package object is to be created. When the name of the local relational database is specified, the program created is still a distributed SQL program. The SQL statements will access the local database.

Top

Compile type (OBJTYPE)

Specifies the type of object to be created.

When OBJTYPE(*PGM) or OBJTYPE(*SRVPGM) is specified and the **Relational database (RDB)** parameter is also specified, the CRTSQLPKG command is issued by the SQL precompiler after the program has been created. When OBJTYPE(*MODULE) is specified, an SQL package is not created and you must issue the CRTSQLPKG command after the CRTPGM or CRTSRVPGM command has created the program.

If OPTION(*NOGEN) is specified, only the SQL temporary source member is generated. No module, program, service program, or SQL package is created.

***PGM** The SQL precompiler calls the compiler to create a program.

***MODULE**
The SQL precompiler calls the compiler to create a module.

***SRVPGM**
The SQL precompiler calls the compiler to create a module and issues the Create Service Program (CRTSRVPGM) command to create a service program.

Top

Listing output (OUTPUT)

Specifies whether the precompiler listing is generated.

***NONE**
The precompiler listing is not generated.

***PRINT**
The precompiler listing is generated.

Top

Text 'description' (TEXT)

Specifies text that briefly describes the program and its function.

***SRCMBRTXT**
The text is taken from the source file member being used to create the program. If the source file is an inline file or a device file, the text is blank.

***BLANK**
No text is specified.

'description'
Specify no more than 50 characters, enclosed in apostrophes.

Top

RDB user (USER)

Specifies the user name sent to the remote system when starting the conversation. This parameter is valid only when **Relational database (RDB)** is specified.

***CURRENT**
The user name associated with the current job is used.

name Specify the user name to be used for the application server job.

Top

RDB user password (PASSWORD)

Specifies the password to be used on the remote system. This parameter is valid only when **Relational database (RDB)** is specified.

***NONE**
No password is sent. A user name cannot be specified for the **RDB user (USER)** parameter if this value is specified.

Note: Specifying a password of a blank is the same as specifying *NONE.

password
Specify the password of the user name specified for the **RDB user (USER)** parameter.

Top

Precompiler options (OPTION)

Specifies whether one or more of the following options are used when the COBOL source is precompiled. If an option is specified more than once, or if two options conflict, the last option specified is used.

Cross reference options:

***XREF** The precompiler cross-references items in the program to the statement numbers in the program that refer to those items.

***NOXREF**
The precompiler does not cross-reference names.

Program creation options:

***GEN** The precompiler creates the object that is specified for the **Compile type (OBJTYPE)** parameter.

***NOGEN**
The precompiler does not call the ILE COBOL compiler. No module, program, service program, or SQL package is created.

Decimal point options:

***JOB** The representation for the decimal point specified for the job at precompile time is used.

Note: If QDECFMT specifies that the value used as the decimal point is a comma, any numeric constants in lists (such as in the SELECT clause and VALUES clause) must be separated by a comma followed by a blank. For example, VALUES(1,1, 2,23, 4,1) is equivalent to VALUES(1.1,2.23,4.1) where the decimal point is the period.

***SYSVAL**
The value used as the decimal point in numeric constants is from the QDECFMT system value. This value is also used as the decimal point character when casting a numeric value to character.

Note: If QDECFMT specifies that the value used as the decimal point is a comma; any numeric constants in lists (such as in the SELECT clause and VALUES clause) must be separated by a comma followed by a blank. For example, VALUES(1,1, 2,23, 4,1) is equivalent to VALUES(1.1,2.23,4.1) where the decimal point is the period.

***PERIOD**
The value used as the decimal point for numeric constants used in SQL statements is a period. This value is also used as the decimal point character when casting a numeric value to character.

***COMMA**
The value used as the decimal point in numeric constants is a comma. Any numeric constants in lists (such as in the SELECT clause and VALUES clause) must be separated by a comma followed by a blank. For example, VALUES(1,1, 2,23, 4,1) is equivalent to VALUES(1.1,2.23,4.1) where the decimal point is the period.

Naming convention options:

***SYS** Specifies that the system naming convention is used (library-name/file-name).

***SQL** Specifies that the SQL naming convention is used (schema-name.table-name).

If a relational database is specified for the **Relational database (RDB)** parameter, and the database is on a system that is not a System i, *SQL must be specified as the naming convention.

Second-level message text options:

***NOSECLVL**
Second-level text descriptions are not added to the listing.

***SECLVL**
Second-level text with replacement data is added for all messages on the listing.

SQL string delimiter options:

***QUOTESQL**
The character used as the string delimiter in the SQL statements is the quote (″).

***APOSTSQL**
The character used as the string delimiter in the SQL statements is the apostrophe (′).

COBOL string delimiter options:

***QUOTE**
The character used for nonnumeric literals and Boolean literals in the COBOL statements is the quote (″).

***APOST**
The character used for nonnumeric literals and Boolean literals in the COBOL statements is the apostrophe (′).

LOB optimization for DRDA options:

***OPTLOB**
The first FETCH for a cursor determines how the cursor will be used for LOBs (Large Objects) on all subsequent FETCHes. This option remains in effect until the cursor is closed.

If the first FETCH uses a LOB locator to access a LOB column, no subsequent FETCH for that cursor can fetch that LOB column into a LOB host variable.

If the first FETCH places the LOB column into a LOB host variable, no subsequent FETCH for that cursor can use a LOB locator for that column.

***NOOPTLOB**
There is no restriction on whether a column is retrieved into a LOB locator or into a LOB host variable. This option can cause performance to degrade.

Event file creation options:

***NOEVENTF**
The compiler will not produce an event file for use by CoOperative Development Environment (CODE).

***EVENTF**
The compiler produces an event file for use by CoOperative Development Environment (CODE). The event file will be created as a member in the file EVFEVENT in your object library. CODE uses this file to offer error feedback integrated with the CODE editor. This option is normally specified by CODE on your behalf.

Character string representations options:

***NOCVTDT**
Specifies that date, time, and timestamp data types which are retrieved from externally-described database files are to be processed using the date, time, and timestamp data types.

***CVTDT**
Specifies that date, time and timestamp data types, which are retrieved from externally-described database files, are to be processed as fixed-length character fields.

Extended indicators options:

***NOEXTIND**
Extended indicator support is not enabled.

***EXTIND**
Extended indicator support is enabled.

Top

Target release (TGTRLS)

Specifies the release of the operating system on which you intend to use the object being created.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

***CURRENT**
The object is to be used on the release of the operating system currently running on your system. The object can also be used on a system with any subsequent release of the operating system installed.

***PRV** The object is to be used on the previous release with modification level 0 of the operating system. The object can also be used on a system with any subsequent release of the operating system installed.

target-release
Specify the release in the format VxRxMx. The object can be used on a system with the specified release or with any subsequent release of the operating system installed.

Top

INCLUDE file (INCFILE)

Specifies the source file that contains members to be included in the program with the SQL INCLUDE statement.

Single values

***SRCFILE**
The qualified source file you specify for the **Source file (SRCFILE)** parameter contains the source file members specified on any SQL INCLUDE statements.

Qualifier 1: INCLUDE file

name Specify the name of the source file that contains the source file members specified on any SQL INCLUDE statements.

The record length of the source file you specify here must be no less than the record length of the source file you specify for the **Source file (SRCFILE)** parameter.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the source file is located.

Top

SQL INCLUDE directory (INCDIR)

Specifies the path name of the directory containing any files to be included in the program with the SQL INCLUDE statement.

Specifying a value for this parameter does not pass an INCDIR value from the SQL precompiler to the compiler. If you want to pass an INCDIR value through to the compiler, this can be done using the **Compiler options (COMPILEOPT)** parameter.

***NONE**
The current directory and the source directory will be searched.

path-name
The path name of the directory that contains the files specified on any SQL INCLUDE statement. The current directory will be searched first, then the specified path name, then the source directory.

Top

Allow copy of data (ALWCPYDTA)

Specifies whether a copy of the data can be used in a SELECT statement.

***OPTIMIZE**
The system determines whether to use the data retrieved directly from the database or to use a copy of the data. The decision is based on which method provides the best performance. If the **Commitment control (COMMIT)** parameter is not *NONE, the **Allow blocking (ALWBLK)** parameter should be set to *ALLREAD, when possible, for best performance.

***YES** A copy of the data is used only when necessary.

***NO** A copy of the data is not used. If a temporary copy of the data is required to perform the query, an error message is returned.

Top

Close SQL cursor (CLOSQLCSR)

Specifies when SQL cursors are implicitly closed, SQL prepared statements are implicitly discarded, and LOCK TABLE locks are released. SQL cursors are explicitly closed when the user issues the CLOSE, COMMIT, or ROLLBACK (without HOLD) SQL statements.

***ENDACTGRP**
SQL cursors are closed and SQL prepared statements are implicitly discarded, and LOCK TABLE locks are released when the activation group ends.

***ENDMOD**
SQL cursors are closed and SQL prepared statements are implicitly discarded when the module is exited. LOCK TABLE locks are released when the first SQL program on the call stack ends.

Top

Allow blocking (ALWBLK)

Specifies whether the database manager can use record blocking and the extent to which blocking can be used for read-only cursors.

***ALLREAD**
Rows are blocked for read-only cursors. All cursors in a program that are not explicitly able to be changed are opened for read-only processing even though there may be EXECUTE or EXECUTE IMMEDIATE statements in the program.

Specifying *ALLREAD:

- Allows record blocking for all read-only cursors.
- Can improve the performance of almost all read-only cursors in programs, but limits queries in the following ways:
  - The Rollback (ROLLBACK) command, a ROLLBACK statement in host languages, or the ROLLBACK HOLD SQL statement does not reposition a read-only cursor when *ALLREAD is specified.
  - Dynamic running of a positioned UPDATE or DELETE statement (for example, using EXECUTE IMMEDIATE), can not be used to update a row in a cursor unless the DECLARE statement for the cursor includes the FOR UPDATE clause.

***NONE**
Rows are not blocked for retrieval of data for cursors.

Specifying *NONE:

- Guarantees that the data retrieved is current.
- May reduce the amount of time required to retrieve the first row of data for a query.
- Stops the database manager from retrieving a block of data rows that is not used by the program when only the first few rows of a query are retrieved before the query is closed.
- Can degrade the overall performance of a query that retrieves a large number of rows.

***READ**

Records are blocked for read-only retrieval of data for cursors when:

- *NONE is specified for the **Commitment control (COMMIT)** parameter, which indicates that commitment control is not used.
- The cursor is declared with a FOR READ ONLY clause or there are no dynamic statements that could run a positioned UPDATE or DELETE statement for the cursor.

Top

Delay PREPARE (DLYPRP)

Specifies whether the dynamic statement validation for a PREPARE statement is delayed until an OPEN, EXECUTE, or DESCRIBE statement is run. Delaying validation improves performance by eliminating redundant validation.

***NO** Dynamic statement validation is not delayed. When the dynamic statement is prepared, the access plan is validated. When the dynamic statement is used in an OPEN or EXECUTE statement, the access plan is revalidated. Because the authority or the existence of objects referred to by the dynamic statement may change, you must still check the SQLCODE or SQLSTATE after issuing the OPEN or EXECUTE statement to ensure that the dynamic statement is still valid.

***YES** Dynamic statement validation is delayed until the dynamic statement is used in an OPEN, EXECUTE, or DESCRIBE SQL statement. When the dynamic statement is used, the validation is completed and an access plan is built. If you specify *YES for this parameter for precompiled programs, you should check the SQLCODE and SQLSTATE after running an OPEN, EXECUTE, or DESCRIBE statement to ensure that the dynamic statement is valid.

Note: If you specify *YES, performance is not improved if the INTO clause is used on the PREPARE statement or if a DESCRIBE statement uses the dynamic statement before an OPEN is issued for the statement.

Top

Severity level (GENLVL)

Specifies whether the compiler is called, depending on the severity of messages generated as a result of errors found by the SQL precompiler. If precompiler errors are generated that have a message severity level greater than the value specified for this parameter, the compiler is not called.

If the **Relational database (RDB)** parameter is specified and the severity of the messages generated as a result of package creation is greater than the severity level specified for this parameter, the SQL package is not created.

10 Do not call the compiler if SQL precompiler messages with a message severity greater than 10 are generated.

0-40 Specify the maximum SQL precompiler message severity level to be used to control whether the compiler is called.

Top

Date format (DATFMT)

Specifies the format used when accessing date result columns. All output date fields are returned in the format you specify. For input date strings, the value you specify is used to determine whether the date is a valid format.

Note: An input date string that uses the format *USA, *ISO, *EUR, or *JIS is always valid.

If you connect to a relational database that is on a system that is not a System i, the format must be *USA, *ISO, *EUR, or *JIS.

***JOB** The format specified for the job at precompile time or when a new interactive SQL session is created is used.

Use the Display Job (DSPJOB) command to determine the current date format for the job.

***USA** The United States date format **mm/dd/yyyy** is used.

***ISO** The International Organization for Standardization (ISO) date format **yyyy-mm-dd** is used.

***EUR** The European date format **dd.mm.yyyy** is used.

***JIS** The Japanese Industrial Standard date format **yyyy-mm-dd** is used.

***MDY** The date format **mm/dd/yy** is used.

***DMY** The date format **dd/mm/yy** is used.

***YMD** The date format **yy/mm/dd** is used.

***JUL** The Julian date format **yy/ddd** is used.

Top

Date separator character (DATSEP)

Specifies the separator to be used when accessing date result columns.

Note: This parameter applies only when *JOB, *MDY, *DMY, *YMD, or *JUL is specified for the **Date format (DATFMT)** parameter.

***JOB** The date separator specified for the job at precompile time, when a new interactive SQL session is created, or when Run SQL Statement (RUNSQLSTM) command is run.

Use the Display Job (DSPJOB) command to determine the current date separator value for the job.

'/' A slash is used as the date separator.

'.' A period is used as the date separator.

'-' A dash is used as the date separator.

',' A comma is used as the date separator.

' ' *or* *BLANK
A blank is used as the date separator.

Top

Time format (TIMFMT)

Specifies the format used when accessing time result columns. All output time fields are returned in the format you specify. For input time strings, the value you specify is used to determine whether the time is specified in a valid format.

Note: An input time string that uses the format *USA, *ISO, *EUR, or *JIS is always valid.

If you connect to a relational database that is on a system that is not another System i, the time format must be *USA, *ISO, *EUR, *JIS, or *HMS with a time separator of a colon or period.

***HMS** The **hh:mm:ss** format is used.

***USA** The United States time format **hh:mmxx** is used, where **xx** is AM or PM.

***ISO** The International Organization for Standardization (ISO) time format **hh.mm.ss** is used.

***EUR** The European time format **hh.mm.ss** is used.

***JIS** The Japanese Industrial Standard time format **hh:mm:ss** is used.

Top

Time separator character (TIMSEP)

Specifies the separator used when accessing time result columns.

Note: This parameter applies only when *HMS is specified for the **Time format (TIMFMT)** parameter.

***JOB** The time separator specified for the job at precompile time, when a new interactive SQL session is created, or when RUNSQLSTM is run is used.

Use the Display Job (DSPJOB) command to determine the current time separator value for the job.

':' A colon is used as the time separator.

'.' A period is used as the time separator.

',' A comma is used as the time separator.

' ' *or* *BLANK
A blank is used as the time separator.

Top

Replace (REPLACE)

Specifies if a SQL module, program, service program or package is created when there is an existing SQL module, program, service program, or package of the same name and type in the same library. The value for this parameter is passed to the CRTCBLMOD, CRTBNDCBL, CRTSRVPGM, and CRTSQLPKG commands.

***YES** A new SQL module, program, service program, or package is created, any existing SQL object of the same name and type in the specified library is moved to QRPLOBJ.

***NO** A new SQL module, program, service program, or package is not created if an SQL object of the same name and type already exists in the specified library.

Top

RDB connect method (RDBCNNMTH)

Specifies the semantics used for CONNECT statements.

***DUW**
CONNECT (Type 2) semantics are used to support distributed unit of work. Consecutive CONNECT statements to additional relational databases do not result in disconnection of previous connections.

***RUW** CONNECT (Type 1) semantics are used to support remote unit of work. Consecutive CONNECT statements result in the previous connection being disconnected before a new connection is established.

Top

Default collection (DFTRDBCOL)

Specifies the name of the schema identifier used for the unqualified names of the tables, views, indexes, SQL packages, aliases, constraints, external programs, node groups, and triggers. This parameter applies only to static SQL statements.

***NONE**
The naming convention specified for the **Precompiler options (OPTION)** parameter is used.

name Specify the name of the schema identifier to be used instead of the naming convention specified for the **Precompiler options (OPTION)** parameter.

Top

Dynamic default collection (DYNDFTCOL)

Specifies whether the default schema name specified for the **Default collection (DFTRDBCOL)** parameter is also used for dynamic statements.

***NO** Do not use the value specified for the **Default collection (DFTRDBCOL)** parameter for unqualified names of tables, views, indexes, SQL packages, aliases, constraints, external programs, node groups, and triggers for dynamic SQL statements. The naming convention specified for the **Precompiler options (OPTION)** parameter is used.

***YES** The schema name specified for the **Default collection (DFTRDBCOL)** parameter will be used for the unqualified names of the tables, views, indexes, SQL packages, aliases, constraints, external programs, node groups, and triggers in dynamic SQL statements.

Top

Package (SQLPKG)

Specifies the qualified name of the SQL package created on the relational database specified for the **Relational database (RDB)** parameter of this command.

Qualifier 1: Package

***OBJ** The name of the SQL package is the same as the object name specified for the **Object (OBJ)** parameter.

name Specify the name of the SQL package. If the remote system is not a System i, no more than 8 characters can be specified.

Qualifier 2: Library

***OBJLIB**
The package is created in the library with the same name as the library specified for the **Object (OBJ)** parameter.

name Specify the name of the library where the package is created.

Top

SQL path (SQLPATH)

Specifies the path to be used to find procedures, functions, and user defined types in static SQL statements.

***NAMING**
The path used depends on the naming convention specified for the **Precompiler options (OPTION)** parameter.

For *SYS naming, the path used is *LIBL, the current library list at runtime.

For *SQL naming, the path used is ″QSYS″, ″QSYS2″, ″userid″, where ″userid″ is the value of the USER special register. If a schema name is specified for the **Default collection (DFTRDBCOL)** parameter, the schema name takes the place of userid.

***LIBL** The path used is the library list at runtime.

name Specify one or more schema names. A maximum of 268 schema names may be specified.

Top

SQL rules (SQLCURRULE)

Specifies the semantics used for SQL statements.

***DB2** The semantics of all SQL statements will default to the rules established for DB2. The following semantics are controlled by this option:

Hexadecimal constants are treated as character data.

***STD** The semantics of all SQL statements will default to the rules established by the ISO and ANSI SQL standards. The following semantics are controlled by this option:

Hexadecimal constants are treated as binary data.

Top

IBM SQL flagging (SAAFLAG)

Specifies the IBM SQL flagging function. This parameter allows you to flag SQL statements to verify whether they conform to IBM SQL syntax.

***NOFLAG**
No checks are made to see whether SQL statements conform to IBM SQL syntax.

***FLAG**
Checks are made to see whether SQL statements conform to IBM SQL syntax.

Top

ANS flagging (FLAGSTD)

Specifies whether non-standard statements are flagged. This parameter allows you to flag SQL statements to verify whether they conform to the Core level of the ISO/IEC 9075-2003 standards.

***NONE**
No checks are made to see whether SQL statements conform to ANSI standards.

***ANS** Checks are made to see whether SQL statements conform to standards.

Top

Print file (PRTFILE)

Specifies the printer device file to be used for the precompiler output listing.

Qualifier 1: Print file

QSYSPRT
The precompiler output file is directed to the IBM-supplied printer file, QSYSPRT. The file QSYSPRT has a record length of 132. If you specify a file whose record length is less than 132, information is lost.

name Specify the name of the printer device file to which the precompiler output is directed.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the printer file is located.

Top

Debugging view (DBGVIEW)

Specifies the type of source debug information to be provided by the SQL precompiler.

***NONE**
The source view will not be generated.

***SOURCE**
The SQL precompiler will provide the source views for the root and if necessary, SQL INCLUDE statements. A view is to be provided which contains the statements generated by the precompiler.

Top

User profile (USRPRF)

Specifies the user profile that is used when the compiled program object and SQL package object is run, including the authority that the program object or SQL package has for each object in static SQL statements. The profile of either the owner or the user is used to control access to objects.

***NAMING**
The user profile is determined by the naming convention. If the naming convention is *SQL, USRPRF(*OWNER) is used. If the naming convention is *SYS, USRPRF(*USER) is used.

***USER**
The profile of the user running the program or SQL package is used.

***OWNER**
The user profiles of both the owner and the user are used when the program or SQL package is run.

Top

Dynamic user profile (DYNUSRPRF)

Specifies the user profile used for dynamic SQL statements.

***USER**
Local dynamic SQL statements are run under the profile of the program's user. Distributed dynamic SQL statements are run under the profile of the application server job.

***OWNER**
Local dynamic SQL statements are run under the profile of the program's owner. Distributed dynamic SQL statements are run under the profile of the SQL package's owner.

Top

Sort sequence (SRTSEQ)

Specifies the sort sequence table to be used for string comparisons in SQL statements.

Note: *HEX must be specified for this parameter on distributed applications where the application server is not on a System i.

Single values

***JOB** The SRTSEQ value for the job is used.

***JOBRUN**
The SRTSEQ value for the job is retrieved when the program is run. For distributed applications, SRTSEQ(*JOBRUN) is valid only when LANGID(*JOBRUN) is also specified.

***LANGIDUNQ**
The unique-weight sort table for the language specified for the **Language id (LANGID)** parameter is used.

***LANGIDSHR**
The shared-weight sort table for the language specified for the LANGID parameter is used.

***HEX** A sort sequence table is not used. The hexadecimal values of the characters are used to determine the sort sequence.

Qualifier 1: Sort sequence

name Specify the name of the sort sequence table to be used with this program.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library to be searched.

Top

Language id (LANGID)

Specifies the language identifier to be used when SRTSEQ(*LANGIDUNQ) or SRTSEQ(*LANGIDSHR) is specified.

***JOB** The LANGID value for the job is retrieved during the precompile.

***JOBRUN**
The LANGID value for the job is retrieved when the program is run. For distributed applications, LANGID(*JOBRUN) is valid only when SRTSEQ(*JOBRUN) is also specified.

language-id
Specify the language identifier to be used by the program.

Top

To source file (TOSRCFILE)

Specifies the source file that is to contain the output source member that has been processed by the SQL precompiler. If the specified source file is not found, it will be created. The output member will have the same name as the name specified for the **Object (OBJ)** or **Program (PGM)** parameter.

Qualifier 1: To source file

QSQLTEMP
The source file QSQLTEMP will be used.

name Specify the name of the source file to contain the output source member.

Qualifier 2: Library

QTEMP
The library QTEMP will be used.

***LIBL** The job's library list is searched for the specified file. If the file is not found in any library in the library list, the file will be created in the current library.

***CURLIB**
The current library for the job will be used. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library that is to contain the output source file.

Top

Decimal result options (DECRESULT)

Specifies the maximum precision, maximum scale and minimum divide scale that should be returned for result data types. The specified limit only applies to numeric (zoned) and decimal (packed) data types used in arithmetic expressions and in SQL column functions AVG and SUM.

Element 1: Maximum precision

31 The maximum precision (length) that should be returned for the result data types is 31 digits.

63 The maximum precision (length) that should be returned for the result data types is 63 digits.

Element 2: Maximum scale

31 The maximum scale (number of decimal positions to the right of the decimal point) that should be returned for the result data types is 31 digits.

0-63 Specify the maximum scale (number of decimal positions to the right of the decimal point) that should be returned for the result data types. The value can range from 0 to the maximum precision.

Element 3: Minimum divide scale

0 The minimum divide scale (number of decimal positions to the right of the decimal point) that should be returned for both intermediate and result data types is 0.

0-9 Specify the minimum divide scale (number of decimal positions to the right of the decimal point) that should be returned for both intermediate and result data types. The value cannot to exceed the maximum scale. If 0 is specified for the maximum scale, minimum divide scale is not used.

Top

Decimal float rounding mode (DECFLTRND)

Specifies the decimal floating point rounding mode used for static SQL statements.

***HALFEVEN**
Round to nearest; if equidistant, round so that the final digit is even. If the discarded digits represent greater than half (0.5) the value of a one in the next left position then the result coefficient should be incremented by 1 (rounded up). If they represent less than half, then the result coefficient is not adjusted (that is, the discarded digits are ignored). Otherwise (they represent exactly half) the result coefficient is unaltered if its rightmost digit is even, or incremented by 1 (rounded up) if its rightmost digit is odd (to make an even digit).

***HALFUP**
Round to nearest; if equidistant, round up. If the discarded digits represent greater than or equal to half (0.5) of the value of a one in the next left position then the result coefficient should be incremented by 1 (rounded up). Otherwise the discarded digits are ignored.

***DOWN**
Round towards 0 (truncation). The discarded digits are ignored.

***CEILING**
Round towards +infinity. If all of the discarded digits are zero or if the sign is negative the result is unchanged other than the removal of discarded digits. Otherwise, the result coefficient should be incremented by 1 (rounded up).

***FLOOR**
Round towards -infinity. If all of the discarded digits are zero or if the sign is positive the result is unchanged other than the removal of discarded digits. Otherwise, the sign is negative and the result coefficient should be incremented by 1.

***HALFDOWN**
Round to nearest; if equidistant, round down. If the discarded digits represent greater than half (0.5) of the value of a one in the next left position then the result coefficient should be incremented by 1 (rounded up). Otherwise (the discarded digits are 0.5 or less) the discarded digits are ignored.

***UP** Round away from 0. Of all of the discarded digits are zero the result is unchanged other than the removal of discarded digits. Otherwise, the result coefficient should be incremented by 1 (rounded up).

Top

Compiler options (COMPILEOPT)

Specifies additional parameters to be used on the compiler command. The COMPILEOPT string is added to the compiler command built by the precompiler. If **INCDIR(** is anywhere in the string, the precompiler will call the compiler using the SRCSTMF parameter. There is no validation of the string. The compiler command will issue an error if any parameter is incorrect. Please refer to the DB2 for i5/OS SQL programming topic collection in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/ for a list of parameters that the precompiler generates for the compiler command. Using any of the keywords that the precompiler passes to the compiler will cause the compiler command to fail because of duplicate parameters.

***NONE**
No additional parameters will be used on the compiler command.

character-value
Specify no more than 5000 characters, enclosed in apostrophes.

Top

Examples

```
CRTSQLCBLI   OBJ(PAYROLL)  OBJTYPE(*MODULE)
             TEXT('Payroll Program')
```

This command runs the SQL precompiler which precompiles the COBOL source and stores the changed source in member PAYROLL of source file QSQLTEMP in library QTEMP. The ILE COBOL compiler is called to create module PAYROLL in the current library by using the source member created by the SQL precompiler.

Top

Error messages

***ESCAPE Messages**

SQL9001
SQL precompile failed.

SQL9002
Conflict in TGTRLS parameters for SQL precompile and &7 compile.

SQL9003
&7 Compile at wrong level for SQL source.

SQL9004
Create of SQL package failed.

SQL9006
DB2 Query Mgr and SQL DevKit not at same install level as the operating system.

Top

Create SQL ILE C object (CRTSQLCI)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Create SQL ILE C Object (CRTSQLCI) command calls the Structured Query Language (SQL) precompiler which precompiles C source containing SQL statements, produces a temporary source member, and then optionally calls the ILE C compiler to create a module, create a program, or create a service program.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Object	*Qualified object name*	Required, Positional 1
	Qualifier 1: Object	*Name*	
	Qualifier 2: Library	*Name*, ***CURLIB**	
SRCFILE	Source file	*Qualified object name*	Optional, Positional 2
	Qualifier 1: Source file	*Name*, **QCSRC**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SRCMBR	Source member	*Name*, ***OBJ**	Optional, Positional 3
SRCSTMF	Source stream file	*Path name*	Optional
COMMIT	Commitment control	***CHG**, *ALL, *CS, *NONE, *RR, *UR, *RS, *NC	Optional
RDB	Relational database	*Simple name*, ***LOCAL**, *NONE	Optional
OBJTYPE	Compile type	***MODULE**, *PGM, *SRVPGM	Optional
OUTPUT	Listing output	***NONE**, *PRINT	Optional
TEXT	Text 'description'	*Character value*, ***SRCMBRTXT**, *BLANK	Optional
USER	RDB user	*Name*, ***CURRENT**	Optional
PASSWORD	RDB user password	*Character value*, ***NONE**, ' '	Optional
OPTION	Precompiler options	Values (up to 12 repetitions): ***XREF**, *NOXREF, *GEN, *NOGEN, *CNULRQD, *NOCNULRQD, *SQL, *SYS, *JOB, *SYSVAL, *PERIOD, *COMMA, *EVENTF, *NOEVENTF, *SECLVL, *NOSECLVL, *OPTLOB, *NOOPTLOB, *NOEXTIND, *EXTIND	Optional
TGTRLS	Target release	*Simple name*, ***CURRENT**, *PRV	Optional
INCFILE	INCLUDE file	*Qualified object name*	Optional
	Qualifier 1: INCLUDE file	*Name*, ***SRCFILE**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
INCDIR	SQL INCLUDE directory	*Path name*, ***NONE**	Optional
ALWCPYDTA	Allow copy of data	***OPTIMIZE**, *YES, *NO	Optional
CLOSQLCSR	Close SQL cursor	***ENDACTGRP**, *ENDMOD	Optional
ALWBLK	Allow blocking	***ALLREAD**, *NONE, *READ	Optional
DLYPRP	Delay PREPARE	***NO**, *YES	Optional
GENLVL	Severity level	0-40, **10**	Optional

Keyword	Description	Choices	Notes
MARGINS	Source margins	*Element list*	Optional
	Element 1: Left margin	1-32754, ***SRCFILE**	
	Element 2: Right margin	1-32754	
DATFMT	Date format	***JOB**, *USA, *ISO, *EUR, *JIS, *MDY, *DMY, *YMD, *JUL	Optional
DATSEP	Date separator character	***JOB**, '/', '.', ',', '-', ' ', *BLANK	Optional
TIMFMT	Time format	***HMS**, *USA, *ISO, *EUR, *JIS	Optional
TIMSEP	Time separator character	***JOB**, ':', '.', ',', ' ', *BLANK	Optional
REPLACE	Replace	***YES**, *NO	Optional
RDBCNNMTH	RDB connect method	***DUW**, *RUW	Optional
DFTRDBCOL	Default collection	*Name*, ***NONE**	Optional
DYNDFTCOL	Dynamic default collection	***NO**, *YES	Optional
SQLPKG	Package	*Qualified object name*	Optional
	Qualifier 1: Package	*Name*, ***OBJ**	
	Qualifier 2: Library	*Name*, ***OBJLIB**	
SQLPATH	SQL path	Single values: ***NAMING**, *LIBL Other values (up to 268 repetitions): *Name*	Optional
SQLCURRULE	SQL rules	***DB2**, *STD	Optional
SAAFLAG	IBM SQL flagging	***NOFLAG**, *FLAG	Optional
FLAGSTD	ANS flagging	***NONE**, *ANS	Optional
PRTFILE	Print file	*Qualified object name*	Optional
	Qualifier 1: Print file	*Name*, **QSYSPRT**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
DBGVIEW	Debugging view	***NONE**, *SOURCE	Optional
USRPRF	User profile	***NAMING**, *USER, *OWNER	Optional
DYNUSRPRF	Dynamic user profile	***USER**, *OWNER	Optional
SRTSEQ	Sort sequence	Single values: ***JOB**, *HEX, *JOBRUN, *LANGIDUNQ, *LANGIDSHR Other values: *Qualified object name*	Optional
	Qualifier 1: Sort sequence	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
LANGID	Language id	*Character value*, ***JOB**, *JOBRUN	Optional
TOSRCFILE	To source file	*Qualified object name*	Optional
	Qualifier 1: To source file	*Name*, ***CALC**	
	Qualifier 2: Library	*Name*, **QTEMP**, *LIBL, *CURLIB	
DECRESULT	Decimal result options	*Element list*	Optional
	Element 1: Maximum precision	**31**, 63	
	Element 2: Maximum scale	0-63, **31**	
	Element 3: Minimum divide scale	0-9, **0**	
DECFLTRND	Decimal float rounding mode	***HALFEVEN**, *HALFUP, *DOWN, *CEILING, *FLOOR, *HALFDOWN, *UP	Optional
COMPILEOPT	Compiler options	*Character value*, ***NONE**	Optional

Top

Object (OBJ)

Specifies the object to be created.

This is a required parameter.

Qualifier 1: Object

name Specify the name of the object to be created.

Qualifier 2: Library

***CURLIB**
The new object is created in the current library for the job. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the object is created.

Top

Source file (SRCFILE)

Specifies the source file that contains the C source statements and SQL statements.

Qualifier 1: Source file

QCSRC
Source file QCSRC contains the C source.

name Specify the name of the source file that contains the C source. The source file can be a database file, device file, or an inline data file.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library to be searched.

Top

Source member (SRCMBR)

Specifies the source file member that contains the input source. This parameter is used only if the source file specified for the **Source file (SRCFILE)** parameter is a database file.

***OBJ** The source file member that has the same name as the value specified for the **Object (OBJ)** parameter contains the input source.

name Specify the name of the source file member that contains the input source.

Top

Source stream file (SRCSTMF)

Specifies the path name to the file that contains the C source statements and SQL statements. The path name can be either absolute or relative.

Lines cannot exceed 32754 characters.

Top

Commitment control (COMMIT)

Specifies whether SQL statements in the compiled program are run under commitment control. Files referred to in the host language source are not affected by this option. Only SQL tables, SQL views, SQL packages, SQL sequences, SQL aliases, SQL Types, SQL procedures, SQL functions, SQL indexes, SQL schemas, SQL triggers, and SQL views referred to in SQL statements are affected.

***CHG** *or* ***UR**
Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs can be seen.

***CS** Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows updated, deleted, and inserted are locked until the end of the unit of work (transaction). A row that is selected, but not updated, is locked until the next row is selected. Uncommitted changes in other jobs cannot be seen.

***ALL** *or* ***RS**
Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows selected, updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs cannot be seen.

***NONE** *or* ***NC**
Specifies that commitment control is not used. Uncommitted changes in other jobs can be seen. If the SQL DROP SCHEMA statement is included in the program, *NONE or *NC must be used. If a relational database is specified for the **Relational database (RDB)** parameter, and the relational database is on a system that is not on a System i, *NONE or *NC cannot be specified.

***RR** Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows selected, updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs cannot be seen. All tables referred to in SELECT, UPDATE, DELETE, and INSERT statements are locked exclusively until the end of the unit of work (transaction).

Top

Relational database (RDB)

Specifies the name of the relational database where the SQL package is to be created.

***LOCAL**
The program is created as a distributed SQL program. The SQL statements will access the local database. An SQL package object is not created as part of the precompile process. The Create Structured Query Language Package (CRTSQLPKG) command can be used.

***NONE**
An SQL package object is not created. The program object is not a distributed program and the Create Structured Query Language Package (CRTSQLPKG) command cannot be used.

name Specify the name of the relational database where the new SQL package object is to be created. When the name of the local relational database is specified, the program created is still a distributed SQL program. The SQL statements will access the local database.

Top

Compile type (OBJTYPE)

Specifies the type of object to be created.

When OBJTYPE(*PGM) or OBJTYPE(*SRVPGM) is specified and the **Relational database (RDB)** parameter is also specified, the CRTSQLPKG command is issued by the SQL precompiler after the program has been created. When OBJTYPE(*MODULE) is specified, an SQL package is not created and you must issue the CRTSQLPKG command after the CRTPGM or CRTSRVPGM command has created the program.

If OPTION(*NOGEN) is specified, only the SQL temporary source member is generated. No module, program, service program, or SQL package is created.

***PGM** The SQL precompiler calls the compiler to create a program.

***MODULE**
The SQL precompiler calls the compiler to create a module.

***SRVPGM**
The SQL precompiler calls the compiler to create a module and issues the Create Service Program (CRTSRVPGM) command to create a service program.

Note: If *SRVPGM is specified, there must be a source member in source file QSRVSRC that has the same name as the name specified for the **Object (OBJ)** parameter. The source member must contain the export information for the module.

Top

Listing output (OUTPUT)

Specifies whether the precompiler listing is generated.

***NONE**
The precompiler listing is not generated.

***PRINT**
The precompiler listing is generated.

Top

Text 'description' (TEXT)

Specifies text that briefly describes the program and its function.

***SRCMBRTXT**
The text is taken from the source file member being used to create the program. If the source file is an inline file or a device file, the text is blank.

***BLANK**
No text is specified.

'description'
Specify no more than 50 characters, enclosed in apostrophes.

Top

RDB user (USER)

Specifies the user name sent to the remote system when starting the conversation. This parameter is valid only when **Relational database (RDB)** is specified.

***CURRENT**
The user name associated with the current job is used.

name Specify the user name to be used for the application server job.

Top

RDB user password (PASSWORD)

Specifies the password to be used on the remote system. This parameter is valid only when **Relational database (RDB)** is specified.

***NONE**
No password is sent. A user name cannot be specified for the **RDB user (USER)** parameter if this value is specified.

Note: Specifying a password of a blank is the same as specifying *NONE.

password
Specify the password of the user name specified for the **RDB user (USER)** parameter.

Top

Precompiler options (OPTION)

Specifies whether one or more of the following options are used when the C source is precompiled. If an option is specified more than once, or if two options conflict, the last option specified is used.

Cross reference options:

***XREF** The precompiler cross-references items in the program to the statement numbers in the program that refer to those items.

***NOXREF**
The precompiler does not cross-reference names.

Program creation options:

***GEN** The precompiler creates the object that is specified for the **Compile type (OBJTYPE)** parameter.

***NOGEN**
The precompiler does not call the ILE C compiler, and a module, program, service program, or SQL package is not created.

Decimal point options:

***PERIOD**
The value used as the decimal point for numeric constants used in SQL statements is a period. This value is also used as the decimal point character when casting a numeric value to character.

***JOB** The representation for the decimal point specified for the job at precompile time is used.

Note: If QDECFMT specifies that the value used as the decimal point is a comma, any numeric constants in lists (such as in the SELECT clause or the VALUES clause) must be separated by a comma followed by a blank. For example, VALUES(1,1, 2,23, 4,1) is equivalent to VALUES(1.1,2.23,4.1) in which the decimal point is a period.

***SYSVAL**
The value used as the decimal point is the QDECFMT system value.

Note: If QDECFMT specifies that the value used as the decimal point is a comma, any numeric constants in lists (such as in the SELECT clause or the VALUES clause) must be separated by a comma followed by a blank. For example, VALUES(1,1, 2,23, 4,1) is equivalent to VALUES(1.1,2.23,4.1) in which the decimal point is a period.

***COMMA**
The value used as the decimal point is a comma.

Note: Any numeric constants in lists (such as in the SELECT clause or the VALUES clause) must be separated by a comma followed by a blank. For example, VALUES(1,1, 2,23, 4,1) is equivalent to VALUES(1.1,2.23,4.1) where the decimal point is a period.

Naming convention options:

***SYS** Specifies that the system naming convention is used (library-name/file-name).

***SQL** Specifies that the SQL naming convention is used (schema-name.table-name).

If a relational database is specified for the **Relational database (RDB)** parameter, and the database is on a system that is not a System i, *SQL must be specified as the naming convention.

Second-level message text options:

***NOSECLVL**
Second-level text descriptions are not added to the listing.

***SECLVL**
Second-level text with replacement data is added for all messages on the listing.

NUL required options:

***NOCNULRQD**
For output character and graphic host variables, the NUL-terminator is not returned when the host variable is exactly the same length as the data. Input character and graphic host variables do not require a NUL-terminator.

***CNULRQD**
Output character and graphic host variables always contain the NUL-terminator. If there is not enough space for the NUL-terminator, the data is truncated and the NUL-terminator is added. Input character and graphic host variables require a NUL-terminator.

LOB optimization for DRDA options:

***OPTLOB**
The first FETCH for a cursor determines how the cursor will be used for LOBs (Large Objects) on all subsequent FETCHes. This option remains in effect until the cursor is closed.

If the first FETCH uses a LOB locator to access a LOB column, no subsequent FETCH for that cursor can fetch that LOB column into a LOB host variable.

If the first FETCH places the LOB column into a LOB host variable, no subsequent FETCH for that cursor can use a LOB locator for that column.

***NOOPTLOB**
There is no restriction on whether a column is retrieved into a LOB locator or into a LOB host variable. This option can cause performance to degrade.

Event file creation options:

***NOEVENTF**
The compiler will not produce an event file for use by CoOperative Development Environment (CODE).

***EVENTF**
The compiler produces an event file for use by CoOperative Development Environment (CODE). The event file will be created as a member in the file EVFEVENT in your object library. CODE uses this file to offer error feedback integrated with the CODE editor. This option is normally specified by CODE on your behalf.

Extended indicators options:

***NOEXTIND**
Extended indicator support is not enabled.

***EXTIND**
Extended indicator support is enabled.

Top

Target release (TGTRLS)

Specifies the release of the operating system on which you intend to use the object being created.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

***CURRENT**
The object is to be used on the release of the operating system currently running on your system. The object can also be used on a system with any subsequent release of the operating system installed.

***PRV** The object is to be used on the previous release with modification level 0 of the operating system. The object can also be used on a system with any subsequent release of the operating system installed.

target-release
Specify the release in the format VxRxMx. The object can be used on a system with the specified release or with any subsequent release of the operating system installed.

Top

INCLUDE file (INCFILE)

Specifies the source file that contains members to be included in the program with the SQL INCLUDE statement.

Single values

***SRCFILE**
The qualified source file you specify for the **Source file (SRCFILE)** parameter contains the source file members specified on any SQL INCLUDE statements.

Qualifier 1: INCLUDE file

name Specify the name of the source file that contains the source file members specified on any SQL INCLUDE statements.

The record length of the source file you specify here must be no less than the record length of the source file you specify for the **Source file (SRCFILE)** parameter.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the source file is located.

Top

SQL INCLUDE directory (INCDIR)

Specifies the path name of the directory containing any files to be included in the program with the SQL INCLUDE statement.

Specifying a value for this parameter does not pass an INCDIR value from the SQL precompiler to the compiler. If you want to pass an INCDIR value through to the compiler, this can be done using the **Compiler options (COMPILEOPT)** parameter.

***NONE**
The current directory and the source directory will be searched.

path-name
The path name of the directory that contains the files specified on any SQL INCLUDE statement. The current directory will be searched first, then the specified path name, then the source directory.

Top

Allow copy of data (ALWCPYDTA)

Specifies whether a copy of the data can be used in a SELECT statement.

***OPTIMIZE**
The system determines whether to use the data retrieved directly from the database or to use a copy of the data. The decision is based on which method provides the best performance. If the **Commitment control (COMMIT)** parameter is not *NONE, the **Allow blocking (ALWBLK)** parameter should be set to *ALLREAD, when possible, for best performance.

***YES** A copy of the data is used only when necessary.

***NO** A copy of the data is not used. If a temporary copy of the data is required to perform the query, an error message is returned.

Top

Close SQL cursor (CLOSQLCSR)

Specifies when SQL cursors are implicitly closed, SQL prepared statements are implicitly discarded, and LOCK TABLE locks are released. SQL cursors are explicitly closed when the user issues the CLOSE, COMMIT, or ROLLBACK (without HOLD) SQL statements.

***ENDACTGRP**
SQL cursors are closed and SQL prepared statements are implicitly discarded, and LOCK TABLE locks are released when the activation group ends.

***ENDMOD**
SQL cursors are closed and SQL prepared statements are implicitly discarded when the module is exited. LOCK TABLE locks are released when the first SQL program on the call stack ends.

Top

Allow blocking (ALWBLK)

Specifies whether the database manager can use record blocking and the extent to which blocking can be used for read-only cursors.

***ALLREAD**
Rows are blocked for read-only cursors. All cursors in a program that are not explicitly able to be changed are opened for read-only processing even though there may be EXECUTE or EXECUTE IMMEDIATE statements in the program.

Specifying *ALLREAD:

- Allows record blocking for all read-only cursors.
- Can improve the performance of almost all read-only cursors in programs, but limits queries in the following ways:
  - The Rollback (ROLLBACK) command, a ROLLBACK statement in host languages, or the ROLLBACK HOLD SQL statement does not reposition a read-only cursor when *ALLREAD is specified.
  - Dynamic running of a positioned UPDATE or DELETE statement (for example, using EXECUTE IMMEDIATE), can not be used to update a row in a cursor unless the DECLARE statement for the cursor includes the FOR UPDATE clause.

***NONE**
Rows are not blocked for retrieval of data for cursors.

Specifying *NONE:

- Guarantees that the data retrieved is current.
- May reduce the amount of time required to retrieve the first row of data for a query.
- Stops the database manager from retrieving a block of data rows that is not used by the program when only the first few rows of a query are retrieved before the query is closed.
- Can degrade the overall performance of a query that retrieves a large number of rows.

***READ**
Records are blocked for read-only retrieval of data for cursors when:

- *NONE is specified for the **Commitment control (COMMIT)** parameter, which indicates that commitment control is not used.
- The cursor is declared with a FOR READ ONLY clause or there are no dynamic statements that could run a positioned UPDATE or DELETE statement for the cursor.

Top

Delay PREPARE (DLYPRP)

Specifies whether the dynamic statement validation for a PREPARE statement is delayed until an OPEN, EXECUTE, or DESCRIBE statement is run. Delaying validation improves performance by eliminating redundant validation.

***NO** Dynamic statement validation is not delayed. When the dynamic statement is prepared, the access plan is validated. When the dynamic statement is used in an OPEN or EXECUTE statement, the access plan is revalidated. Because the authority or the existence of objects referred to by the dynamic statement may change, you must still check the SQLCODE or SQLSTATE after issuing the OPEN or EXECUTE statement to ensure that the dynamic statement is still valid.

***YES** Dynamic statement validation is delayed until the dynamic statement is used in an OPEN, EXECUTE, or DESCRIBE SQL statement. When the dynamic statement is used, the validation is completed and an access plan is built. If you specify *YES for this parameter for precompiled programs, you should check the SQLCODE and SQLSTATE after running an OPEN, EXECUTE, or DESCRIBE statement to ensure that the dynamic statement is valid.

Note: If you specify *YES, performance is not improved if the INTO clause is used on the PREPARE statement or if a DESCRIBE statement uses the dynamic statement before an OPEN is issued for the statement.

Top

Severity level (GENLVL)

Specifies whether the compiler is called, depending on the severity of messages generated as a result of errors found by the SQL precompiler. If precompiler errors are generated that have a message severity level greater than the value specified for this parameter, the compiler is not called.

If the **Relational database (RDB)** parameter is specified and the severity of the messages generated as a result of package creation is greater than the severity level specified for this parameter, the SQL package is not created.

10 Do not call the compiler if SQL precompiler messages with a message severity greater than 10 are generated.

0-40 Specify the maximum SQL precompiler message severity level to be used to control whether the compiler is called.

Top

Source margins (MARGINS)

Specifies the part of the precompiler input record that contains source text.

If the **Source stream file (SRCSTMF)** parameter is specified, margins are ignored.

Element 1: Left margin

***SRCFILE**
The left margin will be set to 1 and the right margin will be set to the record length of the input source file.

1-32754
Specify the beginning position to be used for each input record.

Element 2: Right margin

1-32754
Specify the ending position to be used for each input record.

Top

Date format (DATFMT)

Specifies the format used when accessing date result columns. All output date fields are returned in the format you specify. For input date strings, the value you specify is used to determine whether the date is a valid format.

Note: An input date string that uses the format *USA, *ISO, *EUR, or *JIS is always valid.

If you connect to a relational database that is on a system that is not a System i, the format must be *USA, *ISO, *EUR, or *JIS.

***JOB** The format specified for the job at precompile time or when a new interactive SQL session is created is used.

Use the Display Job (DSPJOB) command to determine the current date format for the job.

***USA** The United States date format **mm/dd/yyyy** is used.

***ISO** The International Organization for Standardization (ISO) date format **yyyy-mm-dd** is used.

***EUR** The European date format **dd.mm.yyyy** is used.

***JIS** The Japanese Industrial Standard date format **yyyy-mm-dd** is used.

***MDY** The date format **mm/dd/yy** is used.

***DMY** The date format **dd/mm/yy** is used.

***YMD** The date format **yy/mm/dd** is used.

***JUL** The Julian date format **yy/ddd** is used.

Top

Date separator character (DATSEP)

Specifies the separator to be used when accessing date result columns.

Note: This parameter applies only when *JOB, *MDY, *DMY, *YMD, or *JUL is specified for the **Date format (DATFMT)** parameter.

***JOB** The date separator specified for the job at precompile time, when a new interactive SQL session is created, or when Run SQL Statement (RUNSQLSTM) command is run.

Use the Display Job (DSPJOB) command to determine the current date separator value for the job.

'/' A slash is used as the date separator.

'.' A period is used as the date separator.

'-' A dash is used as the date separator.

',' A comma is used as the date separator.

' ' *or* *BLANK
A blank is used as the date separator.

Top

Time format (TIMFMT)

Specifies the format used when accessing time result columns. All output time fields are returned in the format you specify. For input time strings, the value you specify is used to determine whether the time is specified in a valid format.

Note: An input time string that uses the format *USA, *ISO, *EUR, or *JIS is always valid.

If you connect to a relational database that is on a system that is not another System i, the time format must be *USA, *ISO, *EUR, *JIS, or *HMS with a time separator of a colon or period.

***HMS** The **hh:mm:ss** format is used.

***USA** The United States time format **hh:mmxx** is used, where **xx** is AM or PM.

***ISO** The International Organization for Standardization (ISO) time format **hh.mm.ss** is used.

***EUR** The European time format **hh.mm.ss** is used.

***JIS** The Japanese Industrial Standard time format **hh:mm:ss** is used.

Top

Time separator character (TIMSEP)

Specifies the separator used when accessing time result columns.

Note: This parameter applies only when *HMS is specified for the **Time format (TIMFMT)** parameter.

***JOB** The time separator specified for the job at precompile time, when a new interactive SQL session is created, or when RUNSQLSTM is run is used.

Use the Display Job (DSPJOB) command to determine the current time separator value for the job.

':' A colon is used as the time separator.

'.' A period is used as the time separator.

',' A comma is used as the time separator.

' ' *or* *BLANK
A blank is used as the time separator.

Top

Replace (REPLACE)

Specifies if a SQL module, program, service program or package is created when there is an existing SQL module, program, service program, or package of the same name and type in the same library. The value for this parameter is passed to the CRTCMOD, CRTBNDC, CRTSRVPGM, and CRTSQLPKG commands.

***YES** A new SQL module, program, service program, or package is created, and any existing object of the same name and type in the specified library is moved to QRPLOBJ.

***NO** A new SQL module, program, service program, or package is not created if an object of the same name and type already exists in the specified library.

Top

RDB connect method (RDBCNNMTH)

Specifies the semantics used for CONNECT statements.

***DUW**
CONNECT (Type 2) semantics are used to support distributed unit of work. Consecutive CONNECT statements to additional relational databases do not result in disconnection of previous connections.

***RUW** CONNECT (Type 1) semantics are used to support remote unit of work. Consecutive CONNECT statements result in the previous connection being disconnected before a new connection is established.

Top

Default collection (DFTRDBCOL)

Specifies the name of the schema identifier used for the unqualified names of the tables, views, indexes, SQL packages, aliases, constraints, external programs, node groups, and triggers. This parameter applies only to static SQL statements.

***NONE**
The naming convention specified for the **Precompiler options (OPTION)** parameter is used.

name Specify the name of the schema identifier to be used instead of the naming convention specified for the **Precompiler options (OPTION)** parameter.

Top

Dynamic default collection (DYNDFTCOL)

Specifies whether the default schema name specified for the **Default collection (DFTRDBCOL)** parameter is also used for dynamic statements.

***NO** Do not use the value specified for the **Default collection (DFTRDBCOL)** parameter for unqualified names of tables, views, indexes, SQL packages, aliases, constraints, external programs, node groups, and triggers for dynamic SQL statements. The naming convention specified for the **Precompiler options (OPTION)** parameter is used.

***YES** The schema name specified for the **Default collection (DFTRDBCOL)** parameter will be used for the unqualified names of the tables, views, indexes, SQL packages, aliases, constraints, external programs, node groups, and triggers in dynamic SQL statements.

Top

Package (SQLPKG)

Specifies the qualified name of the SQL package created on the relational database specified for the **Relational database (RDB)** parameter of this command.

Qualifier 1: Package

***OBJ** The name of the SQL package is the same as the object name specified for the **Object (OBJ)** parameter.

name Specify the name of the SQL package. If the remote system is not a System i, no more than 8 characters can be specified.

Qualifier 2: Library

***OBJLIB**
The package is created in the library with the same name as the library specified for the **Object (OBJ)** parameter.

name Specify the name of the library where the package is created.

Top

SQL path (SQLPATH)

Specifies the path to be used to find procedures, functions, and user defined types in static SQL statements.

***NAMING**
The path used depends on the naming convention specified for the **Precompiler options (OPTION)** parameter.

For *SYS naming, the path used is *LIBL, the current library list at runtime.

For *SQL naming, the path used is ″QSYS″, ″QSYS2″, ″userid″, where ″userid″ is the value of the USER special register. If a schema name is specified for the **Default collection (DFTRDBCOL)** parameter, the schema name takes the place of userid.

***LIBL** The path used is the library list at runtime.

name Specify one or more schema names. A maximum of 268 schema names may be specified.

Top

SQL rules (SQLCURRULE)

Specifies the semantics used for SQL statements.

***DB2** The semantics of all SQL statements will default to the rules established for DB2. The following semantics are controlled by this option:

Hexadecimal constants are treated as character data.

***STD** The semantics of all SQL statements will default to the rules established by the ISO and ANSI SQL standards. The following semantics are controlled by this option:

Hexadecimal constants are treated as binary data.

Top

IBM SQL flagging (SAAFLAG)

Specifies the IBM SQL flagging function. This parameter allows you to flag SQL statements to verify whether they conform to IBM SQL syntax.

***NOFLAG**
No checks are made to see whether SQL statements conform to IBM SQL syntax.

***FLAG**
Checks are made to see whether SQL statements conform to IBM SQL syntax.

Top

ANS flagging (FLAGSTD)

Specifies whether non-standard statements are flagged. This parameter allows you to flag SQL statements to verify whether they conform to the Core level of the ISO/IEC 9075-2003 standards.

***NONE**
No checks are made to see whether SQL statements conform to ANSI standards.

***ANS** Checks are made to see whether SQL statements conform to standards.

Top

Print file (PRTFILE)

Specifies the printer device file to be used for the precompiler output listing.

Qualifier 1: Print file

QSYSPRT
The precompiler output file is directed to the IBM-supplied printer file, QSYSPRT. The file QSYSPRT has a record length of 132. If you specify a file whose record length is less than 132, information is lost.

name Specify the name of the printer device file to which the precompiler output is directed.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the printer file is located.

Top

Debugging view (DBGVIEW)

Specifies the type of source debug information to be provided by the SQL precompiler.

***NONE**
The source view will not be generated.

***SOURCE**
The SQL precompiler will provide the source views for the root and if necessary, SQL INCLUDE statements. A view is to be provided which contains the statements generated by the precompiler.

Top

User profile (USRPRF)

Specifies the user profile that is used when the compiled program object and SQL package object is run, including the authority that the program object or SQL package has for each object in static SQL statements. The profile of either the owner or the user is used to control access to objects.

***NAMING**
The user profile is determined by the naming convention. If the naming convention is *SQL, USRPRF(*OWNER) is used. If the naming convention is *SYS, USRPRF(*USER) is used.

***USER**
The profile of the user running the program or SQL package is used.

***OWNER**
The user profiles of both the owner and the user are used when the program or SQL package is run.

Top

Dynamic user profile (DYNUSRPRF)

Specifies the user profile used for dynamic SQL statements.

***USER**
Local dynamic SQL statements are run under the profile of the program's user. Distributed dynamic SQL statements are run under the profile of the application server job.

***OWNER**
Local dynamic SQL statements are run under the profile of the program's owner. Distributed dynamic SQL statements are run under the profile of the SQL package's owner.

Top

Sort sequence (SRTSEQ)

Specifies the sort sequence table to be used for string comparisons in SQL statements.

Note: *HEX must be specified for this parameter on distributed applications where the application server is not on a System i.

Single values

***JOB** The SRTSEQ value for the job is used.

***JOBRUN**
The SRTSEQ value for the job is retrieved when the program is run. For distributed applications, SRTSEQ(*JOBRUN) is valid only when LANGID(*JOBRUN) is also specified.

***LANGIDUNQ**
The unique-weight sort table for the language specified for the **Language id (LANGID)** parameter is used.

***LANGIDSHR**
The shared-weight sort table for the language specified for the LANGID parameter is used.

***HEX** A sort sequence table is not used. The hexadecimal values of the characters are used to determine the sort sequence.

Qualifier 1: Sort sequence

name Specify the name of the sort sequence table to be used with this program.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library to be searched.

Top

Language id (LANGID)

Specifies the language identifier to be used when SRTSEQ(*LANGIDUNQ) or SRTSEQ(*LANGIDSHR) is specified.

***JOB** The LANGID value for the job is retrieved during the precompile.

***JOBRUN**
The LANGID value for the job is retrieved when the program is run. For distributed applications, LANGID(*JOBRUN) is valid only when SRTSEQ(*JOBRUN) is also specified.

language-id
Specify the language identifier to be used by the program.

Top

To source file (TOSRCFILE)

Specifies the source file that is to contain the output source member that has been processed by the SQL precompiler. If the specified source file is not found, it will be created. The output member will have the same name as the name specified for the OBJECT parameter.

Qualifier 1: To source file

***CALC**
The output source file name will be generated based on the margins of the source file. The name will be QSQLTxxxxx, where xxxxx is the width of the source file. If the source file record length is less than or equal to 92, the name will be QSQLTEMP.

name Specify the name of the source file to contain the output source member.

Qualifier 2: Library

QTEMP
The library QTEMP will be used.

***LIBL** The job's library list is searched for the specified file. If the file is not found in any library in the library list, the file will be created in the current library.

***CURLIB**
The current library for the job will be used. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library that is to contain the output source file.

Top

Decimal result options (DECRESULT)

Specifies the maximum precision, maximum scale and minimum divide scale that should be returned for result data types. The specified limit only applies to numeric (zoned) and decimal (packed) data types used in arithmetic expressions and in SQL column functions AVG and SUM.

Element 1: Maximum precision

31 The maximum precision (length) that should be returned for the result data types is 31 digits.

63 The maximum precision (length) that should be returned for the result data types is 63 digits.

Element 2: Maximum scale

31 The maximum scale (number of decimal positions to the right of the decimal point) that should be returned for the result data types is 31 digits.

0-63 Specify the maximum scale (number of decimal positions to the right of the decimal point) that should be returned for the result data types. The value can range from 0 to the maximum precision.

Element 3: Minimum divide scale

0 The minimum divide scale (number of decimal positions to the right of the decimal point) that should be returned for both intermediate and result data types is 0.

0-9 Specify the minimum divide scale (number of decimal positions to the right of the decimal point) that should be returned for both intermediate and result data types. The value cannot to exceed the maximum scale. If 0 is specified for the maximum scale, minimum divide scale is not used.

Top

Decimal float rounding mode (DECFLTRND)

Specifies the decimal floating point rounding mode used for static SQL statements.

***HALFEVEN**
Round to nearest; if equidistant, round so that the final digit is even. If the discarded digits represent greater than half (0.5) the value of a one in the next left position then the result coefficient should be incremented by 1 (rounded up). If they represent less than half, then the result coefficient is not adjusted (that is, the discarded digits are ignored). Otherwise (they represent exactly half) the result coefficient is unaltered if its rightmost digit is even, or incremented by 1 (rounded up) if its rightmost digit is odd (to make an even digit).

***HALFUP**
Round to nearest; if equidistant, round up. If the discarded digits represent greater than or equal to half (0.5) of the value of a one in the next left position then the result coefficient should be incremented by 1 (rounded up). Otherwise the discarded digits are ignored.

***DOWN**
Round towards 0 (truncation). The discarded digits are ignored.

***CEILING**
Round towards +infinity. If all of the discarded digits are zero or if the sign is negative the result is unchanged other than the removal of discarded digits. Otherwise, the result coefficient should be incremented by 1 (rounded up).

***FLOOR**
Round towards -infinity. If all of the discarded digits are zero or if the sign is positive the result is unchanged other than the removal of discarded digits. Otherwise, the sign is negative and the result coefficient should be incremented by 1.

***HALFDOWN**
Round to nearest; if equidistant, round down. If the discarded digits represent greater than half (0.5) of the value of a one in the next left position then the result coefficient should be incremented by 1 (rounded up). Otherwise (the discarded digits are 0.5 or less) the discarded digits are ignored.

***UP** Round away from 0. Of all of the discarded digits are zero the result is unchanged other than the removal of discarded digits. Otherwise, the result coefficient should be incremented by 1 (rounded up).

Top

Compiler options (COMPILEOPT)

Specifies additional parameters to be used on the compiler command. The COMPILEOPT string is added to the compiler command built by the precompiler. If **INCDIR(** is anywhere in the string, the precompiler will call the compiler using the SRCSTMF parameter. There is no validation of the string. The compiler command will issue an error if any parameter is incorrect. Please refer to the DB2 for i5/OS SQL programming topic collection in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/ for a list of parameters that the precompiler generates for the compiler command. Using any of the keywords that the precompiler passes to the compiler will cause the compiler command to fail because of duplicate parameters.

***NONE**
No additional parameters will be used on the compiler command.

character-value
Specify no more than 5000 characters, enclosed in apostrophes.

Top

Examples

```
CRTSQLCI   OBJ(PAYROLL)  OBJTYPE(*MODULE)
           TEXT('Payroll Program')
```

This command runs the SQL precompiler which precompiles the C source and stores the changed source in member PAYROLL of source file QSQLTEMP in library QTEMP. The ILE C compiler is called to create module PAYROLL in the current library by using the source member created by the SQL precompiler.

Top

Error messages

***ESCAPE Messages**

SQL9001
SQL precompile failed.

SQL9002
Conflict in TGTRLS parameters for SQL precompile and &7 compile.

SQL9003
&7 Compile at wrong level for SQL source.

SQL9004
Create of SQL package failed.

SQL9006
DB2 Query Mgr and SQL DevKit not at same install level as the operating system.

Top

Create SQL ILE C++ Object (CRTSQLCPPI)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Create SQL ILE C++ Object (CRTSQLCPPI) command calls the Structured Query Language (SQL) precompiler which precompiles C++ source containing SQL statements, produces a temporary source member, and then optionally calls the ILE C++ compiler to create a module.

Since this command only creates a module, the user must issue the CRTSQLPKG command after the CRTPGM or CRTSRVPGM command has created the program if an SQL package is needed.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Object	*Qualified object name*	Required, Positional 1
	Qualifier 1: Object	*Name*	
	Qualifier 2: Library	*Name*, ***CURLIB**	
SRCFILE	Source file	*Qualified object name*	Optional, Positional 2
	Qualifier 1: Source file	*Name*, **QCSRC**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SRCMBR	Source member	*Name*, ***OBJ**	Optional, Positional 3
SRCSTMF	Source stream file	*Path name*	Optional
COMMIT	Commitment control	***CHG**, *ALL, *CS, *NONE, *RR, *UR, *RS, *NC	Optional
RDB	Relational database	*Simple name*, ***LOCAL**, *NONE	Optional
OUTPUT	Listing output	***NONE**, *PRINT	Optional
TEXT	Text 'description'	*Character value*, ***SRCMBRTXT**, *BLANK	Optional
USER	RDB user	*Name*, ***CURRENT**	Optional
PASSWORD	RDB user password	*Character value*, ***NONE**, ' '	Optional
OPTION	Precompiler options	Values (up to 12 repetitions): ***XREF**, *NOXREF, *GEN, *NOGEN, *CNULRQD, *NOCNULRQD, *SQL, *SYS, *JOB, *SYSVAL, *PERIOD, *COMMA, *EVENTF, *NOEVENTF, *SECLVL, *NOSECLVL, *OPTLOB, *NOOPTLOB, *NOEXTIND, *EXTIND	Optional
TGTRLS	Target release	*Simple name*, ***CURRENT**, *PRV	Optional
INCFILE	INCLUDE file	*Qualified object name*	Optional
	Qualifier 1: INCLUDE file	*Name*, ***SRCFILE**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
INCDIR	SQL INCLUDE directory	*Path name*, ***NONE**	Optional
ALWCPYDTA	Allow copy of data	***OPTIMIZE**, *YES, *NO	Optional
CLOSQLCSR	Close SQL cursor	***ENDACTGRP**, *ENDMOD	Optional
ALWBLK	Allow blocking	***ALLREAD**, *NONE, *READ	Optional
DLYPRP	Delay PREPARE	***NO**, *YES	Optional
GENLVL	Severity level	0-40, **10**	Optional

Keyword	Description	Choices	Notes
MARGINS	Source margins	*Element list*	Optional
	Element 1: Left margin	1-32754, ***SRCFILE**	
	Element 2: Right margin	1-32754	
DATFMT	Date format	***JOB**, *USA, *ISO, *EUR, *JIS, *MDY, *DMY, *YMD, *JUL	Optional
DATSEP	Date separator character	***JOB**, '/', '.', ',', '-', ' ', *BLANK	Optional
TIMFMT	Time format	***HMS**, *USA, *ISO, *EUR, *JIS	Optional
TIMSEP	Time separator character	***JOB**, ':', '.', ',', ' ', *BLANK	Optional
REPLACE	Replace	***YES**, *NO	Optional
RDBCNNMTH	RDB connect method	***DUW**, *RUW	Optional
DFTRDBCOL	Default collection	*Name*, ***NONE**	Optional
DYNDFTCOL	Dynamic default collection	***NO**, *YES	Optional
SQLPKG	Package	*Qualified object name*	Optional
	Qualifier 1: Package	*Name*, ***OBJ**	
	Qualifier 2: Library	*Name*, ***OBJLIB**	
SQLPATH	SQL path	Single values: ***NAMING**, *LIBL Other values (up to 268 repetitions): *Name*	Optional
SQLCURRULE	SQL rules	***DB2**, *STD	Optional
SAAFLAG	IBM SQL flagging	***NOFLAG**, *FLAG	Optional
FLAGSTD	ANS flagging	***NONE**, *ANS	Optional
PRTFILE	Print file	*Qualified object name*	Optional
	Qualifier 1: Print file	*Name*, **QSYSPRT**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
DBGVIEW	Debugging view	***NONE**, *SOURCE	Optional
USRPRF	User profile	***NAMING**, *USER, *OWNER	Optional
DYNUSRPRF	Dynamic user profile	***USER**, *OWNER	Optional
SRTSEQ	Sort sequence	Single values: ***JOB**, *HEX, *JOBRUN, *LANGIDUNQ, *LANGIDSHR Other values: *Qualified object name*	Optional
	Qualifier 1: Sort sequence	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
LANGID	Language id	*Character value*, ***JOB**, *JOBRUN	Optional
TOSRCFILE	To source file	*Qualified object name*	Optional
	Qualifier 1: To source file	*Name*, ***CALC**	
	Qualifier 2: Library	*Name*, **QTEMP**, *LIBL, *CURLIB	
DECRESULT	Decimal result options	*Element list*	Optional
	Element 1: Maximum precision	**31**, 63	
	Element 2: Maximum scale	0-63, **31**	
	Element 3: Minimum divide scale	0-9, **0**	
DECFLTRND	Decimal float rounding mode	***HALFEVEN**, *HALFUP, *DOWN, *CEILING, *FLOOR, *HALFDOWN, *UP	Optional
COMPILEOPT	Compiler options	*Character value*, ***NONE**	Optional

Top

Object (OBJ)

Specifies the object to be created.

This is a required parameter.

Qualifier 1: Object

name Specify the name of the object to be created.

Qualifier 2: Library

***CURLIB**
The new object is created in the current library for the job. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the object is created.

Top

Source file (SRCFILE)

Specifies the source file that contains the C++ source statements and SQL statements.

Qualifier 1: Source file

QCSRC
Source file QCSRC contains the C++ source.

name Specify the name of the source file that contains the C++ source.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library to be searched.

Top

Source member (SRCMBR)

Specifies the source file member that contains the input source. This parameter is used only if the source file specified for the **Source file (SRCFILE)** parameter is a database file.

***OBJ** The source file member that has the same name as the value specified for the **Object (OBJ)** parameter contains the input source.

name Specify the name of the source file member that contains the input source.

Top

Source stream file (SRCSTMF)

Specifies the path name to the file that contains the C++ source statements and SQL statements. The path name can be either absolute or relative.

Lines cannot exceed 32754 characters.

Top

Commitment control (COMMIT)

Specifies whether SQL statements in the compiled program are run under commitment control. Files referred to in the host language source are not affected by this option. Only SQL tables, SQL views, SQL packages, SQL sequences, SQL aliases, SQL Types, SQL procedures, SQL functions, SQL indexes, SQL schemas, SQL triggers, and SQL views referred to in SQL statements are affected.

***CHG** *or* ***UR**
Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs can be seen.

***CS** Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows updated, deleted, and inserted are locked until the end of the unit of work (transaction). A row that is selected, but not updated, is locked until the next row is selected. Uncommitted changes in other jobs cannot be seen.

***ALL** *or* ***RS**
Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows selected, updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs cannot be seen.

***NONE** *or* ***NC**
Specifies that commitment control is not used. Uncommitted changes in other jobs can be seen. If the SQL DROP SCHEMA statement is included in the program, *NONE or *NC must be used. If a relational database is specified for the **Relational database (RDB)** parameter, and the relational database is on a system that is not on a System i, *NONE or *NC cannot be specified.

***RR** Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows selected, updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs cannot be seen. All tables referred to in SELECT, UPDATE, DELETE, and INSERT statements are locked exclusively until the end of the unit of work (transaction).

Top

Relational database (RDB)

Specifies the name of the relational database where the SQL package is to be created.

***LOCAL**
The program is created as a distributed SQL program. The SQL statements will access the local database. An SQL package object is not created as part of the precompile process. The Create Structured Query Language Package (CRTSQLPKG) command can be used.

***NONE**
An SQL package object is not created. The program object is not a distributed program and the Create Structured Query Language Package (CRTSQLPKG) command cannot be used.

name Specify the name of the relational database where the new SQL package object is to be created. When the name of the local relational database is specified, the program created is still a distributed SQL program. The SQL statements will access the local database.

Top

Listing output (OUTPUT)

Specifies whether the precompiler listing is generated.

***NONE**
The precompiler listing is not generated.

***PRINT**
The precompiler listing is generated.

Top

Text 'description' (TEXT)

Specifies text that briefly describes the program and its function.

***SRCMBRTXT**
The text is taken from the source file member being used to create the program. If the source file is an inline file or a device file, the text is blank.

***BLANK**
No text is specified.

'description'
Specify no more than 50 characters, enclosed in apostrophes.

Top

RDB user (USER)

Specifies the user name sent to the remote system when starting the conversation. This parameter is valid only when **Relational database (RDB)** is specified.

***CURRENT**
The user name associated with the current job is used.

name Specify the user name to be used for the application server job.

Top

RDB user password (PASSWORD)

Specifies the password to be used on the remote system. This parameter is valid only when **Relational database (RDB)** is specified.

***NONE**
No password is sent. A user name cannot be specified for the **RDB user (USER)** parameter if this value is specified.

Note: Specifying a password of a blank is the same as specifying *NONE.

password
Specify the password of the user name specified for the **RDB user (USER)** parameter.

Top

Precompiler options (OPTION)

Specifies whether one or more of the following options are used when the C++ source is precompiled. If an option is specified more than once, or if two options conflict, the last option specified is used.

Cross reference options:

***XREF** The precompiler cross-references items in the program to the statement numbers in the program that refer to those items.

***NOXREF**
The precompiler does not cross-reference names.

Program creation options:

***GEN** The precompiler creates the module object.

***NOGEN**
The precompiler does not call the C++ compiler. No module is created.

Decimal point options:

***JOB** The representation for the decimal point specified for the job at precompile time is used.

Note: If QDECFMT specifies that the value used as the decimal point is a comma, any numeric constants in lists (such as in the SELECT clause and VALUES clause) must be separated by a comma followed by a blank. For example, VALUES(1,1, 2,23, 4,1) is equivalent to VALUES(1.1,2.23,4.1) where the decimal point is the period.

***SYSVAL**
The value used as the decimal point in numeric constants is from the QDECFMT system value. This value is also used as the decimal point character when casting a numeric value to character.

Note: If QDECFMT specifies that the value used as the decimal point is a comma; any numeric constants in lists (such as in the SELECT clause and VALUES clause) must be separated by a comma followed by a blank. For example, VALUES(1,1, 2,23, 4,1) is equivalent to VALUES(1.1,2.23,4.1) where the decimal point is the period.

***PERIOD**
The value used as the decimal point for numeric constants used in SQL statements is a period. This value is also used as the decimal point character when casting a numeric value to character.

***COMMA**
The value used as the decimal point in numeric constants is a comma. Any numeric constants in lists (such as in the SELECT clause and VALUES clause) must be separated by a comma followed by a blank. For example, VALUES(1,1, 2,23, 4,1) is equivalent to VALUES(1.1,2.23,4.1) where the decimal point is the period.

Naming convention options:

***SYS** Specifies that the system naming convention is used (library-name/file-name).

***SQL** Specifies that the SQL naming convention is used (schema-name.table-name).

If a relational database is specified for the **Relational database (RDB)** parameter, and the database is on a system that is not a System i, *SQL must be specified as the naming convention.

Second-level message text options:

***NOSECLVL**
Second-level text descriptions are not added to the listing.

***SECLVL**
Second-level text with replacement data is added for all messages on the listing.

NUL required options:

***NOCNULRQD**
For output character and graphic host variables, the NUL-terminator is not returned when the host variable is exactly the same length as the data. Input character and graphic host variables do not require a NUL-terminator.

***CNULRQD**
Output character and graphic host variables always contain the NUL-terminator. If there is not enough space for the NUL-terminator, the data is truncated and the NUL-terminator is added. Input character and graphic host variables require a NUL-terminator.

LOB optimization for DRDA options:

***OPTLOB**
The first FETCH for a cursor determines how the cursor will be used for LOBs (Large Objects) on all subsequent FETCHes. This option remains in effect until the cursor is closed.

If the first FETCH uses a LOB locator to access a LOB column, no subsequent FETCH for that cursor can fetch that LOB column into a LOB host variable.

If the first FETCH places the LOB column into a LOB host variable, no subsequent FETCH for that cursor can use a LOB locator for that column.

***NOOPTLOB**
There is no restriction on whether a column is retrieved into a LOB locator or into a LOB host variable. This option can cause performance to degrade.

Event file creation options:

***NOEVENTF**
The compiler will not produce an event file for use by CoOperative Development Environment (CODE).

***EVENTF**
The compiler produces an event file for use by CoOperative Development Environment (CODE). The event file will be created as a member in the file EVFEVENT in your object library. CODE uses this file to offer error feedback integrated with the CODE editor. This option is normally specified by CODE on your behalf.

Extended indicators options:

***NOEXTIND**
Extended indicator support is not enabled.

***EXTIND**
Extended indicator support is enabled.

Top

Target release (TGTRLS)

Specifies the release of the operating system on which you intend to use the object being created.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

***CURRENT**
The object is to be used on the release of the operating system currently running on your system. The object can also be used on a system with any subsequent release of the operating system installed.

***PRV** The object is to be used on the previous release with modification level 0 of the operating system. The object can also be used on a system with any subsequent release of the operating system installed.

target-release
Specify the release in the format VxRxMx. The object can be used on a system with the specified release or with any subsequent release of the operating system installed.

Top

INCLUDE file (INCFILE)

Specifies the source file that contains members to be included in the program with the SQL INCLUDE statement.

Single values

***SRCFILE**
The qualified source file you specify for the **Source file (SRCFILE)** parameter contains the source file members specified on any SQL INCLUDE statements.

Qualifier 1: INCLUDE file

name Specify the name of the source file that contains the source file members specified on any SQL INCLUDE statements.

The record length of the source file you specify here must be no less than the record length of the source file you specify for the **Source file (SRCFILE)** parameter.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the source file is located.

Top

SQL INCLUDE directory (INCDIR)

Specifies the path name of the directory containing any files to be included in the program with the SQL INCLUDE statement.

Specifying a value for this parameter does not pass an INCDIR value from the SQL precompiler to the compiler. If you want to pass an INCDIR value through to the compiler, this can be done using the **Compiler options (COMPILEOPT)** parameter.

***NONE**
The current directory and the source directory will be searched.

path-name
The path name of the directory that contains the files specified on any SQL INCLUDE statement. The current directory will be searched first, then the specified path name, then the source directory.

Top

Allow copy of data (ALWCPYDTA)

Specifies whether a copy of the data can be used in a SELECT statement.

***OPTIMIZE**
The system determines whether to use the data retrieved directly from the database or to use a copy of the data. The decision is based on which method provides the best performance. If the **Commitment control (COMMIT)** parameter is not *NONE, the **Allow blocking (ALWBLK)** parameter should be set to *ALLREAD, when possible, for best performance.

***YES** A copy of the data is used only when necessary.

***NO** A copy of the data is not used. If a temporary copy of the data is required to perform the query, an error message is returned.

Top

Close SQL cursor (CLOSQLCSR)

Specifies when SQL cursors are implicitly closed, SQL prepared statements are implicitly discarded, and LOCK TABLE locks are released. SQL cursors are explicitly closed when the user issues the CLOSE, COMMIT, or ROLLBACK (without HOLD) SQL statements.

***ENDACTGRP**
SQL cursors are closed and SQL prepared statements are implicitly discarded, and LOCK TABLE locks are released when the activation group ends.

***ENDMOD**
SQL cursors are closed and SQL prepared statements are implicitly discarded when the module is exited. LOCK TABLE locks are released when the first SQL program on the call stack ends.

Top

Allow blocking (ALWBLK)

Specifies whether the database manager can use record blocking and the extent to which blocking can be used for read-only cursors.

***ALLREAD**

Rows are blocked for read-only cursors. All cursors in a program that are not explicitly able to be changed are opened for read-only processing even though there may be EXECUTE or EXECUTE IMMEDIATE statements in the program.

Specifying *ALLREAD:

- Allows record blocking for all read-only cursors.
- Can improve the performance of almost all read-only cursors in programs, but limits queries in the following ways:
  - The Rollback (ROLLBACK) command, a ROLLBACK statement in host languages, or the ROLLBACK HOLD SQL statement does not reposition a read-only cursor when *ALLREAD is specified.
  - Dynamic running of a positioned UPDATE or DELETE statement (for example, using EXECUTE IMMEDIATE), can not be used to update a row in a cursor unless the DECLARE statement for the cursor includes the FOR UPDATE clause.

***NONE**

Rows are not blocked for retrieval of data for cursors.

Specifying *NONE:

- Guarantees that the data retrieved is current.
- May reduce the amount of time required to retrieve the first row of data for a query.
- Stops the database manager from retrieving a block of data rows that is not used by the program when only the first few rows of a query are retrieved before the query is closed.
- Can degrade the overall performance of a query that retrieves a large number of rows.

***READ**

Records are blocked for read-only retrieval of data for cursors when:

- *NONE is specified for the **Commitment control (COMMIT)** parameter, which indicates that commitment control is not used.
- The cursor is declared with a FOR READ ONLY clause or there are no dynamic statements that could run a positioned UPDATE or DELETE statement for the cursor.

Top

Delay PREPARE (DLYPRP)

Specifies whether the dynamic statement validation for a PREPARE statement is delayed until an OPEN, EXECUTE, or DESCRIBE statement is run. Delaying validation improves performance by eliminating redundant validation.

***NO** Dynamic statement validation is not delayed. When the dynamic statement is prepared, the access plan is validated. When the dynamic statement is used in an OPEN or EXECUTE statement, the access plan is revalidated. Because the authority or the existence of objects referred to by the dynamic statement may change, you must still check the SQLCODE or SQLSTATE after issuing the OPEN or EXECUTE statement to ensure that the dynamic statement is still valid.

***YES** Dynamic statement validation is delayed until the dynamic statement is used in an OPEN, EXECUTE, or DESCRIBE SQL statement. When the dynamic statement is used, the validation is completed and an access plan is built. If you specify *YES for this parameter for precompiled programs, you should check the SQLCODE and SQLSTATE after running an OPEN, EXECUTE, or DESCRIBE statement to ensure that the dynamic statement is valid.

Note: If you specify *YES, performance is not improved if the INTO clause is used on the PREPARE statement or if a DESCRIBE statement uses the dynamic statement before an OPEN is issued for the statement.

Top

Severity level (GENLVL)

Specifies whether the compiler is called, depending on the severity of messages generated as a result of errors found by the SQL precompiler. If precompiler errors are generated that have a message severity level greater than the value specified for this parameter, the compiler is not called.

If the **Relational database (RDB)** parameter is specified and the severity of the messages generated as a result of package creation is greater than the severity level specified for this parameter, the SQL package is not created.

10 Do not call the compiler if SQL precompiler messages with a message severity greater than 10 are generated.

0-40 Specify the maximum SQL precompiler message severity level to be used to control whether the compiler is called.

Top

Source margins (MARGINS)

Specifies the part of the precompiler input record that contains source text.

If the **Source stream file (SRCSTMF)** parameter is specified, margins are ignored.

Element 1: Left margin

***SRCFILE**
The left margin will be set to 1 and the right margin will be set to the record length of the input source file.

1-32754
Specify the beginning position to be used for each input record.

Element 2: Right margin

1-32754
Specify the ending position to be used for each input record.

Top

Date format (DATFMT)

Specifies the format used when accessing date result columns. All output date fields are returned in the format you specify. For input date strings, the value you specify is used to determine whether the date is a valid format.

Note: An input date string that uses the format *USA, *ISO, *EUR, or *JIS is always valid.

If you connect to a relational database that is on a system that is not a System i, the format must be *USA, *ISO, *EUR, or *JIS.

***JOB** The format specified for the job at precompile time or when a new interactive SQL session is created is used.

Use the Display Job (DSPJOB) command to determine the current date format for the job.

***USA** The United States date format **mm/dd/yyyy** is used.

***ISO** The International Organization for Standardization (ISO) date format **yyyy-mm-dd** is used.

***EUR** The European date format **dd.mm.yyyy** is used.

***JIS** The Japanese Industrial Standard date format **yyyy-mm-dd** is used.

***MDY** The date format **mm/dd/yy** is used.

***DMY** The date format **dd/mm/yy** is used.

***YMD** The date format **yy/mm/dd** is used.

***JUL** The Julian date format **yy/ddd** is used.

Top

Date separator character (DATSEP)

Specifies the separator to be used when accessing date result columns.

Note: This parameter applies only when *JOB, *MDY, *DMY, *YMD, or *JUL is specified for the **Date format (DATFMT)** parameter.

***JOB** The date separator specified for the job at precompile time, when a new interactive SQL session is created, or when Run SQL Statement (RUNSQLSTM) command is run.

Use the Display Job (DSPJOB) command to determine the current date separator value for the job.

'/' A slash is used as the date separator.

'.' A period is used as the date separator.

'-' A dash is used as the date separator.

',' A comma is used as the date separator.

' ' *or* *BLANK
A blank is used as the date separator.

Top

Time format (TIMFMT)

Specifies the format used when accessing time result columns. All output time fields are returned in the format you specify. For input time strings, the value you specify is used to determine whether the time is specified in a valid format.

Note: An input time string that uses the format *USA, *ISO, *EUR, or *JIS is always valid.

If you connect to a relational database that is on a system that is not another System i, the time format must be *USA, *ISO, *EUR, *JIS, or *HMS with a time separator of a colon or period.

***HMS** The **hh:mm:ss** format is used.

***USA** The United States time format **hh:mmxx** is used, where **xx** is AM or PM.

***ISO** The International Organization for Standardization (ISO) time format **hh.mm.ss** is used.

***EUR** The European time format **hh.mm.ss** is used.

***JIS** The Japanese Industrial Standard time format **hh:mm:ss** is used.

Top

Time separator character (TIMSEP)

Specifies the separator used when accessing time result columns.

Note: This parameter applies only when *HMS is specified for the **Time format (TIMFMT)** parameter.

***JOB** The time separator specified for the job at precompile time, when a new interactive SQL session is created, or when RUNSQLSTM is run is used.

Use the Display Job (DSPJOB) command to determine the current time separator value for the job.

':' A colon is used as the time separator.

'.' A period is used as the time separator.

',' A comma is used as the time separator.

' ' *or* *BLANK
A blank is used as the time separator.

Top

Replace (REPLACE)

Specifies if a SQL module is created when there is an existing SQL module of the same name in the same library. The value for this parameter is passed to the CRTCPPMOD command.

***YES** A new SQL module is created, and any existing module of the same name in the specified library is moved to QRPLOBJ.

***NO** A new SQL module is not created if a module of the same name already exists in the specified library.

Top

RDB connect method (RDBCNNMTH)

Specifies the semantics used for CONNECT statements.

***DUW**
CONNECT (Type 2) semantics are used to support distributed unit of work. Consecutive CONNECT statements to additional relational databases do not result in disconnection of previous connections.

***RUW** CONNECT (Type 1) semantics are used to support remote unit of work. Consecutive CONNECT statements result in the previous connection being disconnected before a new connection is established.

Top

Default collection (DFTRDBCOL)

Specifies the name of the schema identifier used for the unqualified names of the tables, views, indexes, SQL packages, aliases, constraints, external programs, node groups, and triggers. This parameter applies only to static SQL statements.

***NONE**
The naming convention specified for the **Precompiler options (OPTION)** parameter is used.

name Specify the name of the schema identifier to be used instead of the naming convention specified for the **Precompiler options (OPTION)** parameter.

Top

Dynamic default collection (DYNDFTCOL)

Specifies whether the default schema name specified for the **Default collection (DFTRDBCOL)** parameter is also used for dynamic statements.

***NO** Do not use the value specified for the **Default collection (DFTRDBCOL)** parameter for unqualified names of tables, views, indexes, SQL packages, aliases, constraints, external programs, node groups, and triggers for dynamic SQL statements. The naming convention specified for the **Precompiler options (OPTION)** parameter is used.

***YES** The schema name specified for the **Default collection (DFTRDBCOL)** parameter will be used for the unqualified names of the tables, views, indexes, SQL packages, aliases, constraints, external programs, node groups, and triggers in dynamic SQL statements.

Top

Package (SQLPKG)

Specifies the qualified name of the SQL package created on the relational database specified for the **Relational database (RDB)** parameter of this command.

Qualifier 1: Package

***OBJ** The name of the SQL package is the same as the object name specified for the **Object (OBJ)** parameter.

name Specify the name of the SQL package. If the remote system is not a System i, no more than 8 characters can be specified.

Qualifier 2: Library

***OBJLIB**
The package is created in the library with the same name as the library specified for the **Object (OBJ)** parameter.

name Specify the name of the library where the package is created.

Top

SQL path (SQLPATH)

Specifies the path to be used to find procedures, functions, and user defined types in static SQL statements.

***NAMING**
The path used depends on the naming convention specified for the **Precompiler options (OPTION)** parameter.

For *SYS naming, the path used is *LIBL, the current library list at runtime.

For *SQL naming, the path used is ″QSYS″, ″QSYS2″, ″userid″, where ″userid″ is the value of the USER special register. If a schema name is specified for the **Default collection (DFTRDBCOL)** parameter, the schema name takes the place of userid.

***LIBL** The path used is the library list at runtime.

name Specify one or more schema names. A maximum of 268 schema names may be specified.

Top

SQL rules (SQLCURRULE)

Specifies the semantics used for SQL statements.

***DB2** The semantics of all SQL statements will default to the rules established for DB2. The following semantics are controlled by this option:

Hexadecimal constants are treated as character data.

***STD** The semantics of all SQL statements will default to the rules established by the ISO and ANSI SQL standards. The following semantics are controlled by this option:

Hexadecimal constants are treated as binary data.

Top

IBM SQL flagging (SAAFLAG)

Specifies the IBM SQL flagging function. This parameter allows you to flag SQL statements to verify whether they conform to IBM SQL syntax.

***NOFLAG**
No checks are made to see whether SQL statements conform to IBM SQL syntax.

***FLAG**
Checks are made to see whether SQL statements conform to IBM SQL syntax.

Top

ANS flagging (FLAGSTD)

Specifies whether non-standard statements are flagged. This parameter allows you to flag SQL statements to verify whether they conform to the Core level of the ISO/IEC 9075-2003 standards.

***NONE**
No checks are made to see whether SQL statements conform to ANSI standards.

***ANS** Checks are made to see whether SQL statements conform to standards.

Top

Print file (PRTFILE)

Specifies the printer device file to be used for the precompiler output listing.

Qualifier 1: Print file

QSYSPRT
The precompiler output file is directed to the IBM-supplied printer file, QSYSPRT. The file QSYSPRT has a record length of 132. If you specify a file whose record length is less than 132, information is lost.

name Specify the name of the printer device file to which the precompiler output is directed.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the printer file is located.

Top

Debugging view (DBGVIEW)

Specifies the type of source debug information to be provided by the SQL precompiler.

***NONE**
The source view will not be generated.

***SOURCE**
The SQL precompiler will provide the source views for the root and if necessary, SQL INCLUDE statements. A view is to be provided which contains the statements generated by the precompiler.

Top

User profile (USRPRF)

Specifies the user profile that is used when the compiled program object and SQL package object is run, including the authority that the program object or SQL package has for each object in static SQL statements. The profile of either the owner or the user is used to control access to objects.

***NAMING**
The user profile is determined by the naming convention. If the naming convention is *SQL, USRPRF(*OWNER) is used. If the naming convention is *SYS, USRPRF(*USER) is used.

***USER**
The profile of the user running the program or SQL package is used.

***OWNER**
The user profiles of both the owner and the user are used when the program or SQL package is run.

Top

Dynamic user profile (DYNUSRPRF)

Specifies the user profile used for dynamic SQL statements.

***USER**
Local dynamic SQL statements are run under the profile of the program's user. Distributed dynamic SQL statements are run under the profile of the application server job.

***OWNER**
Local dynamic SQL statements are run under the profile of the program's owner. Distributed dynamic SQL statements are run under the profile of the SQL package's owner.

Top

Sort sequence (SRTSEQ)

Specifies the sort sequence table to be used for string comparisons in SQL statements.

Note: *HEX must be specified for this parameter on distributed applications where the application server is not on a System i.

Single values

***JOB** The SRTSEQ value for the job is used.

***JOBRUN**
The SRTSEQ value for the job is retrieved when the program is run. For distributed applications, SRTSEQ(*JOBRUN) is valid only when LANGID(*JOBRUN) is also specified.

***LANGIDUNQ**
The unique-weight sort table for the language specified for the **Language id (LANGID)** parameter is used.

***LANGIDSHR**
The shared-weight sort table for the language specified for the LANGID parameter is used.

***HEX** A sort sequence table is not used. The hexadecimal values of the characters are used to determine the sort sequence.

Qualifier 1: Sort sequence

name Specify the name of the sort sequence table to be used with this program.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library to be searched.

Top

Language id (LANGID)

Specifies the language identifier to be used when SRTSEQ(*LANGIDUNQ) or SRTSEQ(*LANGIDSHR) is specified.

***JOB** The LANGID value for the job is retrieved during the precompile.

***JOBRUN**
The LANGID value for the job is retrieved when the program is run. For distributed applications, LANGID(*JOBRUN) is valid only when SRTSEQ(*JOBRUN) is also specified.

language-id
Specify the language identifier to be used by the program.

Top

To source file (TOSRCFILE)

Specifies the source file that is to contain the output source member that has been processed by the SQL precompiler. If the specified source file is not found, it will be created. The output member will have the same name as the name specified for the OBJECT parameter.

Qualifier 1: To source file

***CALC**
The output source file name will be generated based on the margins of the source file. The name will be QSQLTxxxxx, where xxxxx is the width of the source file. If the source file record length is less than or equal to 92, the name will be QSQLTEMP.

name Specify the name of the source file to contain the output source member.

Qualifier 2: Library

QTEMP
The library QTEMP will be used.

***LIBL** The job's library list is searched for the specified file. If the file is not found in any library in the library list, the file will be created in the current library.

***CURLIB**
The current library for the job will be used. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library that is to contain the output source file.

Top

Decimal result options (DECRESULT)

Specifies the maximum precision, maximum scale and minimum divide scale that should be returned for result data types. The specified limit only applies to numeric (zoned) and decimal (packed) data types used in arithmetic expressions and in SQL column functions AVG and SUM.

Element 1: Maximum precision

31 The maximum precision (length) that should be returned for the result data types is 31 digits.

63 The maximum precision (length) that should be returned for the result data types is 63 digits.

Element 2: Maximum scale

31 The maximum scale (number of decimal positions to the right of the decimal point) that should be returned for the result data types is 31 digits.

0-63 Specify the maximum scale (number of decimal positions to the right of the decimal point) that should be returned for the result data types. The value can range from 0 to the maximum precision.

Element 3: Minimum divide scale

0 The minimum divide scale (number of decimal positions to the right of the decimal point) that should be returned for both intermediate and result data types is 0.

0-9 Specify the minimum divide scale (number of decimal positions to the right of the decimal point)

that should be returned for both intermediate and result data types. The value cannot to exceed the maximum scale. If 0 is specified for the maximum scale, minimum divide scale is not used.

Top

Decimal float rounding mode (DECFLTRND)

Specifies the decimal floating point rounding mode used for static SQL statements.

***HALFEVEN**
Round to nearest; if equidistant, round so that the final digit is even. If the discarded digits represent greater than half (0.5) the value of a one in the next left position then the result coefficient should be incremented by 1 (rounded up). If they represent less than half, then the result coefficient is not adjusted (that is, the discarded digits are ignored). Otherwise (they represent exactly half) the result coefficient is unaltered if its rightmost digit is even, or incremented by 1 (rounded up) if its rightmost digit is odd (to make an even digit).

***HALFUP**
Round to nearest; if equidistant, round up. If the discarded digits represent greater than or equal to half (0.5) of the value of a one in the next left position then the result coefficient should be incremented by 1 (rounded up). Otherwise the discarded digits are ignored.

***DOWN**
Round towards 0 (truncation). The discarded digits are ignored.

***CEILING**
Round towards +infinity. If all of the discarded digits are zero or if the sign is negative the result is unchanged other than the removal of discarded digits. Otherwise, the result coefficient should be incremented by 1 (rounded up).

***FLOOR**
Round towards -infinity. If all of the discarded digits are zero or if the sign is positive the result is unchanged other than the removal of discarded digits. Otherwise, the sign is negative and the result coefficient should be incremented by 1.

***HALFDOWN**
Round to nearest; if equidistant, round down. If the discarded digits represent greater than half (0.5) of the value of a one in the next left position then the result coefficient should be incremented by 1 (rounded up). Otherwise (the discarded digits are 0.5 or less) the discarded digits are ignored.

***UP** Round away from 0. Of all of the discarded digits are zero the result is unchanged other than the removal of discarded digits. Otherwise, the result coefficient should be incremented by 1 (rounded up).

Top

Compiler options (COMPILEOPT)

Specifies additional parameters to be used on the compiler command. The COMPILEOPT string is added to the compiler command built by the precompiler. If **INCDIR(** is anywhere in the string, the precompiler will call the compiler using the SRCSTMF parameter. There is no validation of the string. The compiler command will issue an error if any parameter is incorrect. Please refer to the DB2 for i5/OS SQL programming topic collection in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/ for a list of parameters that the precompiler generates for the compiler command. Using any of the keywords that the precompiler passes to the compiler will cause the compiler command to fail because of duplicate parameters.

***NONE**
No additional parameters will be used on the compiler command.

character-value
Specify no more than 5000 characters, enclosed in apostrophes.

Top

Examples

```
CRTSQLCPPI   OBJ(PAYROLL)  OBJTYPE(*MODULE)
             TEXT('Payroll Program')
```

This command runs the SQL precompiler which precompiles the C++ source and stores the changed source in member PAYROLL of source file QSQLTEMP in library QTEMP. The command calls the ILE C++ compiler to create module PAYROLL in the current library by using the source member that is created by the SQL precompiler.

Top

Error messages

***ESCAPE Messages**

SQL9001
SQL precompile failed.

SQL9002
Conflict in TGTRLS parameters for SQL precompile and &7 compile.

SQL9003
&7 Compile at wrong level for SQL source.

SQL9006
DB2 Query Mgr and SQL DevKit not at same install level as the operating system.

Top

Create SQL PL/I Program (CRTSQLPLI)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Create SQL PL/I Program (CRTSQLPLI) command calls the Structured Query Language (SQL) precompiler which precompiles PL/I source containing SQL statements, produces a temporary source member, and then optionally calls the PL/I compiler to compile the program.

If the **Relational database (RDB)** parameter is specified and a program is created, an SQL package will be created at the specified relational database.

Top

Parameters

Keyword	Description	Choices	Notes
PGM	Program	*Qualified object name*	Required, Positional 1
	Qualifier 1: Program	*Name*	
	Qualifier 2: Library	*Name*, ***CURLIB**	
SRCFILE	Source file	*Qualified object name*	Optional, Positional 2
	Qualifier 1: Source file	*Name*, **QPLISRC**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SRCMBR	Source member	*Name*, ***PGM**	Optional, Positional 3
COMMIT	Commitment control	***CHG**, *ALL, *CS, *NONE, *RR, *UR, *RS, *NC	Optional
RDB	Relational database	*Simple name*, ***LOCAL**, *NONE	Optional
TEXT	Text 'description'	*Character value*, ***SRCMBRTXT**, *BLANK	Optional
USER	RDB user	*Name*, ***CURRENT**	Optional
PASSWORD	RDB user password	*Character value*, ***NONE**, ' '	Optional
OPTION	Precompiler options	Values (up to 10 repetitions): ***NOSRC**, *NOSOURCE, *SRC, *SOURCE, *XREF, *NOXREF, *GEN, *NOGEN, *JOB, *SYSVAL, *PERIOD, *COMMA, *SECLVL, *NOSECLVL, *SQL, *SYS, *OPTLOB, *NOOPTLOB, *NOEXTIND, *EXTIND	Optional
TGTRLS	Target release	*Simple name*, ***CURRENT**, *PRV	Optional
INCFILE	INCLUDE file	*Qualified object name*	Optional
	Qualifier 1: INCLUDE file	*Name*, ***SRCFILE**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
ALWCPYDTA	Allow copy of data	***OPTIMIZE**, *YES, *NO	Optional
CLOSQLCSR	Close SQL cursor	***ENDPGM**, *ENDSQL, *ENDJOB	Optional
ALWBLK	Allow blocking	***ALLREAD**, *NONE, *READ	Optional
DLYPRP	Delay PREPARE	***NO**, *YES	Optional
GENLVL	Severity level	0-40, **10**	Optional
MARGINS	Source margins	*Element list*	Optional
	Element 1: Left margin	1-80, ***SRCFILE**	
	Element 2: Right margin	1-80	

Keyword	Description	Choices	Notes
DATFMT	Date format	***JOB**, *USA, *ISO, *EUR, *JIS, *MDY, *DMY, *YMD, *JUL	Optional
DATSEP	Date separator character	***JOB**, '/', '.', ',', '-', ' ', *BLANK	Optional
TIMFMT	Time format	***HMS**, *USA, *ISO, *EUR, *JIS	Optional
TIMSEP	Time separator character	***JOB**, ':', '.', ',', ' ', *BLANK	Optional
REPLACE	Replace	***YES**, *NO	Optional
RDBCNNMTH	RDB connect method	***DUW**, *RUW	Optional
DFTRDBCOL	Default collection	*Name*, ***NONE**	Optional
DYNDFTCOL	Dynamic default collection	***NO**, *YES	Optional
SQLPKG	Package	*Qualified object name*	Optional
	Qualifier 1: Package	*Name*, ***PGM**	
	Qualifier 2: Library	*Name*, ***PGMLIB**	
SQLPATH	SQL path	Single values: ***NAMING**, *LIBL Other values (up to 268 repetitions): *Name*	Optional
SQLCURRULE	SQL rules	***DB2**, *STD	Optional
SAAFLAG	IBM SQL flagging	***NOFLAG**, *FLAG	Optional
FLAGSTD	ANS flagging	***NONE**, *ANS	Optional
PRTFILE	Print file	*Qualified object name*	Optional
	Qualifier 1: Print file	*Name*, **QSYSPRT**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
USRPRF	User profile	***NAMING**, *USER, *OWNER	Optional
DYNUSRPRF	Dynamic user profile	***USER**, *OWNER	Optional
SRTSEQ	Sort sequence	Single values: ***JOB**, *HEX, *JOBRUN, *LANGIDUNQ, *LANGIDSHR Other values: *Qualified object name*	Optional
	Qualifier 1: Sort sequence	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
LANGID	Language id	*Character value*, ***JOB**, *JOBRUN	Optional
TOSRCFILE	To source file	*Qualified object name*	Optional
	Qualifier 1: To source file	*Name*, **QSQLTEMP**	
	Qualifier 2: Library	*Name*, **QTEMP**, *LIBL, *CURLIB	
DECRESULT	Decimal result options	*Element list*	Optional
	Element 1: Maximum precision	**31**, 63	
	Element 2: Maximum scale	0-63, **31**	
	Element 3: Minimum divide scale	0-9, **0**	
DECFLTRND	Decimal float rounding mode	***HALFEVEN**, *HALFUP, *DOWN, *CEILING, *FLOOR, *HALFDOWN, *UP	Optional
COMPILEOPT	Compiler options	*Character value*, ***NONE**	Optional

Top

Program (PGM)

Specifies the program to be created.

This is a required parameter.

Qualifier 1: Program

name Specify the name of the program to be created.

Qualifier 2: Library

***CURLIB**
The current library for the job is used to locate the compiled program. If no current library entry exists in the library list, QGPL is used.

name Specify the name of the library where the compiled program is located.

Top

Source file (SRCFILE)

Specifies the source file that contains the PL/I source statements and SQL statements.

Qualifier 1: Source file

QPLISRC
Source file QPLISRC contains the PL/I source.

name Specify the name of the source file that contains the PL/I source.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library to be searched.

Top

Source member (SRCMBR)

Specifies the source file member that contains the input source. This parameter is used only if the source file specified for the **Source file (SRCFILE)** parameter is a database file.

***PGM** The source file member that has the same name as the program name specified for the **Program (PGM)** parameter contains the input source.

name Specify the name of the source file member that contains the input source.

Top

Commitment control (COMMIT)

Specifies whether SQL statements in the compiled program are run under commitment control. Files referred to in the host language source are not affected by this option. Only SQL tables, SQL views, SQL packages, SQL sequences, SQL aliases, SQL Types, SQL procedures, SQL functions, SQL indexes, SQL schemas, SQL triggers, and SQL views referred to in SQL statements are affected.

***CHG** *or* ***UR**
Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs can be seen.

***CS** Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows updated, deleted, and inserted are locked until the end of the unit of work (transaction). A row that is selected, but not updated, is locked until the next row is selected. Uncommitted changes in other jobs cannot be seen.

***ALL** *or* ***RS**
Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows selected, updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs cannot be seen.

***NONE** *or* ***NC**
Specifies that commitment control is not used. Uncommitted changes in other jobs can be seen. If the SQL DROP SCHEMA statement is included in the program, *NONE or *NC must be used. If a relational database is specified for the **Relational database (RDB)** parameter, and the relational database is on a system that is not on a System i, *NONE or *NC cannot be specified.

***RR** Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows selected, updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs cannot be seen. All tables referred to in SELECT, UPDATE, DELETE, and INSERT statements are locked exclusively until the end of the unit of work (transaction).

Top

Relational database (RDB)

Specifies the name of the relational database where the SQL package is to be created.

***LOCAL**
The program is created as a distributed SQL program. The SQL statements will access the local database. An SQL package object is not created as part of the precompile process. The Create Structured Query Language Package (CRTSQLPKG) command can be used.

***NONE**
An SQL package object is not created. The program object is not a distributed program and the Create Structured Query Language Package (CRTSQLPKG) command cannot be used.

name Specify the name of the relational database where the new SQL package object is to be created. When the name of the local relational database is specified, the program created is still a distributed SQL program. The SQL statements will access the local database.

Top

Text 'description' (TEXT)

Specifies text that briefly describes the program and its function.

***SRCMBRTXT**
The text is taken from the source file member being used to create the program. If the source file is an inline file or a device file, the text is blank.

***BLANK**
No text is specified.

'description'
Specify no more than 50 characters, enclosed in apostrophes.

Top

RDB user (USER)

Specifies the user name sent to the remote system when starting the conversation. This parameter is valid only when **Relational database (RDB)** is specified.

***CURRENT**
The user name associated with the current job is used.

name Specify the user name to be used for the application server job.

Top

RDB user password (PASSWORD)

Specifies the password to be used on the remote system. This parameter is valid only when **Relational database (RDB)** is specified.

***NONE**
No password is sent. A user name cannot be specified for the **RDB user (USER)** parameter if this value is specified.

Note: Specifying a password of a blank is the same as specifying *NONE.

password
Specify the password of the user name specified for the **RDB user (USER)** parameter.

Top

Precompiler options (OPTION)

Specifies whether the following options are used when the PL/I source is precompiled. If an option is specified more than once, or if two options conflict, the last option specified is used.

Source listing options:

***NOSRC** *or* ***NOSOURCE**
The precompiler does not produce a source printout unless errors are detected during precompile or create package.

***SRC** *or* ***SOURCE**
The precompiler produces a source printout.

Cross reference options:

***NOXREF**
The precompiler does not produce a cross-reference of names.

***XREF** The precompiler produces a cross-reference between items declared in your program and the numbers of the statements in your program that refer to these items.

Host language compiler call options:

***GEN** The precompiler calls the PL/I compiler. If a relational database name is specified for the **Relational database (RDB)** parameter, and the compile is successful, an SQL package is also created.

***NOGEN**
The precompiler does not call the PL/I compiler. Neither a program nor an SQL package is created.

Decimal point options:

***PERIOD**
The value used as the decimal point for numeric constants used in SQL statements is a period. This value is also used as the decimal point character when casting a numeric value to character.

***JOB** The representation for the decimal point specified for the job at precompile time is used.

Note: If QDECFMT specifies that the value used as the decimal point is a comma, any numeric constants in lists (such as in the SELECT clause or the VALUES clause) must be separated by a comma followed by a blank. For example, VALUES(1,1, 2,23, 4,1) is equivalent to VALUES(1.1,2.23,4.1) in which the decimal point is a period.

***SYSVAL**
The value used as the decimal point is the QDECFMT system value.

Note: If QDECFMT specifies that the value used as the decimal point is a comma, any numeric constants in lists (such as in the SELECT clause or the VALUES clause) must be separated by a comma followed by a blank. For example, VALUES(1,1, 2,23, 4,1) is equivalent to VALUES(1.1,2.23,4.1) in which the decimal point is a period.

***COMMA**
The value used as the decimal point is a comma.

Note: Any numeric constants in lists (such as in the SELECT clause or the VALUES clause) must be separated by a comma followed by a blank. For example, VALUES(1,1, 2,23, 4,1) is equivalent to VALUES(1.1,2.23,4.1) where the decimal point is a period.

Naming convention options:

***SYS** Specifies that the system naming convention is used (library-name/file-name).

***SQL** Specifies that the SQL naming convention is used (schema-name.table-name).

If a relational database is specified for the **Relational database (RDB)** parameter, and the database is on a system that is not a System i, *SQL must be specified as the naming convention.

Second-level message text options:

***NOSECLVL**
Second-level text descriptions are not added to the listing.

***SECLVL**
Second-level text with replacement data is added for all messages on the listing.

LOB optimization for DRDA options:

***OPTLOB**
The first FETCH for a cursor determines how the cursor will be used for LOBs (Large Objects) on all subsequent FETCHes. This option remains in effect until the cursor is closed.

If the first FETCH uses a LOB locator to access a LOB column, no subsequent FETCH for that cursor can fetch that LOB column into a LOB host variable.

If the first FETCH places the LOB column into a LOB host variable, no subsequent FETCH for that cursor can use a LOB locator for that column.

***NOOPTLOB**
There is no restriction on whether a column is retrieved into a LOB locator or into a LOB host variable. This option can cause performance to degrade.

Extended indicators options:

***NOEXTIND**
Extended indicator support is not enabled.

***EXTIND**
Extended indicator support is enabled.

Top

Target release (TGTRLS)

Specifies the release of the operating system on which you intend to use the object being created.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

***CURRENT**
The object is to be used on the release of the operating system currently running on your system. The object can also be used on a system with any subsequent release of the operating system installed.

***PRV** The object is to be used on the previous release with modification level 0 of the operating system. The object can also be used on a system with any subsequent release of the operating system installed.

target-release
Specify the release in the format VxRxMx. The object can be used on a system with the specified release or with any subsequent release of the operating system installed.

Top

INCLUDE file (INCFILE)

Specifies the source file that contains members to be included in the program with the SQL INCLUDE statement.

Single values

***SRCFILE**
The qualified source file you specify for the **Source file (SRCFILE)** parameter contains the source file members specified on any SQL INCLUDE statements.

Qualifier 1: INCLUDE file

name Specify the name of the source file that contains the source file members specified on any SQL INCLUDE statements.

The record length of the source file you specify here must be no less than the record length of the source file you specify for the **Source file (SRCFILE)** parameter.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the source file is located.

Top

Allow copy of data (ALWCPYDTA)

Specifies whether a copy of the data can be used in a SELECT statement.

***OPTIMIZE**
The system determines whether to use the data retrieved directly from the database or to use a copy of the data. The decision is based on which method provides the best performance. If the **Commitment control (COMMIT)** parameter is not *NONE, the **Allow blocking (ALWBLK)** parameter should be set to *ALLREAD, when possible, for best performance.

***YES** A copy of the data is used only when necessary.

***NO** A copy of the data is not used. If a temporary copy of the data is required to perform the query, an error message is returned.

Top

Close SQL cursor (CLOSQLCSR)

Specifies when SQL cursors are implicitly closed, SQL prepared statements are implicitly discarded, and LOCK TABLE locks released. SQL cursors are explicitly closed by issuing the CLOSE, COMMIT (without HOLD), ROLLBACK (without HOLD), or CONNECT (Type 1) SQL statements.

***ENDPGM**
The SQL cursors are closed and SQL prepared statements are discarded when the program ends. LOCK TABLE locks are released when the first SQL program on the call stack ends.

***ENDSQL**
The SQL cursors remain open between calls and rows can be fetched without running another SQL OPEN statement. One of the programs higher on the call stack must have run at least one SQL statement. The SQL cursors are closed, SQL prepared statements are discarded, and LOCK TABLE locks are released when the first SQL program on the call stack ends. If you specify *ENDSQL for a program that is the first SQL program called (the first SQL program on the call stack), the program is treated as if *ENDPGM was specified.

***ENDJOB**
SQL cursors remain open between calls and can be fetched without running another SQL OPEN statement. The programs higher on the call stack do not need to have run SQL statements. SQL cursors are left open, SQL prepared statements are preserved, and LOCK TABLE locks are held when the first SQL program on the call stack ends. SQL cursors are closed, SQL prepared statements are discarded, and LOCK TABLE locks are released when the job ends.

Top

Allow blocking (ALWBLK)

Specifies whether the database manager can use record blocking and the extent to which blocking can be used for read-only cursors.

***ALLREAD**
Rows are blocked for read-only cursors. All cursors in a program that are not explicitly able to be changed are opened for read-only processing even though there may be EXECUTE or EXECUTE IMMEDIATE statements in the program.

Specifying *ALLREAD:

- Allows record blocking for all read-only cursors.
- Can improve the performance of almost all read-only cursors in programs, but limits queries in the following ways:
  - The Rollback (ROLLBACK) command, a ROLLBACK statement in host languages, or the ROLLBACK HOLD SQL statement does not reposition a read-only cursor when *ALLREAD is specified.
  - Dynamic running of a positioned UPDATE or DELETE statement (for example, using EXECUTE IMMEDIATE), can not be used to update a row in a cursor unless the DECLARE statement for the cursor includes the FOR UPDATE clause.

***NONE**
Rows are not blocked for retrieval of data for cursors.

Specifying *NONE:

- Guarantees that the data retrieved is current.
- May reduce the amount of time required to retrieve the first row of data for a query.
- Stops the database manager from retrieving a block of data rows that is not used by the program when only the first few rows of a query are retrieved before the query is closed.
- Can degrade the overall performance of a query that retrieves a large number of rows.

***READ**
Records are blocked for read-only retrieval of data for cursors when:

- *NONE is specified for the **Commitment control (COMMIT)** parameter, which indicates that commitment control is not used.
- The cursor is declared with a FOR READ ONLY clause or there are no dynamic statements that could run a positioned UPDATE or DELETE statement for the cursor.

Top

Delay PREPARE (DLYPRP)

Specifies whether the dynamic statement validation for a PREPARE statement is delayed until an OPEN, EXECUTE, or DESCRIBE statement is run. Delaying validation improves performance by eliminating redundant validation.

***NO** Dynamic statement validation is not delayed. When the dynamic statement is prepared, the access plan is validated. When the dynamic statement is used in an OPEN or EXECUTE statement, the access plan is revalidated. Because the authority or the existence of objects referred to by the dynamic statement may change, you must still check the SQLCODE or SQLSTATE after issuing the OPEN or EXECUTE statement to ensure that the dynamic statement is still valid.

***YES** Dynamic statement validation is delayed until the dynamic statement is used in an OPEN, EXECUTE, or DESCRIBE SQL statement. When the dynamic statement is used, the validation is completed and an access plan is built. If you specify *YES for this parameter for precompiled programs, you should check the SQLCODE and SQLSTATE after running an OPEN, EXECUTE, or DESCRIBE statement to ensure that the dynamic statement is valid.

Note: If you specify *YES, performance is not improved if the INTO clause is used on the PREPARE statement or if a DESCRIBE statement uses the dynamic statement before an OPEN is issued for the statement.

Top

Severity level (GENLVL)

Specifies whether the compiler is called, depending on the severity of messages generated as a result of errors found by the SQL precompiler. If precompiler errors are generated that have a message severity level greater than the value specified for this parameter, the compiler is not called.

If the **Relational database (RDB)** parameter is specified and the severity of the messages generated as a result of package creation is greater than the severity level specified for this parameter, the SQL package is not created.

10 Do not call the compiler if SQL precompiler messages with a message severity greater than 10 are generated.

0-40 Specify the maximum SQL precompiler message severity level to be used to control whether the compiler is called.

Top

Source margins (MARGINS)

Specifies the part of the precompiler input record that contains source text.

Element 1: Left margin

***SRCFILE**
The margin values of the file member you specify for the **Source member (SRCMBR)** parameter are used. If the member is of source type SQLPLI, the margin values are the values specified on the SEU services display. If the member is of a different source type, the margin values are the default values of 2 and 72.

1-80 Specify the beginning position to be used for each input record.

Element 2: Right margin

1-80 Specify the ending position to be used for each input record.

Top

Date format (DATFMT)

Specifies the format used when accessing date result columns. All output date fields are returned in the format you specify. For input date strings, the value you specify is used to determine whether the date is a valid format.

Note: An input date string that uses the format *USA, *ISO, *EUR, or *JIS is always valid.

If you connect to a relational database that is on a system that is not a System i, the format must be *USA, *ISO, *EUR, or *JIS.

***JOB** The format specified for the job at precompile time or when a new interactive SQL session is created is used.

Use the Display Job (DSPJOB) command to determine the current date format for the job.

***USA** The United States date format **mm/dd/yyyy** is used.

***ISO** The International Organization for Standardization (ISO) date format **yyyy-mm-dd** is used.

***EUR** The European date format **dd.mm.yyyy** is used.

***JIS** The Japanese Industrial Standard date format **yyyy-mm-dd** is used.

***MDY** The date format **mm/dd/yy** is used.

***DMY** The date format **dd/mm/yy** is used.

***YMD** The date format **yy/mm/dd** is used.

***JUL** The Julian date format **yy/ddd** is used.

Top

Date separator character (DATSEP)

Specifies the separator to be used when accessing date result columns.

Note: This parameter applies only when *JOB, *MDY, *DMY, *YMD, or *JUL is specified for the **Date format (DATFMT)** parameter.

***JOB** The date separator specified for the job at precompile time, when a new interactive SQL session is created, or when Run SQL Statement (RUNSQLSTM) command is run.

Use the Display Job (DSPJOB) command to determine the current date separator value for the job.

'/' A slash is used as the date separator.

'.' A period is used as the date separator.

'-' A dash is used as the date separator.

',' A comma is used as the date separator.

' ' *or* *BLANK
A blank is used as the date separator.

Top

Time format (TIMFMT)

Specifies the format used when accessing time result columns. All output time fields are returned in the format you specify. For input time strings, the value you specify is used to determine whether the time is specified in a valid format.

Note: An input time string that uses the format *USA, *ISO, *EUR, or *JIS is always valid.

If you connect to a relational database that is on a system that is not another System i, the time format must be *USA, *ISO, *EUR, *JIS, or *HMS with a time separator of a colon or period.

***HMS** The **hh:mm:ss** format is used.

***USA** The United States time format **hh:mmxx** is used, where **xx** is AM or PM.

***ISO** The International Organization for Standardization (ISO) time format **hh.mm.ss** is used.

***EUR** The European time format **hh.mm.ss** is used.

***JIS** The Japanese Industrial Standard time format **hh:mm:ss** is used.

Top

Time separator character (TIMSEP)

Specifies the separator used when accessing time result columns.

Note: This parameter applies only when *HMS is specified for the **Time format (TIMFMT)** parameter.

***JOB** The time separator specified for the job at precompile time, when a new interactive SQL session is created, or when RUNSQLSTM is run is used.

Use the Display Job (DSPJOB) command to determine the current time separator value for the job.

':' A colon is used as the time separator.

'.' A period is used as the time separator.

',' A comma is used as the time separator.

' ' *or* *BLANK
A blank is used as the time separator.

Top

Replace (REPLACE)

Specifies whether a SQL program or SQL package is created when there is an existing SQL program or SQL package of the same name in the same library. The value is passed to the CRTxxxPGM command (where xxx is the language of the program being created) and the Create SQL Package (CRTSQLPKG) command if the **Relational database (RDB)** parameter is specified.

***YES** An SQL program or SQL package is created and any existing SQL program or SQL package of the same name in the specified library is moved to QRPLOBJ. The authorities for the existing SQL package are kept for the new SQL package.

***NO** An SQL program or package is not created if an SQL program or package of the same name already exists in the specified library.

Top

RDB connect method (RDBCNNMTH)

Specifies the semantics used for CONNECT statements.

***DUW**
CONNECT (Type 2) semantics are used to support distributed unit of work. Consecutive CONNECT statements to additional relational databases do not result in disconnection of previous connections.

***RUW** CONNECT (Type 1) semantics are used to support remote unit of work. Consecutive CONNECT statements result in the previous connection being disconnected before a new connection is established.

Top

Default collection (DFTRDBCOL)

Specifies the name of the schema identifier used for the unqualified names of the tables, views, indexes, SQL packages, aliases, constraints, external programs, node groups, and triggers. This parameter applies only to static SQL statements.

***NONE**
The naming convention specified for the **Precompiler options (OPTION)** parameter is used.

name Specify the name of the schema identifier to be used instead of the naming convention specified for the **Precompiler options (OPTION)** parameter.

Top

Dynamic default collection (DYNDFTCOL)

Specifies whether the default schema name specified for the **Default collection (DFTRDBCOL)** parameter is also used for dynamic statements.

***NO** Do not use the value specified for the **Default collection (DFTRDBCOL)** parameter for unqualified names of tables, views, indexes, SQL packages, aliases, constraints, external programs, node groups, and triggers for dynamic SQL statements. The naming convention specified for the **Precompiler options (OPTION)** parameter is used.

***YES** The schema name specified for the **Default collection (DFTRDBCOL)** parameter will be used for the unqualified names of the tables, views, indexes, SQL packages, aliases, constraints, external programs, node groups, and triggers in dynamic SQL statements.

Top

Package (SQLPKG)

Specifies the name and library of the SQL package to be created on the remote relational database specified for the **Relational database (RDB)** parameter of this command.

Qualifier 1: Package

***PGM** The name of the SQL package is the same as the program name.

name Specify the name of the SQL package. If the remote system is not a System i, a maximum of 8 characters can be specified.

Qualifier 2: Library

***PGMLIB**
The SQL package is placed in the schema that has the same name as the library containing the program.

name Specify the name of the schema where the SQL package is to be placed. If the remote system is not a System i, a maximum of 8 characters can be specified.

Top

SQL path (SQLPATH)

Specifies the path to be used to find procedures, functions, and user defined types in static SQL statements.

***NAMING**
The path used depends on the naming convention specified for the **Precompiler options (OPTION)** parameter.

For *SYS naming, the path used is *LIBL, the current library list at runtime.

For *SQL naming, the path used is ″QSYS″, ″QSYS2″, ″userid″, where ″userid″ is the value of the USER special register. If a schema name is specified for the **Default collection (DFTRDBCOL)** parameter, the schema name takes the place of userid.

***LIBL** The path used is the library list at runtime.

name Specify one or more schema names. A maximum of 268 schema names may be specified.

Top

SQL rules (SQLCURRULE)

Specifies the semantics used for SQL statements.

***DB2** The semantics of all SQL statements will default to the rules established for DB2. The following semantics are controlled by this option:

Hexadecimal constants are treated as character data.

***STD** The semantics of all SQL statements will default to the rules established by the ISO and ANSI SQL standards. The following semantics are controlled by this option:

Hexadecimal constants are treated as binary data.

Top

IBM SQL flagging (SAAFLAG)

Specifies the IBM SQL flagging function. This parameter allows you to flag SQL statements to verify whether they conform to IBM SQL syntax.

***NOFLAG**
No checks are made to see whether SQL statements conform to IBM SQL syntax.

***FLAG**
Checks are made to see whether SQL statements conform to IBM SQL syntax.

Top

ANS flagging (FLAGSTD)

Specifies whether non-standard statements are flagged. This parameter allows you to flag SQL statements to verify whether they conform to the Core level of the ISO/IEC 9075-2003 standards.

***NONE**
No checks are made to see whether SQL statements conform to ANSI standards.

***ANS** Checks are made to see whether SQL statements conform to standards.

Top

Print file (PRTFILE)

Specifies the printer device file to be used for the precompiler output listing.

Qualifier 1: Print file

QSYSPRT
The precompiler output file is directed to the IBM-supplied printer file, QSYSPRT. The file QSYSPRT has a record length of 132. If you specify a file whose record length is less than 132, information is lost.

name Specify the name of the printer device file to which the precompiler output is directed.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the printer file is located.

Top

User profile (USRPRF)

Specifies the user profile that is used when the compiled program object and SQL package object is run, including the authority that the program object or SQL package has for each object in static SQL statements. The profile of either the owner or the user is used to control access to objects.

***NAMING**
The user profile is determined by the naming convention. If the naming convention is *SQL, USRPRF(*OWNER) is used. If the naming convention is *SYS, USRPRF(*USER) is used.

***USER**
The profile of the user running the program or SQL package is used.

***OWNER**
The user profiles of both the owner and the user are used when the program or SQL package is run.

Top

Dynamic user profile (DYNUSRPRF)

Specifies the user profile used for dynamic SQL statements.

***USER**
Local dynamic SQL statements are run under the profile of the program's user. Distributed dynamic SQL statements are run under the profile of the application server job.

***OWNER**
Local dynamic SQL statements are run under the profile of the program's owner. Distributed dynamic SQL statements are run under the profile of the SQL package's owner.

Top

Sort sequence (SRTSEQ)

Specifies the sort sequence table to be used for string comparisons in SQL statements.

Note: *HEX must be specified for this parameter on distributed applications where the application server is not on a System i.

Single values

***JOB** The SRTSEQ value for the job is used.

***JOBRUN**
The SRTSEQ value for the job is retrieved when the program is run. For distributed applications, SRTSEQ(*JOBRUN) is valid only when LANGID(*JOBRUN) is also specified.

***LANGIDUNQ**
The unique-weight sort table for the language specified for the **Language id (LANGID)** parameter is used.

***LANGIDSHR**
The shared-weight sort table for the language specified for the LANGID parameter is used.

***HEX** A sort sequence table is not used. The hexadecimal values of the characters are used to determine the sort sequence.

Qualifier 1: Sort sequence

name Specify the name of the sort sequence table to be used with this program.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library to be searched.

Top

Language id (LANGID)

Specifies the language identifier to be used when SRTSEQ(*LANGIDUNQ) or SRTSEQ(*LANGIDSHR) is specified.

***JOB** The LANGID value for the job is retrieved during the precompile.

***JOBRUN**
The LANGID value for the job is retrieved when the program is run. For distributed applications, LANGID(*JOBRUN) is valid only when SRTSEQ(*JOBRUN) is also specified.

language-id
Specify the language identifier to be used by the program.

Top

To source file (TOSRCFILE)

Specifies the source file that is to contain the output source member that has been processed by the SQL precompiler. If the specified source file is not found, it will be created. The output member will have the same name as the name specified for the **Object (OBJ)** or **Program (PGM)** parameter.

Qualifier 1: To source file

QSQLTEMP
The source file QSQLTEMP will be used.

name Specify the name of the source file to contain the output source member.

Qualifier 2: Library

QTEMP
The library QTEMP will be used.

***LIBL** The job's library list is searched for the specified file. If the file is not found in any library in the library list, the file will be created in the current library.

***CURLIB**
The current library for the job will be used. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library that is to contain the output source file.

Top

Decimal result options (DECRESULT)

Specifies the maximum precision, maximum scale and minimum divide scale that should be returned for result data types. The specified limit only applies to numeric (zoned) and decimal (packed) data types used in arithmetic expressions and in SQL column functions AVG and SUM.

Element 1: Maximum precision

31 The maximum precision (length) that should be returned for the result data types is 31 digits.

63 The maximum precision (length) that should be returned for the result data types is 63 digits.

Element 2: Maximum scale

31 The maximum scale (number of decimal positions to the right of the decimal point) that should be returned for the result data types is 31 digits.

0-63 Specify the maximum scale (number of decimal positions to the right of the decimal point) that should be returned for the result data types. The value can range from 0 to the maximum precision.

Element 3: Minimum divide scale

0 The minimum divide scale (number of decimal positions to the right of the decimal point) that should be returned for both intermediate and result data types is 0.

0-9 Specify the minimum divide scale (number of decimal positions to the right of the decimal point)

that should be returned for both intermediate and result data types. The value cannot to exceed the maximum scale. If 0 is specified for the maximum scale, minimum divide scale is not used.

Top

Decimal float rounding mode (DECFLTRND)

Specifies the decimal floating point rounding mode used for static SQL statements.

***HALFEVEN**
Round to nearest; if equidistant, round so that the final digit is even. If the discarded digits represent greater than half (0.5) the value of a one in the next left position then the result coefficient should be incremented by 1 (rounded up). If they represent less than half, then the result coefficient is not adjusted (that is, the discarded digits are ignored). Otherwise (they represent exactly half) the result coefficient is unaltered if its rightmost digit is even, or incremented by 1 (rounded up) if its rightmost digit is odd (to make an even digit).

***HALFUP**
Round to nearest; if equidistant, round up. If the discarded digits represent greater than or equal to half (0.5) of the value of a one in the next left position then the result coefficient should be incremented by 1 (rounded up). Otherwise the discarded digits are ignored.

***DOWN**
Round towards 0 (truncation). The discarded digits are ignored.

***CEILING**
Round towards +infinity. If all of the discarded digits are zero or if the sign is negative the result is unchanged other than the removal of discarded digits. Otherwise, the result coefficient should be incremented by 1 (rounded up).

***FLOOR**
Round towards -infinity. If all of the discarded digits are zero or if the sign is positive the result is unchanged other than the removal of discarded digits. Otherwise, the sign is negative and the result coefficient should be incremented by 1.

***HALFDOWN**
Round to nearest; if equidistant, round down. If the discarded digits represent greater than half (0.5) of the value of a one in the next left position then the result coefficient should be incremented by 1 (rounded up). Otherwise (the discarded digits are 0.5 or less) the discarded digits are ignored.

***UP** Round away from 0. Of all of the discarded digits are zero the result is unchanged other than the removal of discarded digits. Otherwise, the result coefficient should be incremented by 1 (rounded up).

Top

Compiler options (COMPILEOPT)

Specifies additional parameters to be used on the compiler command. The COMPILEOPT string is added to the compiler command built by the precompiler. There is no validation of the string. The compiler command will issue an error if any parameter is incorrect. Please refer to the DB2 for i5/OS SQL programming topic collection in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/ for a list of parameters that the precompiler generates for the compiler command. Using any of the keywords that the precompiler passes to the compiler will cause the compiler command to fail because of duplicate parameters.

***NONE**
No additional parameters will be used on the compiler command.

character-value
Specify no more than 5000 characters, enclosed in apostrophes.

Top

Examples

```
CRTSQLPLI   PGM(PAYROLL)  TEXT('Payroll Program')
```

This command runs the SQL precompiler, which precompiles the PL/I source and stores the changed source in member PAYROLL of source file QSQLTEMP in library QTEMP. The PL/I compiler is called to create program PAYROLL in the current library using the source member created by the SQL precompiler.

Top

Error messages

***ESCAPE Messages**

SQL9001
SQL precompile failed.

SQL9002
Conflict in TGTRLS parameters for SQL precompile and &7 compile.

SQL9003
&7 Compile at wrong level for SQL source.

SQL9004
Create of SQL package failed.

SQL9006
DB2 Query Mgr and SQL DevKit not at same install level as the operating system.

Top

Create SQL RPG Program (CRTSQLRPG)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Create SQL RPG Program (CRTSQLRPG) command calls the Structured Query Language (SQL) precompiler which precompiles RPG source containing SQL statements, produces a temporary source member, and then optionally calls the RPG compiler to compile the program.

If the **Relational database (RDB)** parameter is specified and a program is created, an SQL package will be created at the specified relational database.

Top

Parameters

Keyword	Description	Choices	Notes
PGM	Program	*Qualified object name*	Required, Positional 1
	Qualifier 1: Program	*Name*	
	Qualifier 2: Library	*Name*, ***CURLIB**	
SRCFILE	Source file	*Qualified object name*	Optional, Positional 2
	Qualifier 1: Source file	*Name*, **QRPGSRC**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SRCMBR	Source member	*Name*, ***PGM**	Optional, Positional 3
COMMIT	Commitment control	***CHG**, *ALL, *CS, *NONE, *RR, *UR, *RS, *NC	Optional
RDB	Relational database	*Simple name*, ***LOCAL**, *NONE	Optional
TEXT	Text 'description'	*Character value*, ***SRCMBRTXT**, *BLANK	Optional
USER	RDB user	*Name*, ***CURRENT**	Optional
PASSWORD	RDB user password	*Character value*, ***NONE**, ' '	Optional
OPTION	Precompiler options	Values (up to 14 repetitions): ***NOSRC**, *NOSOURCE, *SRC, *SOURCE, *XREF, *NOXREF, *GEN, *NOGEN, *COMMA, *PERIOD, *JOB, *SYSVAL, *SECLVL, *NOSECLVL, *SEQSRC, *NOSEQSRC, *LSTDBG, *NOLSTDBG, *SQL, *SYS, *NOEXTIND, *EXTIND	Optional
TGTRLS	Target release	*Simple name*, ***CURRENT**, *PRV	Optional
INCFILE	INCLUDE file	*Qualified object name*	Optional
	Qualifier 1: INCLUDE file	*Name*, ***SRCFILE**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
ALWCPYDTA	Allow copy of data	***OPTIMIZE**, *YES, *NO	Optional
CLOSQLCSR	Close SQL cursor	***ENDPGM**, *ENDSQL, *ENDJOB	Optional
ALWBLK	Allow blocking	***ALLREAD**, *NONE, *READ	Optional
DLYPRP	Delay PREPARE	***NO**, *YES	Optional
GENLVL	Severity level	0-40, **10**	Optional
DATFMT	Date format	***JOB**, *USA, *ISO, *EUR, *JIS, *MDY, *DMY, *YMD, *JUL	Optional
DATSEP	Date separator character	***JOB**, '/', '.', ',', '-', ' ', *BLANK	Optional
TIMFMT	Time format	***HMS**, *USA, *ISO, *EUR, *JIS	Optional

Keyword	Description	Choices	Notes
TIMSEP	Time separator character	***JOB**, ':', '.', ',', ' ', *BLANK	Optional
REPLACE	Replace	***YES**, *NO	Optional
RDBCNNMTH	RDB connect method	***DUW**, *RUW	Optional
DFTRDBCOL	Default collection	*Name*, ***NONE**	Optional
DYNDFTCOL	Dynamic default collection	***NO**, *YES	Optional
SQLPKG	Package	*Qualified object name*	Optional
	Qualifier 1: Package	*Name*, ***PGM**	
	Qualifier 2: Library	*Name*, ***PGMLIB**	
SQLPATH	SQL path	Single values: ***NAMING**, *LIBL Other values (up to 268 repetitions): *Name*	Optional
SQLCURRULE	SQL rules	***DB2**, *STD	Optional
SAAFLAG	IBM SQL flagging	***NOFLAG**, *FLAG	Optional
FLAGSTD	ANS flagging	***NONE**, *ANS	Optional
PRTFILE	Print file	*Qualified object name*	Optional
	Qualifier 1: Print file	*Name*, **QSYSPRT**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
USRPRF	User profile	***NAMING**, *USER, *OWNER	Optional
DYNUSRPRF	Dynamic user profile	***USER**, *OWNER	Optional
SRTSEQ	Sort sequence	Single values: ***JOB**, *HEX, *JOBRUN, *LANGIDUNQ, *LANGIDSHR Other values: *Qualified object name*	Optional
	Qualifier 1: Sort sequence	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
LANGID	Language id	*Character value*, ***JOB**, *JOBRUN	Optional
TOSRCFILE	To source file	*Qualified object name*	Optional
	Qualifier 1: To source file	*Name*, **QSQLTEMP**	
	Qualifier 2: Library	*Name*, **QTEMP**, *LIBL, *CURLIB	
DECRESULT	Decimal result options	*Element list*	Optional
	Element 1: Maximum precision	**31**, 63	
	Element 2: Maximum scale	0-63, **31**	
	Element 3: Minimum divide scale	0-9, **0**	
DECFLTRND	Decimal float rounding mode	***HALFEVEN**, *HALFUP, *DOWN, *CEILING, *FLOOR, *HALFDOWN, *UP	Optional
COMPILEOPT	Compiler options	*Character value*, ***NONE**	Optional

Top

Program (PGM)

Specifies the program to be created.

This is a required parameter.

Qualifier 1: Program

name Specify the name of the program to be created.

Qualifier 2: Library

***CURLIB**
The current library for the job is used to locate the compiled program. If no current library entry exists in the library list, QGPL is used.

name Specify the name of the library where the compiled program is located.

Top

Source file (SRCFILE)

Specifies the source file that contains the RPG source statements and SQL statements.

Qualifier 1: Source file

QRPGSRC
Source file QRPGSRC contains the RPG source.

name Specify the name of the source file that contains the RPG source.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library to be searched.

Top

Source member (SRCMBR)

Specifies the source file member that contains the input source. This parameter is used only if the source file specified for the **Source file (SRCFILE)** parameter is a database file.

***PGM** The source file member that has the same name as the program name specified for the **Program (PGM)** parameter contains the input source.

name Specify the name of the source file member that contains the input source.

Top

Commitment control (COMMIT)

Specifies whether SQL statements in the compiled program are run under commitment control. Files referred to in the host language source are not affected by this option. Only SQL tables, SQL views, SQL packages, SQL sequences, SQL aliases, SQL Types, SQL procedures, SQL functions, SQL indexes, SQL schemas, SQL triggers, and SQL views referred to in SQL statements are affected.

***CHG** *or* ***UR**
Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs can be seen.

***CS** Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows updated, deleted, and inserted are

locked until the end of the unit of work (transaction). A row that is selected, but not updated, is locked until the next row is selected. Uncommitted changes in other jobs cannot be seen.

ALL** ***or ***RS**
Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows selected, updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs cannot be seen.

NONE** ***or ***NC**
Specifies that commitment control is not used. Uncommitted changes in other jobs can be seen. If the SQL DROP SCHEMA statement is included in the program, *NONE or *NC must be used. If a relational database is specified for the **Relational database (RDB)** parameter, and the relational database is on a system that is not on a System i, *NONE or *NC cannot be specified.

***RR** Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows selected, updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs cannot be seen. All tables referred to in SELECT, UPDATE, DELETE, and INSERT statements are locked exclusively until the end of the unit of work (transaction).

Top

Relational database (RDB)

Specifies the name of the relational database where the SQL package is to be created.

***LOCAL**
The program is created as a distributed SQL program. The SQL statements will access the local database. An SQL package object is not created as part of the precompile process. The Create Structured Query Language Package (CRTSQLPKG) command can be used.

***NONE**
An SQL package object is not created. The program object is not a distributed program and the Create Structured Query Language Package (CRTSQLPKG) command cannot be used.

name Specify the name of the relational database where the new SQL package object is to be created. When the name of the local relational database is specified, the program created is still a distributed SQL program. The SQL statements will access the local database.

Top

Text 'description' (TEXT)

Specifies text that briefly describes the program and its function.

***SRCMBRTXT**
The text is taken from the source file member being used to create the program. If the source file is an inline file or a device file, the text is blank.

***BLANK**
No text is specified.

'description'
Specify no more than 50 characters, enclosed in apostrophes.

Top

RDB user (USER)

Specifies the user name sent to the remote system when starting the conversation. This parameter is valid only when **Relational database (RDB)** is specified.

***CURRENT**
The user name associated with the current job is used.

name Specify the user name to be used for the application server job.

Top

RDB user password (PASSWORD)

Specifies the password to be used on the remote system. This parameter is valid only when **Relational database (RDB)** is specified.

***NONE**
No password is sent. A user name cannot be specified for the **RDB user (USER)** parameter if this value is specified.

Note: Specifying a password of a blank is the same as specifying *NONE.

password
Specify the password of the user name specified for the **RDB user (USER)** parameter.

Top

Precompiler options (OPTION)

Specifies whether the following options are used when the RPG source is precompiled. If an option is specified more than once, or if two options conflict, the last option specified is used.

Source listing options:

***NOSRC** *or* ***NOSOURCE**
The precompiler does not produce a source printout unless errors are detected during precompile or create package.

***SRC** *or* ***SOURCE**
The precompiler produces a source printout.

Cross reference options:

***NOXREF**
The precompiler does not produce a cross-reference of names.

***XREF** The precompiler produces a cross-reference between items declared in your program and the numbers of the statements in your program that refer to these items.

Host language compiler call options:

***GEN** The precompiler calls the RPG compiler. If a relational database name is specified for the **Relational database (RDB)** parameter, and the compile is successful, an SQL package is also created.

***NOGEN**
The precompiler does not call the RPG compiler. Neither a program nor an SQL package is created.

Decimal point options:

***JOB** The representation for the decimal point specified for the job at precompile time is used.

Note: If QDECFMT specifies that the value used as the decimal point is a comma, any numeric constants in lists (such as in the SELECT clause and VALUES clause) must be separated by a comma followed by a blank. For example, VALUES(1,1, 2,23, 4,1) is equivalent to VALUES(1.1,2.23,4.1) where the decimal point is the period.

***SYSVAL**
The value used as the decimal point in numeric constants is from the QDECFMT system value. This value is also used as the decimal point character when casting a numeric value to character.

Note: If QDECFMT specifies that the value used as the decimal point is a comma; any numeric constants in lists (such as in the SELECT clause and VALUES clause) must be separated by a comma followed by a blank. For example, VALUES(1,1, 2,23, 4,1) is equivalent to VALUES(1.1,2.23,4.1) where the decimal point is the period.

***PERIOD**
The value used as the decimal point for numeric constants used in SQL statements is a period. This value is also used as the decimal point character when casting a numeric value to character.

***COMMA**
The value used as the decimal point in numeric constants is a comma. Any numeric constants in lists (such as in the SELECT clause and VALUES clause) must be separated by a comma followed by a blank. For example, VALUES(1,1, 2,23, 4,1) is equivalent to VALUES(1.1,2.23,4.1) where the decimal point is the period.

Naming convention options:

***SYS** Specifies that the system naming convention is used (library-name/file-name).

***SQL** Specifies that the SQL naming convention is used (schema-name.table-name).

If a relational database is specified for the **Relational database (RDB)** parameter, and the database is on a system that is not a System i, *SQL must be specified as the naming convention.

Second-level message text options:

***NOSECLVL**
Second-level text descriptions are not added to the listing.

***SECLVL**
Second-level text with replacement data is added for all messages on the listing.

Debug listing view options:

***NOLSTDBG**
Error and debug information is not generated.

***LSTDBG**
The SQL precompiler generates a listing view and error and debug information required for this view.

Note: You can only use *LSTDBG if you are using the CODE product to compile your program.

Sequence source options:

***NOSEQSRC**
The source file member created into QSQLTEMP1 or QSQLTEMP has the same sequence numbers as the original source read by the precompiler.

***SEQSRC**
The source file member created into QSQLTEMP1 or QSQLTEMP contains sequence numbers starting at 000001 and incremented by 000001.

Extended indicators options:

***NOEXTIND**
Extended indicator support is not enabled.

***EXTIND**
Extended indicator support is enabled.

Top

Target release (TGTRLS)

Specifies the release of the operating system on which you intend to use the object being created.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

***CURRENT**
The object is to be used on the release of the operating system currently running on your system. The object can also be used on a system with any subsequent release of the operating system installed.

***PRV** The object is to be used on the previous release with modification level 0 of the operating system. The object can also be used on a system with any subsequent release of the operating system installed.

target-release
Specify the release in the format VxRxMx. The object can be used on a system with the specified release or with any subsequent release of the operating system installed.

Top

INCLUDE file (INCFILE)

Specifies the source file that contains members to be included in the program with the SQL INCLUDE statement.

Single values

***SRCFILE**
The qualified source file you specify for the **Source file (SRCFILE)** parameter contains the source file members specified on any SQL INCLUDE statements.

Qualifier 1: INCLUDE file

name Specify the name of the source file that contains the source file members specified on any SQL INCLUDE statements.

The record length of the source file you specify here must be no less than the record length of the source file you specify for the **Source file (SRCFILE)** parameter.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the source file is located.

Top

Allow copy of data (ALWCPYDTA)

Specifies whether a copy of the data can be used in a SELECT statement.

***OPTIMIZE**
The system determines whether to use the data retrieved directly from the database or to use a copy of the data. The decision is based on which method provides the best performance. If the **Commitment control (COMMIT)** parameter is not *NONE, the **Allow blocking (ALWBLK)** parameter should be set to *ALLREAD, when possible, for best performance.

***YES** A copy of the data is used only when necessary.

***NO** A copy of the data is not used. If a temporary copy of the data is required to perform the query, an error message is returned.

Top

Close SQL cursor (CLOSQLCSR)

Specifies when SQL cursors are implicitly closed, SQL prepared statements are implicitly discarded, and LOCK TABLE locks released. SQL cursors are explicitly closed by issuing the CLOSE, COMMIT (without HOLD), ROLLBACK (without HOLD), or CONNECT (Type 1) SQL statements.

***ENDPGM**
The SQL cursors are closed and SQL prepared statements are discarded when the program ends. LOCK TABLE locks are released when the first SQL program on the call stack ends.

***ENDSQL**
The SQL cursors remain open between calls and rows can be fetched without running another SQL OPEN statement. One of the programs higher on the call stack must have run at least one SQL statement. The SQL cursors are closed, SQL prepared statements are discarded, and LOCK TABLE locks are released when the first SQL program on the call stack ends. If you specify *ENDSQL for a program that is the first SQL program called (the first SQL program on the call stack), the program is treated as if *ENDPGM was specified.

***ENDJOB**
SQL cursors remain open between calls and can be fetched without running another SQL OPEN statement. The programs higher on the call stack do not need to have run SQL statements. SQL cursors are left open, SQL prepared statements are preserved, and LOCK TABLE locks are held when the first SQL program on the call stack ends. SQL cursors are closed, SQL prepared statements are discarded, and LOCK TABLE locks are released when the job ends.

Top

Allow blocking (ALWBLK)

Specifies whether the database manager can use record blocking and the extent to which blocking can be used for read-only cursors.

***ALLREAD**
Rows are blocked for read-only cursors. All cursors in a program that are not explicitly able to be changed are opened for read-only processing even though there may be EXECUTE or EXECUTE IMMEDIATE statements in the program.

Specifying *ALLREAD:

- Allows record blocking for all read-only cursors.
- Can improve the performance of almost all read-only cursors in programs, but limits queries in the following ways:
  - The Rollback (ROLLBACK) command, a ROLLBACK statement in host languages, or the ROLLBACK HOLD SQL statement does not reposition a read-only cursor when *ALLREAD is specified.
  - Dynamic running of a positioned UPDATE or DELETE statement (for example, using EXECUTE IMMEDIATE), can not be used to update a row in a cursor unless the DECLARE statement for the cursor includes the FOR UPDATE clause.

***NONE**
Rows are not blocked for retrieval of data for cursors.

Specifying *NONE:

- Guarantees that the data retrieved is current.
- May reduce the amount of time required to retrieve the first row of data for a query.
- Stops the database manager from retrieving a block of data rows that is not used by the program when only the first few rows of a query are retrieved before the query is closed.
- Can degrade the overall performance of a query that retrieves a large number of rows.

***READ**
Records are blocked for read-only retrieval of data for cursors when:

- *NONE is specified for the **Commitment control (COMMIT)** parameter, which indicates that commitment control is not used.
- The cursor is declared with a FOR READ ONLY clause or there are no dynamic statements that could run a positioned UPDATE or DELETE statement for the cursor.

Top

Delay PREPARE (DLYPRP)

Specifies whether the dynamic statement validation for a PREPARE statement is delayed until an OPEN, EXECUTE, or DESCRIBE statement is run. Delaying validation improves performance by eliminating redundant validation.

***NO** Dynamic statement validation is not delayed. When the dynamic statement is prepared, the access plan is validated. When the dynamic statement is used in an OPEN or EXECUTE statement, the access plan is revalidated. Because the authority or the existence of objects referred to by the dynamic statement may change, you must still check the SQLCODE or SQLSTATE after issuing the OPEN or EXECUTE statement to ensure that the dynamic statement is still valid.

***YES** Dynamic statement validation is delayed until the dynamic statement is used in an OPEN, EXECUTE, or DESCRIBE SQL statement. When the dynamic statement is used, the validation is completed and an access plan is built. If you specify *YES for this parameter for precompiled

programs, you should check the SQLCODE and SQLSTATE after running an OPEN, EXECUTE, or DESCRIBE statement to ensure that the dynamic statement is valid.

Note: If you specify *YES, performance is not improved if the INTO clause is used on the PREPARE statement or if a DESCRIBE statement uses the dynamic statement before an OPEN is issued for the statement.

Top

Severity level (GENLVL)

Specifies whether the compiler is called, depending on the severity of messages generated as a result of errors found by the SQL precompiler. If precompiler errors are generated that have a message severity level greater than the value specified for this parameter, the compiler is not called.

If the **Relational database (RDB)** parameter is specified and the severity of the messages generated as a result of package creation is greater than the severity level specified for this parameter, the SQL package is not created.

10 Do not call the compiler if SQL precompiler messages with a message severity greater than 10 are generated.

0-40 Specify the maximum SQL precompiler message severity level to be used to control whether the compiler is called.

Top

Date format (DATFMT)

Specifies the format used when accessing date result columns. All output date fields are returned in the format you specify. For input date strings, the value you specify is used to determine whether the date is a valid format.

Note: An input date string that uses the format *USA, *ISO, *EUR, or *JIS is always valid.

If you connect to a relational database that is on a system that is not a System i, the format must be *USA, *ISO, *EUR, or *JIS.

***JOB** The format specified for the job at precompile time or when a new interactive SQL session is created is used.

Use the Display Job (DSPJOB) command to determine the current date format for the job.

***USA** The United States date format **mm/dd/yyyy** is used.

***ISO** The International Organization for Standardization (ISO) date format **yyyy-mm-dd** is used.

***EUR** The European date format **dd.mm.yyyy** is used.

***JIS** The Japanese Industrial Standard date format **yyyy-mm-dd** is used.

***MDY** The date format **mm/dd/yy** is used.

***DMY** The date format **dd/mm/yy** is used.

***YMD** The date format **yy/mm/dd** is used.

***JUL** The Julian date format **yy/ddd** is used.

Top

Date separator character (DATSEP)

Specifies the separator to be used when accessing date result columns.

Note: This parameter applies only when *JOB, *MDY, *DMY, *YMD, or *JUL is specified for the **Date format (DATFMT)** parameter.

***JOB** The date separator specified for the job at precompile time, when a new interactive SQL session is created, or when Run SQL Statement (RUNSQLSTM) command is run.

Use the Display Job (DSPJOB) command to determine the current date separator value for the job.

'/' A slash is used as the date separator.

'.' A period is used as the date separator.

'-' A dash is used as the date separator.

',' A comma is used as the date separator.

' ' *or* *BLANK
A blank is used as the date separator.

Top

Time format (TIMFMT)

Specifies the format used when accessing time result columns. All output time fields are returned in the format you specify. For input time strings, the value you specify is used to determine whether the time is specified in a valid format.

Note: An input time string that uses the format *USA, *ISO, *EUR, or *JIS is always valid.

If you connect to a relational database that is on a system that is not another System i, the time format must be *USA, *ISO, *EUR, *JIS, or *HMS with a time separator of a colon or period.

***HMS** The **hh:mm:ss** format is used.

***USA** The United States time format **hh:mmxx** is used, where **xx** is AM or PM.

***ISO** The International Organization for Standardization (ISO) time format **hh.mm.ss** is used.

***EUR** The European time format **hh.mm.ss** is used.

***JIS** The Japanese Industrial Standard time format **hh:mm:ss** is used.

Top

Time separator character (TIMSEP)

Specifies the separator used when accessing time result columns.

Note: This parameter applies only when *HMS is specified for the **Time format (TIMFMT)** parameter.

***JOB** The time separator specified for the job at precompile time, when a new interactive SQL session is created, or when RUNSQLSTM is run is used.

Use the Display Job (DSPJOB) command to determine the current time separator value for the job.

':' A colon is used as the time separator.

'.' A period is used as the time separator.

',' A comma is used as the time separator.

' ' or ***BLANK**
A blank is used as the time separator.

Top

Replace (REPLACE)

Specifies whether a SQL program or SQL package is created when there is an existing SQL program or SQL package of the same name in the same library. The value is passed to the CRTxxxPGM command (where xxx is the language of the program being created) and the Create SQL Package (CRTSQLPKG) command if the **Relational database (RDB)** parameter is specified.

***YES** An SQL program or SQL package is created and any existing SQL program or SQL package of the same name in the specified library is moved to QRPLOBJ. The authorities for the existing SQL package are kept for the new SQL package.

***NO** An SQL program or package is not created if an SQL program or package of the same name already exists in the specified library.

Top

RDB connect method (RDBCNNMTH)

Specifies the semantics used for CONNECT statements.

***DUW**
CONNECT (Type 2) semantics are used to support distributed unit of work. Consecutive CONNECT statements to additional relational databases do not result in disconnection of previous connections.

***RUW** CONNECT (Type 1) semantics are used to support remote unit of work. Consecutive CONNECT statements result in the previous connection being disconnected before a new connection is established.

Top

Default collection (DFTRDBCOL)

Specifies the name of the schema identifier used for the unqualified names of the tables, views, indexes, SQL packages, aliases, constraints, external programs, node groups, and triggers. This parameter applies only to static SQL statements.

***NONE**
The naming convention specified for the **Precompiler options (OPTION)** parameter is used.

name Specify the name of the schema identifier to be used instead of the naming convention specified for the **Precompiler options (OPTION)** parameter.

Top

Dynamic default collection (DYNDFTCOL)

Specifies whether the default schema name specified for the **Default collection (DFTRDBCOL)** parameter is also used for dynamic statements.

***NO** Do not use the value specified for the **Default collection (DFTRDBCOL)** parameter for unqualified names of tables, views, indexes, SQL packages, aliases, constraints, external programs, node groups, and triggers for dynamic SQL statements. The naming convention specified for the **Precompiler options (OPTION)** parameter is used.

***YES** The schema name specified for the **Default collection (DFTRDBCOL)** parameter will be used for the unqualified names of the tables, views, indexes, SQL packages, aliases, constraints, external programs, node groups, and triggers in dynamic SQL statements.

Top

Package (SQLPKG)

Specifies the name and library of the SQL package to be created on the remote relational database specified for the **Relational database (RDB)** parameter of this command.

Qualifier 1: Package

***PGM** The name of the SQL package is the same as the program name.

name Specify the name of the SQL package. If the remote system is not a System i, a maximum of 8 characters can be specified.

Qualifier 2: Library

***PGMLIB**
The SQL package is placed in the schema that has the same name as the library containing the program.

name Specify the name of the schema where the SQL package is to be placed. If the remote system is not a System i, a maximum of 8 characters can be specified.

Top

SQL path (SQLPATH)

Specifies the path to be used to find procedures, functions, and user defined types in static SQL statements.

***NAMING**
The path used depends on the naming convention specified for the **Precompiler options (OPTION)** parameter.

For *SYS naming, the path used is *LIBL, the current library list at runtime.

For *SQL naming, the path used is ″QSYS″, ″QSYS2″, ″userid″, where ″userid″ is the value of the USER special register. If a schema name is specified for the **Default collection (DFTRDBCOL)** parameter, the schema name takes the place of userid.

***LIBL** The path used is the library list at runtime.

name Specify one or more schema names. A maximum of 268 schema names may be specified.

Top

SQL rules (SQLCURRULE)

Specifies the semantics used for SQL statements.

***DB2** The semantics of all SQL statements will default to the rules established for DB2. The following semantics are controlled by this option:

Hexadecimal constants are treated as character data.

***STD** The semantics of all SQL statements will default to the rules established by the ISO and ANSI SQL standards. The following semantics are controlled by this option:

Hexadecimal constants are treated as binary data.

Top

IBM SQL flagging (SAAFLAG)

Specifies the IBM SQL flagging function. This parameter allows you to flag SQL statements to verify whether they conform to IBM SQL syntax.

***NOFLAG**
No checks are made to see whether SQL statements conform to IBM SQL syntax.

***FLAG**
Checks are made to see whether SQL statements conform to IBM SQL syntax.

Top

ANS flagging (FLAGSTD)

Specifies whether non-standard statements are flagged. This parameter allows you to flag SQL statements to verify whether they conform to the Core level of the ISO/IEC 9075-2003 standards.

***NONE**
No checks are made to see whether SQL statements conform to ANSI standards.

***ANS** Checks are made to see whether SQL statements conform to standards.

Top

Print file (PRTFILE)

Specifies the printer device file to be used for the precompiler output listing.

Qualifier 1: Print file

QSYSPRT
The precompiler output file is directed to the IBM-supplied printer file, QSYSPRT. The file QSYSPRT has a record length of 132. If you specify a file whose record length is less than 132, information is lost.

name Specify the name of the printer device file to which the precompiler output is directed.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the printer file is located.

Top

User profile (USRPRF)

Specifies the user profile that is used when the compiled program object and SQL package object is run, including the authority that the program object or SQL package has for each object in static SQL statements. The profile of either the owner or the user is used to control access to objects.

***NAMING**
The user profile is determined by the naming convention. If the naming convention is *SQL, USRPRF(*OWNER) is used. If the naming convention is *SYS, USRPRF(*USER) is used.

***USER**
The profile of the user running the program or SQL package is used.

***OWNER**
The user profiles of both the owner and the user are used when the program or SQL package is run.

Top

Dynamic user profile (DYNUSRPRF)

Specifies the user profile used for dynamic SQL statements.

***USER**
Local dynamic SQL statements are run under the profile of the program's user. Distributed dynamic SQL statements are run under the profile of the application server job.

***OWNER**
Local dynamic SQL statements are run under the profile of the program's owner. Distributed dynamic SQL statements are run under the profile of the SQL package's owner.

Top

Sort sequence (SRTSEQ)

Specifies the sort sequence table to be used for string comparisons in SQL statements.

Note: *HEX must be specified for this parameter on distributed applications where the application server is not on a System i.

Single values

***JOB** The SRTSEQ value for the job is used.

***JOBRUN**
The SRTSEQ value for the job is retrieved when the program is run. For distributed applications, SRTSEQ(*JOBRUN) is valid only when LANGID(*JOBRUN) is also specified.

***LANGIDUNQ**
The unique-weight sort table for the language specified for the **Language id (LANGID)** parameter is used.

***LANGIDSHR**
The shared-weight sort table for the language specified for the LANGID parameter is used.

***HEX** A sort sequence table is not used. The hexadecimal values of the characters are used to determine the sort sequence.

Qualifier 1: Sort sequence

name Specify the name of the sort sequence table to be used with this program.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library to be searched.

Top

Language id (LANGID)

Specifies the language identifier to be used when SRTSEQ(*LANGIDUNQ) or SRTSEQ(*LANGIDSHR) is specified.

***JOB** The LANGID value for the job is retrieved during the precompile.

***JOBRUN**
The LANGID value for the job is retrieved when the program is run. For distributed applications, LANGID(*JOBRUN) is valid only when SRTSEQ(*JOBRUN) is also specified.

language-id
Specify the language identifier to be used by the program.

Top

To source file (TOSRCFILE)

Specifies the source file that is to contain the output source member that has been processed by the SQL precompiler. If the specified source file is not found, it will be created. The output member will have the same name as the name specified for the **Object (OBJ)** or **Program (PGM)** parameter.

Qualifier 1: To source file

QSQLTEMP
The source file QSQLTEMP will be used.

name Specify the name of the source file to contain the output source member.

Qualifier 2: Library

QTEMP
The library QTEMP will be used.

***LIBL** The job's library list is searched for the specified file. If the file is not found in any library in the library list, the file will be created in the current library.

***CURLIB**
The current library for the job will be used. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library that is to contain the output source file.

Top

Decimal result options (DECRESULT)

Specifies the maximum precision, maximum scale and minimum divide scale that should be returned for result data types. The specified limit only applies to numeric (zoned) and decimal (packed) data types used in arithmetic expressions and in SQL column functions AVG and SUM.

Element 1: Maximum precision

31 The maximum precision (length) that should be returned for the result data types is 31 digits.

63 The maximum precision (length) that should be returned for the result data types is 63 digits.

Element 2: Maximum scale

31 The maximum scale (number of decimal positions to the right of the decimal point) that should be returned for the result data types is 31 digits.

0-63 Specify the maximum scale (number of decimal positions to the right of the decimal point) that should be returned for the result data types. The value can range from 0 to the maximum precision.

Element 3: Minimum divide scale

0 The minimum divide scale (number of decimal positions to the right of the decimal point) that should be returned for both intermediate and result data types is 0.

0-9 Specify the minimum divide scale (number of decimal positions to the right of the decimal point) that should be returned for both intermediate and result data types. The value cannot to exceed the maximum scale. If 0 is specified for the maximum scale, minimum divide scale is not used.

Top

Decimal float rounding mode (DECFLTRND)

Specifies the decimal floating point rounding mode used for static SQL statements.

***HALFEVEN**
Round to nearest; if equidistant, round so that the final digit is even. If the discarded digits represent greater than half (0.5) the value of a one in the next left position then the result coefficient should be incremented by 1 (rounded up). If they represent less than half, then the result coefficient is not adjusted (that is, the discarded digits are ignored). Otherwise (they represent exactly half) the result coefficient is unaltered if its rightmost digit is even, or incremented by 1 (rounded up) if its rightmost digit is odd (to make an even digit).

***HALFUP**
Round to nearest; if equidistant, round up. If the discarded digits represent greater than or equal to half (0.5) of the value of a one in the next left position then the result coefficient should be incremented by 1 (rounded up). Otherwise the discarded digits are ignored.

***DOWN**
Round towards 0 (truncation). The discarded digits are ignored.

***CEILING**
Round towards +infinity. If all of the discarded digits are zero or if the sign is negative the result is unchanged other than the removal of discarded digits. Otherwise, the result coefficient should be incremented by 1 (rounded up).

***FLOOR**
Round towards -infinity. If all of the discarded digits are zero or if the sign is positive the result is unchanged other than the removal of discarded digits. Otherwise, the sign is negative and the result coefficient should be incremented by 1.

***HALFDOWN**
Round to nearest; if equidistant, round down. If the discarded digits represent greater than half (0.5) of the value of a one in the next left position then the result coefficient should be incremented by 1 (rounded up). Otherwise (the discarded digits are 0.5 or less) the discarded digits are ignored.

***UP** Round away from 0. Of all of the discarded digits are zero the result is unchanged other than the removal of discarded digits. Otherwise, the result coefficient should be incremented by 1 (rounded up).

Top

Compiler options (COMPILEOPT)

Specifies additional parameters to be used on the compiler command. The COMPILEOPT string is added to the compiler command built by the precompiler. There is no validation of the string. The compiler command will issue an error if any parameter is incorrect. Please refer to the DB2 for i5/OS SQL programming topic collection in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/ for a list of parameters that the precompiler generates for the compiler command. Using any of the keywords that the precompiler passes to the compiler will cause the compiler command to fail because of duplicate parameters.

***NONE**
No additional parameters will be used on the compiler command.

character-value
Specify no more than 5000 characters, enclosed in apostrophes.

Top

Examples

```
CRTSQLRPG   PGM(JONES/ARBR5)
            TEXT('Accounts Receivable Branch 5')
```

This command runs the SQL precompiler which precompiles the RPG source and stores the changed source in member ARBR5 of source file QSQLTEMP in library QTEMP. The RPG compiler is called to create program ARBR5 in library JONES by using the source member created by the SQL precompiler.

Top

Error messages

***ESCAPE Messages**

SQL9001
SQL precompile failed.

SQL9002
Conflict in TGTRLS parameters for SQL precompile and &7 compile.

SQL9003
&7 Compile at wrong level for SQL source.

SQL9004
Create of SQL package failed.

SQL9006

DB2 Query Mgr and SQL DevKit not at same install level as the operating system.

Top

Create SQL ILE RPG Object (CRTSQLRPGI)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Create SQL ILE RPG Object (CRTSQLRPGI) command calls the Structured Query Language (SQL) precompiler which precompiles RPG source containing SQL statements, produces a temporary source member, and then optionally calls the ILE RPG compiler to create a module, create a program, or create a service program.

Top

Parameters

Keyword	Description	Choices	Notes
OBJ	Object	*Qualified object name*	Required, Positional 1
	Qualifier 1: Object	*Name*	
	Qualifier 2: Library	*Name*, ***CURLIB**	
SRCFILE	Source file	*Qualified object name*	Optional, Positional 2
	Qualifier 1: Source file	*Name*, **QRPGLESRC**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SRCMBR	Source member	*Name*, ***OBJ**	Optional, Positional 3
SRCSTMF	Source stream file	*Path name*	Optional
COMMIT	Commitment control	***CHG**, *ALL, *CS, *NONE, *RR, *UR, *RS, *NC	Optional
RDB	Relational database	*Simple name*, ***LOCAL**, *NONE	Optional
OBJTYPE	Compile type	***PGM**, *SRVPGM, *MODULE	Optional
OUTPUT	Listing output	***NONE**, *PRINT	Optional
TEXT	Text 'description'	*Character value*, ***SRCMBRTXT**, *BLANK	Optional
USER	RDB user	*Name*, ***CURRENT**	Optional
PASSWORD	RDB user password	*Character value*, ***NONE**, ' '	Optional
OPTION	Precompiler options	Values (up to 14 repetitions): ***XREF**, *NOXREF, *GEN, *NOGEN, *COMMA, *PERIOD, *JOB, *SYSVAL, *SECLVL, *NOSECLVL, *SEQSRC, *NOSEQSRC, *EVENTF, *NOEVENTF, *CVTDT, *NOCVTDT, *SQL, *SYS, *OPTLOB, *NOOPTLOB, *NOEXTIND, *EXTIND	Optional
RPGPPOPT	RPG preprocessor options	***NONE**, *LVL1, *LVL2	Optional
TGTRLS	Target release	*Simple name*, ***CURRENT**, *PRV	Optional
INCFILE	INCLUDE file	*Qualified object name*	Optional
	Qualifier 1: INCLUDE file	*Name*, ***SRCFILE**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
INCDIR	SQL INCLUDE directory	*Path name*, ***NONE**	Optional
ALWCPYDTA	Allow copy of data	***OPTIMIZE**, *YES, *NO	Optional
CLOSQLCSR	Close SQL cursor	***ENDACTGRP**, *ENDMOD	Optional
ALWBLK	Allow blocking	***ALLREAD**, *NONE, *READ	Optional
DLYPRP	Delay PREPARE	***NO**, *YES	Optional
GENLVL	Severity level	0-40, **10**	Optional

Keyword	Description	Choices	Notes
DATFMT	Date format	***JOB**, *USA, *ISO, *EUR, *JIS, *MDY, *DMY, *YMD, *JUL	Optional
DATSEP	Date separator character	***JOB**, '/', '.', ',', '-', ' ', *BLANK	Optional
TIMFMT	Time format	***HMS**, *USA, *ISO, *EUR, *JIS	Optional
TIMSEP	Time separator character	***JOB**, ':', '.', ',', ' ', *BLANK	Optional
REPLACE	Replace	***YES**, *NO	Optional
RDBCNNMTH	RDB connect method	***DUW**, *RUW	Optional
DFTRDBCOL	Default collection	*Name*, ***NONE**	Optional
DYNDFTCOL	Dynamic default collection	***NO**, *YES	Optional
SQLPKG	Package	*Qualified object name*	Optional
	Qualifier 1: Package	*Name*, ***OBJ**	
	Qualifier 2: Library	*Name*, ***OBJLIB**	
SQLPATH	SQL path	Single values: ***NAMING**, *LIBL Other values (up to 268 repetitions): *Name*	Optional
SQLCURRULE	SQL rules	***DB2**, *STD	Optional
SAAFLAG	IBM SQL flagging	***NOFLAG**, *FLAG	Optional
FLAGSTD	ANS flagging	***NONE**, *ANS	Optional
PRTFILE	Print file	*Qualified object name*	Optional
	Qualifier 1: Print file	*Name*, **QSYSPRT**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
DBGVIEW	Debugging view	***NONE**, *SOURCE	Optional
USRPRF	User profile	***NAMING**, *USER, *OWNER	Optional
DYNUSRPRF	Dynamic user profile	***USER**, *OWNER	Optional
SRTSEQ	Sort sequence	Single values: ***JOB**, *HEX, *JOBRUN, *LANGIDUNQ, *LANGIDSHR Other values: *Qualified object name*	Optional
	Qualifier 1: Sort sequence	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
LANGID	Language id	*Character value*, ***JOB**, *JOBRUN	Optional
TOSRCFILE	To source file	*Qualified object name*	Optional
	Qualifier 1: To source file	*Name*, **QSQLTEMP1**	
	Qualifier 2: Library	*Name*, **QTEMP**, *LIBL, *CURLIB	
DECRESULT	Decimal result options	*Element list*	Optional
	Element 1: Maximum precision	**31**, 63	
	Element 2: Maximum scale	0-63, **31**	
	Element 3: Minimum divide scale	0-9, **0**	
DECFLTRND	Decimal float rounding mode	***HALFEVEN**, *HALFUP, *DOWN, *CEILING, *FLOOR, *HALFDOWN, *UP	Optional
COMPILEOPT	Compiler options	*Character value*, ***NONE**	Optional

Top

Object (OBJ)

Specifies the object to be created.

This is a required parameter.

Qualifier 1: Object

name Specify the name of the object to be created.

Qualifier 2: Library

***CURLIB**
The new object is created in the current library for the job. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the object is created.

Top

Source file (SRCFILE)

Specifies the source file that contains the RPG source statements and SQL statements.

Qualifier 1: Source file

QRPGLESRC
Source file QRPGLESRC contains the RPG source.

name Specify the name of the source file that contains the RPG source.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library to be searched.

Top

Source member (SRCMBR)

Specifies the source file member that contains the input source. This parameter is used only if the source file specified for the **Source file (SRCFILE)** parameter is a database file.

***OBJ** The source file member that has the same name as the value specified for the **Object (OBJ)** parameter contains the input source.

name Specify the name of the source file member that contains the input source.

Top

Source stream file (SRCSTMF)

Specifies the path name to the file that contains the RPG source statements and SQL statements. The path name can be either absolute or relative.

Source text must be in positions 1 to 100. Characters after position 100 will be ignored.

If a value is specified for this parameter, the precompiler will not expand /COPY statements. If a /COPY statement needs to be expanded for the precompile, you must specify a value other than *NONE on the **RPG preprocessor options (RPGPPOPT)** parameter.

Top

Commitment control (COMMIT)

Specifies whether SQL statements in the compiled program are run under commitment control. Files referred to in the host language source are not affected by this option. Only SQL tables, SQL views, SQL packages, SQL sequences, SQL aliases, SQL Types, SQL procedures, SQL functions, SQL indexes, SQL schemas, SQL triggers, and SQL views referred to in SQL statements are affected.

***CHG** *or* ***UR**
Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs can be seen.

***CS** Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows updated, deleted, and inserted are locked until the end of the unit of work (transaction). A row that is selected, but not updated, is locked until the next row is selected. Uncommitted changes in other jobs cannot be seen.

***ALL** *or* ***RS**
Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows selected, updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs cannot be seen.

***NONE** *or* ***NC**
Specifies that commitment control is not used. Uncommitted changes in other jobs can be seen. If the SQL DROP SCHEMA statement is included in the program, *NONE or *NC must be used. If a relational database is specified for the **Relational database (RDB)** parameter, and the relational database is on a system that is not on a System i, *NONE or *NC cannot be specified.

***RR** Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows selected, updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs cannot be seen. All tables referred to in SELECT, UPDATE, DELETE, and INSERT statements are locked exclusively until the end of the unit of work (transaction).

Top

Relational database (RDB)

Specifies the name of the relational database where the SQL package is to be created.

***LOCAL**
The program is created as a distributed SQL program. The SQL statements will access the local database. An SQL package object is not created as part of the precompile process. The Create Structured Query Language Package (CRTSQLPKG) command can be used.

***NONE**
An SQL package object is not created. The program object is not a distributed program and the Create Structured Query Language Package (CRTSQLPKG) command cannot be used.

name Specify the name of the relational database where the new SQL package object is to be created. When the name of the local relational database is specified, the program created is still a distributed SQL program. The SQL statements will access the local database.

Top

Compile type (OBJTYPE)

Specifies the type of object to be created.

When OBJTYPE(*PGM) or OBJTYPE(*SRVPGM) is specified and the **Relational database (RDB)** parameter is also specified, the CRTSQLPKG command is issued by the SQL precompiler after the program has been created. When OBJTYPE(*MODULE) is specified, an SQL package is not created and you must issue the CRTSQLPKG command after the CRTPGM or CRTSRVPGM command has created the program.

If OPTION(*NOGEN) is specified, only the SQL temporary source member is generated. No module, program, service program, or SQL package is created.

***PGM** The SQL precompiler calls the compiler to create a program.

***MODULE**
The SQL precompiler calls the compiler to create a module.

***SRVPGM**
The SQL precompiler calls the compiler to create a module and issues the Create Service Program (CRTSRVPGM) command to create a service program.

Top

Listing output (OUTPUT)

Specifies whether the precompiler listing is generated.

***NONE**
The precompiler listing is not generated.

***PRINT**
The precompiler listing is generated.

Top

Text 'description' (TEXT)

Specifies text that briefly describes the program and its function.

***SRCMBRTXT**
The text is taken from the source file member being used to create the program. If the source file is an inline file or a device file, the text is blank.

***BLANK**
No text is specified.

'description'
Specify no more than 50 characters, enclosed in apostrophes.

Top

RDB user (USER)

Specifies the user name sent to the remote system when starting the conversation. This parameter is valid only when **Relational database (RDB)** is specified.

***CURRENT**
The user name associated with the current job is used.

name Specify the user name to be used for the application server job.

Top

RDB user password (PASSWORD)

Specifies the password to be used on the remote system. This parameter is valid only when **Relational database (RDB)** is specified.

***NONE**
No password is sent. A user name cannot be specified for the **RDB user (USER)** parameter if this value is specified.

Note: Specifying a password of a blank is the same as specifying *NONE.

password
Specify the password of the user name specified for the **RDB user (USER)** parameter.

Top

Precompiler options (OPTION)

Specifies whether one or more of the following options are used when the RPG source is precompiled. If an option is specified more than once, or if two options conflict, the last option specified is used.

Cross reference options:

***XREF** The precompiler cross-references items in the program to the statement numbers in the program that refer to those items.

***NOXREF**
The precompiler does not cross-reference names.

Object creation options:

***GEN** The precompiler creates the object that is specified for the **Compile type (OBJTYPE)** parameter.

***NOGEN**
The precompiler does not call the RPG compiler. No module, program, service program, or SQL package is created.

Decimal point options:

***JOB** The representation for the decimal point specified for the job at precompile time is used.

Note: If QDECFMT specifies that the value used as the decimal point is a comma, any numeric constants in lists (such as in the SELECT clause and VALUES clause) must be separated by a comma followed by a blank. For example, VALUES(1,1, 2,23, 4,1) is equivalent to VALUES(1.1,2.23,4.1) where the decimal point is the period.

***SYSVAL**
The value used as the decimal point in numeric constants is from the QDECFMT system value. This value is also used as the decimal point character when casting a numeric value to character.

Note: If QDECFMT specifies that the value used as the decimal point is a comma; any numeric constants in lists (such as in the SELECT clause and VALUES clause) must be separated by a comma followed by a blank. For example, VALUES(1,1, 2,23, 4,1) is equivalent to VALUES(1.1,2.23,4.1) where the decimal point is the period.

***PERIOD**
The value used as the decimal point for numeric constants used in SQL statements is a period. This value is also used as the decimal point character when casting a numeric value to character.

***COMMA**
The value used as the decimal point in numeric constants is a comma. Any numeric constants in lists (such as in the SELECT clause and VALUES clause) must be separated by a comma followed by a blank. For example, VALUES(1,1, 2,23, 4,1) is equivalent to VALUES(1.1,2.23,4.1) where the decimal point is the period.

Naming convention options:

***SYS** Specifies that the system naming convention is used (library-name/file-name).

***SQL** Specifies that the SQL naming convention is used (schema-name.table-name).

If a relational database is specified for the **Relational database (RDB)** parameter, and the database is on a system that is not a System i, *SQL must be specified as the naming convention.

Second-level message text options:

***NOSECLVL**
Second-level text descriptions are not added to the listing.

***SECLVL**
Second-level text with replacement data is added for all messages on the listing.

Sequence source options:

***NOSEQSRC**
The source file member created into QSQLTEMP1 or QSQLTEMP has the same sequence numbers as the original source read by the precompiler.

***SEQSRC**
The source file member created into QSQLTEMP1 or QSQLTEMP contains sequence numbers starting at 000001 and incremented by 000001.

LOB optimization for DRDA options:

***OPTLOB**
The first FETCH for a cursor determines how the cursor will be used for LOBs (Large Objects) on all subsequent FETCHes. This option remains in effect until the cursor is closed.

If the first FETCH uses a LOB locator to access a LOB column, no subsequent FETCH for that cursor can fetch that LOB column into a LOB host variable.

If the first FETCH places the LOB column into a LOB host variable, no subsequent FETCH for that cursor can use a LOB locator for that column.

***NOOPTLOB**
There is no restriction on whether a column is retrieved into a LOB locator or into a LOB host variable. This option can cause performance to degrade.

Event file creation options:

***NOEVENTF**
The compiler will not produce an event file for use by CoOperative Development Environment (CODE).

***EVENTF**
The compiler produces an event file for use by CoOperative Development Environment (CODE). The event file will be created as a member in the file EVFEVENT in your object library. CODE uses this file to offer error feedback integrated with the CODE editor. This option is normally specified by CODE on your behalf.

Character string representation options:

***NOCVTDT**
Specifies that date, time, and timestamp data types which are retrieved from externally-described database files are to be processed using the date, time, and timestamp data types.

***CVTDT**
Specifies that date, time and timestamp data types, which are retrieved from externally-described database files, are to be processed as fixed-length character fields.

Extended indicators options:

***NOEXTIND**
Extended indicator support is not enabled.

***EXTIND**
Extended indicator support is enabled.

Top

RPG preprocessor options (RPGPPOPT)

Specifies if the ILE RPG compiler will be called to preprocess the source member before the SQL precompile is run. Preprocessing the SQL source member will allow some compiler directives to be handled before the SQL precompile. The preprocessed source will be placed in file QSQLPRE in QTEMP. This source will be used for the SQL precompile.

***NONE**
The compiler is not called for preprocessing.

***LVL1** The compiler is called for preprocessing to expand /COPY and handle the conditional compilation directives except the /INCLUDE directive.

***LVL2** The compiler will be called for preprocessing to expand /COPY and /INCLUDE and handle the conditional compilation directives.

Top

Target release (TGTRLS)

Specifies the release of the operating system on which you intend to use the object being created.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

***CURRENT**
The object is to be used on the release of the operating system currently running on your system. The object can also be used on a system with any subsequent release of the operating system installed.

***PRV** The object is to be used on the previous release with modification level 0 of the operating system. The object can also be used on a system with any subsequent release of the operating system installed.

target-release
Specify the release in the format VxRxMx. The object can be used on a system with the specified release or with any subsequent release of the operating system installed.

Top

INCLUDE file (INCFILE)

Specifies the source file that contains members to be included in the program with the SQL INCLUDE statement.

Single values

***SRCFILE**
The qualified source file you specify for the **Source file (SRCFILE)** parameter contains the source file members specified on any SQL INCLUDE statements.

Qualifier 1: INCLUDE file

name Specify the name of the source file that contains the source file members specified on any SQL INCLUDE statements.

The record length of the source file you specify here must be no less than the record length of the source file you specify for the **Source file (SRCFILE)** parameter.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the source file is located.

Top

SQL INCLUDE directory (INCDIR)

Specifies the path name of the directory containing any files to be included in the program with the SQL INCLUDE statement.

Specifying a value for this parameter does not pass an INCDIR value from the SQL precompiler to the compiler. If you want to pass an INCDIR value through to the compiler, this can be done using the **Compiler options (COMPILEOPT)** parameter.

***NONE**
The current directory and the source directory will be searched.

path-name
The path name of the directory that contains the files specified on any SQL INCLUDE statement. The current directory will be searched first, then the specified path name, then the source directory.

Top

Allow copy of data (ALWCPYDTA)

Specifies whether a copy of the data can be used in a SELECT statement.

***OPTIMIZE**
The system determines whether to use the data retrieved directly from the database or to use a copy of the data. The decision is based on which method provides the best performance. If the **Commitment control (COMMIT)** parameter is not *NONE, the **Allow blocking (ALWBLK)** parameter should be set to *ALLREAD, when possible, for best performance.

***YES** A copy of the data is used only when necessary.

***NO** A copy of the data is not used. If a temporary copy of the data is required to perform the query, an error message is returned.

Top

Close SQL cursor (CLOSQLCSR)

Specifies when SQL cursors are implicitly closed, SQL prepared statements are implicitly discarded, and LOCK TABLE locks are released. SQL cursors are explicitly closed when the user issues the CLOSE, COMMIT, or ROLLBACK (without HOLD) SQL statements.

***ENDACTGRP**
SQL cursors are closed and SQL prepared statements are implicitly discarded, and LOCK TABLE locks are released when the activation group ends.

***ENDMOD**
SQL cursors are closed and SQL prepared statements are implicitly discarded when the module is exited. LOCK TABLE locks are released when the first SQL program on the call stack ends.

Top

Allow blocking (ALWBLK)

Specifies whether the database manager can use record blocking and the extent to which blocking can be used for read-only cursors.

***ALLREAD**
Rows are blocked for read-only cursors. All cursors in a program that are not explicitly able to be changed are opened for read-only processing even though there may be EXECUTE or EXECUTE IMMEDIATE statements in the program.

Specifying *ALLREAD:

- Allows record blocking for all read-only cursors.

- Can improve the performance of almost all read-only cursors in programs, but limits queries in the following ways:
  - The Rollback (ROLLBACK) command, a ROLLBACK statement in host languages, or the ROLLBACK HOLD SQL statement does not reposition a read-only cursor when *ALLREAD is specified.
  - Dynamic running of a positioned UPDATE or DELETE statement (for example, using EXECUTE IMMEDIATE), can not be used to update a row in a cursor unless the DECLARE statement for the cursor includes the FOR UPDATE clause.

***NONE**

Rows are not blocked for retrieval of data for cursors.

Specifying *NONE:

- Guarantees that the data retrieved is current.
- May reduce the amount of time required to retrieve the first row of data for a query.
- Stops the database manager from retrieving a block of data rows that is not used by the program when only the first few rows of a query are retrieved before the query is closed.
- Can degrade the overall performance of a query that retrieves a large number of rows.

***READ**

Records are blocked for read-only retrieval of data for cursors when:

- *NONE is specified for the **Commitment control (COMMIT)** parameter, which indicates that commitment control is not used.
- The cursor is declared with a FOR READ ONLY clause or there are no dynamic statements that could run a positioned UPDATE or DELETE statement for the cursor.

Top

Delay PREPARE (DLYPRP)

Specifies whether the dynamic statement validation for a PREPARE statement is delayed until an OPEN, EXECUTE, or DESCRIBE statement is run. Delaying validation improves performance by eliminating redundant validation.

***NO** Dynamic statement validation is not delayed. When the dynamic statement is prepared, the access plan is validated. When the dynamic statement is used in an OPEN or EXECUTE statement, the access plan is revalidated. Because the authority or the existence of objects referred to by the dynamic statement may change, you must still check the SQLCODE or SQLSTATE after issuing the OPEN or EXECUTE statement to ensure that the dynamic statement is still valid.

***YES** Dynamic statement validation is delayed until the dynamic statement is used in an OPEN, EXECUTE, or DESCRIBE SQL statement. When the dynamic statement is used, the validation is completed and an access plan is built. If you specify *YES for this parameter for precompiled programs, you should check the SQLCODE and SQLSTATE after running an OPEN, EXECUTE, or DESCRIBE statement to ensure that the dynamic statement is valid.

Note: If you specify *YES, performance is not improved if the INTO clause is used on the PREPARE statement or if a DESCRIBE statement uses the dynamic statement before an OPEN is issued for the statement.

Top

Severity level (GENLVL)

Specifies whether the compiler is called, depending on the severity of messages generated as a result of errors found by the SQL precompiler. If precompiler errors are generated that have a message severity level greater than the value specified for this parameter, the compiler is not called.

If the **Relational database (RDB)** parameter is specified and the severity of the messages generated as a result of package creation is greater than the severity level specified for this parameter, the SQL package is not created.

10 Do not call the compiler if SQL precompiler messages with a message severity greater than 10 are generated.

0-40 Specify the maximum SQL precompiler message severity level to be used to control whether the compiler is called.

Top

Date format (DATFMT)

Specifies the format used when accessing date result columns. All output date fields are returned in the format you specify. For input date strings, the value you specify is used to determine whether the date is a valid format.

Note: An input date string that uses the format *USA, *ISO, *EUR, or *JIS is always valid.

If you connect to a relational database that is on a system that is not a System i, the format must be *USA, *ISO, *EUR, or *JIS.

***JOB** The format specified for the job at precompile time or when a new interactive SQL session is created is used.

Use the Display Job (DSPJOB) command to determine the current date format for the job.

***USA** The United States date format **mm/dd/yyyy** is used.

***ISO** The International Organization for Standardization (ISO) date format **yyyy-mm-dd** is used.

***EUR** The European date format **dd.mm.yyyy** is used.

***JIS** The Japanese Industrial Standard date format **yyyy-mm-dd** is used.

***MDY** The date format **mm/dd/yy** is used.

***DMY** The date format **dd/mm/yy** is used.

***YMD** The date format **yy/mm/dd** is used.

***JUL** The Julian date format **yy/ddd** is used.

Top

Date separator character (DATSEP)

Specifies the separator to be used when accessing date result columns.

Note: This parameter applies only when *JOB, *MDY, *DMY, *YMD, or *JUL is specified for the **Date format (DATFMT)** parameter.

***JOB** The date separator specified for the job at precompile time, when a new interactive SQL session is created, or when Run SQL Statement (RUNSQLSTM) command is run.

Use the Display Job (DSPJOB) command to determine the current date separator value for the job.

'/' A slash is used as the date separator.

'.' A period is used as the date separator.

'-' A dash is used as the date separator.

',' A comma is used as the date separator.

' ' *or* *BLANK
A blank is used as the date separator.

Top

Time format (TIMFMT)

Specifies the format used when accessing time result columns. All output time fields are returned in the format you specify. For input time strings, the value you specify is used to determine whether the time is specified in a valid format.

Note: An input time string that uses the format *USA, *ISO, *EUR, or *JIS is always valid.

If you connect to a relational database that is on a system that is not another System i, the time format must be *USA, *ISO, *EUR, *JIS, or *HMS with a time separator of a colon or period.

***HMS** The **hh:mm:ss** format is used.

***USA** The United States time format **hh:mmxx** is used, where **xx** is AM or PM.

***ISO** The International Organization for Standardization (ISO) time format **hh.mm.ss** is used.

***EUR** The European time format **hh.mm.ss** is used.

***JIS** The Japanese Industrial Standard time format **hh:mm:ss** is used.

Top

Time separator character (TIMSEP)

Specifies the separator used when accessing time result columns.

Note: This parameter applies only when *HMS is specified for the **Time format (TIMFMT)** parameter.

***JOB** The time separator specified for the job at precompile time, when a new interactive SQL session is created, or when RUNSQLSTM is run is used.

Use the Display Job (DSPJOB) command to determine the current time separator value for the job.

':' A colon is used as the time separator.

'.' A period is used as the time separator.

',' A comma is used as the time separator.

' ' *or* *BLANK
A blank is used as the time separator.

Top

Replace (REPLACE)

Specifies if a SQL module, program, service program or package is created when there is an existing SQL module, program, service program, or package of the same name and type in the same library. The value for this parameter is passed to the CRTRPGMOD, CRTBNDRPG, CRTSRVPGM, and CRTSQLPKG commands.

***YES** A new SQL module, program, service program, or package is created, any existing SQL object of the same name and type in the specified library is moved to QRPLOBJ.

***NO** A new SQL module, program, service program, or package is not created if an SQL object of the same name and type already exists in the specified library.

Top

RDB connect method (RDBCNNMTH)

Specifies the semantics used for CONNECT statements.

***DUW**
CONNECT (Type 2) semantics are used to support distributed unit of work. Consecutive CONNECT statements to additional relational databases do not result in disconnection of previous connections.

***RUW** CONNECT (Type 1) semantics are used to support remote unit of work. Consecutive CONNECT statements result in the previous connection being disconnected before a new connection is established.

Top

Default collection (DFTRDBCOL)

Specifies the name of the schema identifier used for the unqualified names of the tables, views, indexes, SQL packages, aliases, constraints, external programs, node groups, and triggers. This parameter applies only to static SQL statements.

***NONE**
The naming convention specified for the **Precompiler options (OPTION)** parameter is used.

name Specify the name of the schema identifier to be used instead of the naming convention specified for the **Precompiler options (OPTION)** parameter.

Top

Dynamic default collection (DYNDFTCOL)

Specifies whether the default schema name specified for the **Default collection (DFTRDBCOL)** parameter is also used for dynamic statements.

***NO** Do not use the value specified for the **Default collection (DFTRDBCOL)** parameter for unqualified names of tables, views, indexes, SQL packages, aliases, constraints, external programs, node groups, and triggers for dynamic SQL statements. The naming convention specified for the **Precompiler options (OPTION)** parameter is used.

***YES** The schema name specified for the **Default collection (DFTRDBCOL)** parameter will be used for the unqualified names of the tables, views, indexes, SQL packages, aliases, constraints, external programs, node groups, and triggers in dynamic SQL statements.

Top

Package (SQLPKG)

Specifies the qualified name of the SQL package created on the relational database specified for the **Relational database (RDB)** parameter of this command.

Qualifier 1: Package

***OBJ** The name of the SQL package is the same as the object name specified for the **Object (OBJ)** parameter.

name Specify the name of the SQL package. If the remote system is not a System i, no more than 8 characters can be specified.

Qualifier 2: Library

***OBJLIB**
The package is created in the library with the same name as the library specified for the **Object (OBJ)** parameter.

name Specify the name of the library where the package is created.

Top

SQL path (SQLPATH)

Specifies the path to be used to find procedures, functions, and user defined types in static SQL statements.

***NAMING**
The path used depends on the naming convention specified for the **Precompiler options (OPTION)** parameter.

For *SYS naming, the path used is *LIBL, the current library list at runtime.

For *SQL naming, the path used is ″QSYS″, ″QSYS2″, ″userid″, where ″userid″ is the value of the USER special register. If a schema name is specified for the **Default collection (DFTRDBCOL)** parameter, the schema name takes the place of userid.

***LIBL** The path used is the library list at runtime.

name Specify one or more schema names. A maximum of 268 schema names may be specified.

Top

SQL rules (SQLCURRULE)

Specifies the semantics used for SQL statements.

***DB2** The semantics of all SQL statements will default to the rules established for DB2. The following semantics are controlled by this option:

Hexadecimal constants are treated as character data.

***STD** The semantics of all SQL statements will default to the rules established by the ISO and ANSI SQL standards. The following semantics are controlled by this option:

Hexadecimal constants are treated as binary data.

Top

IBM SQL flagging (SAAFLAG)

Specifies the IBM SQL flagging function. This parameter allows you to flag SQL statements to verify whether they conform to IBM SQL syntax.

***NOFLAG**
No checks are made to see whether SQL statements conform to IBM SQL syntax.

***FLAG**
Checks are made to see whether SQL statements conform to IBM SQL syntax.

Top

ANS flagging (FLAGSTD)

Specifies whether non-standard statements are flagged. This parameter allows you to flag SQL statements to verify whether they conform to the Core level of the ISO/IEC 9075-2003 standards.

***NONE**
No checks are made to see whether SQL statements conform to ANSI standards.

***ANS** Checks are made to see whether SQL statements conform to standards.

Top

Print file (PRTFILE)

Specifies the printer device file to be used for the precompiler output listing.

Qualifier 1: Print file

QSYSPRT
The precompiler output file is directed to the IBM-supplied printer file, QSYSPRT. The file QSYSPRT has a record length of 132. If you specify a file whose record length is less than 132, information is lost.

name Specify the name of the printer device file to which the precompiler output is directed.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the printer file is located.

Top

Debugging view (DBGVIEW)

Specifies the type of source debug information to be provided by the SQL precompiler.

***NONE**
The source view will not be generated.

***SOURCE**
The SQL precompiler will provide the source views for the root and if necessary, SQL INCLUDE statements. A view is to be provided which contains the statements generated by the precompiler.

Top

User profile (USRPRF)

Specifies the user profile that is used when the compiled program object and SQL package object is run, including the authority that the program object or SQL package has for each object in static SQL statements. The profile of either the owner or the user is used to control access to objects.

***NAMING**
The user profile is determined by the naming convention. If the naming convention is *SQL, USRPRF(*OWNER) is used. If the naming convention is *SYS, USRPRF(*USER) is used.

***USER**
The profile of the user running the program or SQL package is used.

***OWNER**
The user profiles of both the owner and the user are used when the program or SQL package is run.

Top

Dynamic user profile (DYNUSRPRF)

Specifies the user profile used for dynamic SQL statements.

***USER**
Local dynamic SQL statements are run under the profile of the program's user. Distributed dynamic SQL statements are run under the profile of the application server job.

***OWNER**
Local dynamic SQL statements are run under the profile of the program's owner. Distributed dynamic SQL statements are run under the profile of the SQL package's owner.

Top

Sort sequence (SRTSEQ)

Specifies the sort sequence table to be used for string comparisons in SQL statements.

Note: *HEX must be specified for this parameter on distributed applications where the application server is not on a System i.

Single values

***JOB** The SRTSEQ value for the job is used.

***JOBRUN**
The SRTSEQ value for the job is retrieved when the program is run. For distributed applications, SRTSEQ(*JOBRUN) is valid only when LANGID(*JOBRUN) is also specified.

***LANGIDUNQ**
The unique-weight sort table for the language specified for the **Language id (LANGID)** parameter is used.

***LANGIDSHR**
The shared-weight sort table for the language specified for the LANGID parameter is used.

***HEX** A sort sequence table is not used. The hexadecimal values of the characters are used to determine the sort sequence.

Qualifier 1: Sort sequence

name Specify the name of the sort sequence table to be used with this program.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library to be searched.

Top

Language id (LANGID)

Specifies the language identifier to be used when SRTSEQ(*LANGIDUNQ) or SRTSEQ(*LANGIDSHR) is specified.

***JOB** The LANGID value for the job is retrieved during the precompile.

***JOBRUN**
The LANGID value for the job is retrieved when the program is run. For distributed applications, LANGID(*JOBRUN) is valid only when SRTSEQ(*JOBRUN) is also specified.

language-id
Specify the language identifier to be used by the program.

Top

To source file (TOSRCFILE)

Specifies the source file that is to contain the output source member that has been processed by the SQL precompiler. If the specified source file is not found, it will be created. The output member will have the same name as the name specified for the **Object (OBJ)** parameter. If this is an existing source file, it should have a record length of 112.

Qualifier 1: To source file

QSQLTEMP1
The source file QSQLTEMP1 will be used.

name Specify the name of the source file to contain the output source member.

Qualifier 2: Library

QTEMP
The library QTEMP will be used.

***LIBL** The job's library list is searched for the specified file. If the file is not found in any library in the library list, the file will be created in the current library.

***CURLIB**
The current library for the job will be used. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library that is to contain the output source file.

Top

Decimal result options (DECRESULT)

Specifies the maximum precision, maximum scale and minimum divide scale that should be returned for result data types. The specified limit only applies to numeric (zoned) and decimal (packed) data types used in arithmetic expressions and in SQL column functions AVG and SUM.

Element 1: Maximum precision

31 The maximum precision (length) that should be returned for the result data types is 31 digits.

63 The maximum precision (length) that should be returned for the result data types is 63 digits.

Element 2: Maximum scale

31 The maximum scale (number of decimal positions to the right of the decimal point) that should be returned for the result data types is 31 digits.

0-63 Specify the maximum scale (number of decimal positions to the right of the decimal point) that should be returned for the result data types. The value can range from 0 to the maximum precision.

Element 3: Minimum divide scale

0 The minimum divide scale (number of decimal positions to the right of the decimal point) that should be returned for both intermediate and result data types is 0.

0-9 Specify the minimum divide scale (number of decimal positions to the right of the decimal point) that should be returned for both intermediate and result data types. The value cannot to exceed the maximum scale. If 0 is specified for the maximum scale, minimum divide scale is not used.

Top

Decimal float rounding mode (DECFLTRND)

Specifies the decimal floating point rounding mode used for static SQL statements.

***HALFEVEN**
Round to nearest; if equidistant, round so that the final digit is even. If the discarded digits represent greater than half (0.5) the value of a one in the next left position then the result coefficient should be incremented by 1 (rounded up). If they represent less than half, then the result coefficient is not adjusted (that is, the discarded digits are ignored). Otherwise (they represent exactly half) the result coefficient is unaltered if its rightmost digit is even, or incremented by 1 (rounded up) if its rightmost digit is odd (to make an even digit).

***HALFUP**
Round to nearest; if equidistant, round up. If the discarded digits represent greater than or equal to half (0.5) of the value of a one in the next left position then the result coefficient should be incremented by 1 (rounded up). Otherwise the discarded digits are ignored.

***DOWN**
Round towards 0 (truncation). The discarded digits are ignored.

***CEILING**
Round towards +infinity. If all of the discarded digits are zero or if the sign is negative the result is unchanged other than the removal of discarded digits. Otherwise, the result coefficient should be incremented by 1 (rounded up).

***FLOOR**
Round towards -infinity. If all of the discarded digits are zero or if the sign is positive the result is unchanged other than the removal of discarded digits. Otherwise, the sign is negative and the result coefficient should be incremented by 1.

***HALFDOWN**
Round to nearest; if equidistant, round down. If the discarded digits represent greater than half (0.5) of the value of a one in the next left position then the result coefficient should be incremented by 1 (rounded up). Otherwise (the discarded digits are 0.5 or less) the discarded digits are ignored.

***UP** Round away from 0. Of all of the discarded digits are zero the result is unchanged other than the removal of discarded digits. Otherwise, the result coefficient should be incremented by 1 (rounded up).

Top

Compiler options (COMPILEOPT)

Specifies additional parameters to be used on the compiler command. The COMPILEOPT string is added to the compiler command built by the precompiler. If **INCDIR(** is anywhere in the string, the precompiler will call the compiler using the SRCSTMF parameter. There is no validation of the string. The compiler command will issue an error if any parameter is incorrect. Please refer to the DB2 for i5/OS SQL programming topic collection in the i5/OS Information Center at http://www.ibm.com/systems/i/infocenter/ for a list of parameters that the precompiler generates for the compiler command. Using any of the keywords that the precompiler passes to the compiler will cause the compiler command to fail because of duplicate parameters.

***NONE**
No additional parameters will be used on the compiler command.

character-value
Specify no more than 5000 characters, enclosed in apostrophes.

Top

Examples

```
CRTSQLRPGI   OBJ(PAYROLL)  OBJTYPE(*PGM)
             TEXT('Payroll Program')
```

This command runs the SQL precompiler which precompiles the RPG source and stores the changed source in member PAYROLL of source file QSQLTEMP1 in library QTEMP. The ILE RPG compiler is called to create program PAYROLL in the current library by using the source member created by the SQL precompiler.

Top

Error messages

***ESCAPE Messages**

SQL9001
SQL precompile failed.

SQL9002
Conflict in TGTRLS parameters for SQL precompile and &7 compile.

SQL9003
&7 Compile at wrong level for SQL source.

SQL9004
Create of SQL package failed.

SQL9006
DB2 Query Mgr and SQL DevKit not at same install level as the operating system.

SQL9019
SQL precompile failed.

Top

Convert SQL C++ Source (CVTSQLCPP)

Where allowed to run: All environments (*ALL)
Threadsafe: No

Parameters
Examples
Error messages

The Convert Structured Query Language C++ Source (CVTSQLCPP) command calls the Structured Query Language (SQL) precompiler which precompiles C++ source containing SQL statements and produces an output source member. This source member can then be provided as input to the C++ cross compiler. For most C++ precompiles, the CRTSQLCPPI command should be use.

Top

Parameters

Keyword	Description	Choices	Notes
SRCFILE	Source file	*Qualified object name*	Optional, Positional 1
	Qualifier 1: Source file	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
SRCMBR	Source member	*Name*	Optional, Positional 2
SRCSTMF	Source stream file	*Path name*	Optional
TOSRCFILE	To source file	*Qualified object name*	Optional
	Qualifier 1: To source file	*Name*, ***CALC**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
COMMIT	Commitment control	***CHG**, *ALL, *CS, *NONE, *RR, *UR, *RS, *NC	Optional
RDB	Relational database	*Simple name*, ***LOCAL**, *NONE	Optional
OUTPUT	Listing output	***NONE**, *PRINT	Optional
TEXT	Text 'description'	*Character value*, ***SRCMBRTXT**, *BLANK	Optional
USER	RDB user	*Name*, ***CURRENT**	Optional
PASSWORD	RDB user password	*Character value*, ***NONE**, ' '	Optional
OPTION	Precompiler options	Values (up to 12 repetitions): ***XREF**, *NOXREF, *CNULRQD, *NOCNULRQD, *SQL, *SYS, *JOB, *SYSVAL, *PERIOD, *COMMA, *EVENTF, *NOEVENTF, *SECLVL, *NOSECLVL, *OPTLOB, *NOOPTLOB, *NOEXTIND, *EXTIND	Optional, Positional 3
TGTRLS	Target release	*Simple name*, ***CURRENT**	Optional
INCFILE	INCLUDE file	*Qualified object name*	Optional
	Qualifier 1: INCLUDE file	*Name*, ***SRCFILE**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
INCDIR	SQL INCLUDE directory	*Path name*, ***NONE**	Optional
ALWCPYDTA	Allow copy of data	***OPTIMIZE**, *YES, *NO	Optional
CLOSQLCSR	Close SQL cursor	***ENDACTGRP**, *ENDMOD	Optional
ALWBLK	Allow blocking	***ALLREAD**, *NONE, *READ	Optional
DLYPRP	Delay PREPARE	***NO**, *YES	Optional
GENLVL	Severity level	0-40, **10**	Optional

Keyword	Description	Choices	Notes
MARGINS	Source margins	*Element list*	Optional
	Element 1: Left margin	1-32754, ***SRCFILE**	
	Element 2: Right margin	1-32754	
DATFMT	Date format	***JOB**, *USA, *ISO, *EUR, *JIS, *MDY, *DMY, *YMD, *JUL	Optional
DATSEP	Date separator character	***JOB**, '/', '.', ',', '-', ' ', *BLANK	Optional
TIMFMT	Time format	***HMS**, *USA, *ISO, *EUR, *JIS	Optional
TIMSEP	Time separator character	***JOB**, ':', '.', ',', ' ', *BLANK	Optional
RDBCNNMTH	RDB connect method	***DUW**, *RUW	Optional
DFTRDBCOL	Default collection	*Name*, ***NONE**	Optional
DYNDFTCOL	Dynamic default collection	***NO**, *YES	Optional
SQLPKG	Package	*Qualified object name*	Optional
	Qualifier 1: Package	*Name*, ***OBJ**	
	Qualifier 2: Library	*Name*, ***OBJLIB**	
SQLPATH	SQL path	Single values: ***NAMING**, *LIBL Other values (up to 268 repetitions): *Name*	Optional
SQLCURRULE	SQL rules	***DB2**, *STD	Optional
SAAFLAG	IBM SQL flagging	***NOFLAG**, *FLAG	Optional
FLAGSTD	ANS flagging	***NONE**, *ANS	Optional
PRTFILE	Print file	*Qualified object name*	Optional
	Qualifier 1: Print file	*Name*, **QSYSPRT**	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
DBGVIEW	Debugging view	***NONE**, *SOURCE	Optional
USRPRF	User profile	***NAMING**, *USER, *OWNER	Optional
DYNUSRPRF	Dynamic user profile	***USER**, *OWNER	Optional
SRTSEQ	Sort sequence	Single values: ***JOB**, *HEX, *JOBRUN, *LANGIDUNQ, *LANGIDSHR Other values: *Qualified object name*	Optional
	Qualifier 1: Sort sequence	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
LANGID	Language id	*Character value*, ***JOB**, *JOBRUN	Optional
DECRESULT	Decimal result options	*Element list*	Optional
	Element 1: Maximum precision	**31**, 63	
	Element 2: Maximum scale	0-63, **31**	
	Element 3: Minimum divide scale	0-9, **0**	
DECFLTRND	Decimal float rounding mode	***HALFEVEN**, *HALFUP, *DOWN, *CEILING, *FLOOR, *HALFDOWN, *UP	Optional

Top

Source file (SRCFILE)

Specifies the source file that contains the C++ source statements and SQL statements.

Qualifier 1: Source file

QCSRC

Source file QCSRC contains the C++ source.

name Specify the name of the source file that contains the C++ source.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library to be searched.

Top

Source member (SRCMBR)

Specifies the name of the source file member that contains the C++ source.

Top

Source stream file (SRCSTMF)

Specifies the path name to the file that contains the C++ source statements and SQL statements. The path name can be either absolute or relative.

Lines cannot exceed 32754 characters.

Top

To source file (TOSRCFILE)

Specifies the qualified name of the source file that is to contain the output C++ source member that has been processed by the SQL C++ precompiler. If the specified source file is not found, it will be created. If **Source member (SRCMBR)** parameter is specified, the output member will have the same name as the **Source member (SRCMBR)** parameter. If **Source stream file (SRCSTMF)** parameter is specified, the output member name will be generated from the first ten characters of the file name with the extension removed.

Qualifier 1: To source file

***CALC**
The output source file name will be generated based on the margins of the source file. The name will be QSQLTxxxxx, where xxxxx is the width of the source file. If the source file record length is less than or equal to 92, the name will be QSQLTEMP.

name Specify the name of the source file to contain the output source member.

Qualifier 2: Library

***LIBL** The job's library list is searched for the specified file. If the file is not found in any library in the library list, the file will be created in the current library.

***CURLIB**
The current library for the job will be used. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library that is to contain the output source file.

Top

Commitment control (COMMIT)

Specifies whether SQL statements in the compiled program are run under commitment control. Files referred to in the host language source are not affected by this option. Only SQL tables, SQL views, SQL packages, SQL sequences, SQL aliases, SQL Types, SQL procedures, SQL functions, SQL indexes, SQL schemas, SQL triggers, and SQL views referred to in SQL statements are affected.

***CHG** *or* ***UR**
Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs can be seen.

***CS** Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows updated, deleted, and inserted are locked until the end of the unit of work (transaction). A row that is selected, but not updated, is locked until the next row is selected. Uncommitted changes in other jobs cannot be seen.

***ALL** *or* ***RS**
Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows selected, updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs cannot be seen.

***NONE** *or* ***NC**
Specifies that commitment control is not used. Uncommitted changes in other jobs can be seen. If the SQL DROP SCHEMA statement is included in the program, *NONE or *NC must be used. If a relational database is specified for the **Relational database (RDB)** parameter, and the relational database is on a system that is not on a System i, *NONE or *NC cannot be specified.

***RR** Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows selected, updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs cannot be seen. All tables referred to in SELECT, UPDATE, DELETE, and INSERT statements are locked exclusively until the end of the unit of work (transaction).

Top

Relational database (RDB)

Specifies the name of the relational database where the SQL package is to be created.

***LOCAL**
The program is created as a distributed SQL program. The SQL statements will access the local database. An SQL package object is not created as part of the precompile process. The Create Structured Query Language Package (CRTSQLPKG) command can be used.

***NONE**
An SQL package object is not created. The program object is not a distributed program and the Create Structured Query Language Package (CRTSQLPKG) command cannot be used.

name Specify the name of the relational database where the new SQL package object is to be created. When the name of the local relational database is specified, the program created is still a distributed SQL program. The SQL statements will access the local database.

Top

Listing output (OUTPUT)

Specifies whether the precompiler listing is generated.

***NONE**
The precompiler listing is not generated.

***PRINT**
The precompiler listing is generated.

Top

Text 'description' (TEXT)

Specifies text that briefly describes the program and its function.

***SRCMBRTXT**
The text is taken from the source file member being used to create the program. If the source file is an inline file or a device file, the text is blank.

***BLANK**
No text is specified.

'description'
Specify no more than 50 characters, enclosed in apostrophes.

Top

RDB user (USER)

Specifies the user name sent to the remote system when starting the conversation. This parameter is valid only when **Relational database (RDB)** is specified.

***CURRENT**
The user name associated with the current job is used.

name Specify the user name to be used for the application server job.

Top

RDB user password (PASSWORD)

Specifies the password to be used on the remote system. This parameter is valid only when **Relational database (RDB)** is specified.

***NONE**
No password is sent. A user name cannot be specified for the **RDB user (USER)** parameter if this value is specified.

Note: Specifying a password of a blank is the same as specifying *NONE.

password
Specify the password of the user name specified for the **RDB user (USER)** parameter.

Top

Precompiler options (OPTION)

Specifies whether one or more of the following options are used when the C++ source is precompiled. If an option is specified more than once, or if two options conflict, the last option specified is used.

Cross reference options:

***XREF** The precompiler cross-references items in the program to the statement numbers in the program that refer to those items.

***NOXREF**
The precompiler does not cross-reference names.

Decimal point options:

***JOB** The representation for the decimal point specified for the job at precompile time is used.

Note: If QDECFMT specifies that the value used as the decimal point is a comma, any numeric constants in lists (such as in the SELECT clause and VALUES clause) must be separated by a comma followed by a blank. For example, VALUES(1,1, 2,23, 4,1) is equivalent to VALUES(1.1,2.23,4.1) where the decimal point is the period.

***SYSVAL**
The value used as the decimal point in numeric constants is from the QDECFMT system value. This value is also used as the decimal point character when casting a numeric value to character.

Note: If QDECFMT specifies that the value used as the decimal point is a comma; any numeric constants in lists (such as in the SELECT clause and VALUES clause) must be separated by a comma followed by a blank. For example, VALUES(1,1, 2,23, 4,1) is equivalent to VALUES(1.1,2.23,4.1) where the decimal point is the period.

***PERIOD**
The value used as the decimal point for numeric constants used in SQL statements is a period. This value is also used as the decimal point character when casting a numeric value to character.

***COMMA**
The value used as the decimal point in numeric constants is a comma. Any numeric constants in lists (such as in the SELECT clause and VALUES clause) must be separated by a comma followed by a blank. For example, VALUES(1,1, 2,23, 4,1) is equivalent to VALUES(1.1,2.23,4.1) where the decimal point is the period.

Naming convention options:

***SYS** Specifies that the system naming convention is used (library-name/file-name).

***SQL** Specifies that the SQL naming convention is used (schema-name.table-name).

If a relational database is specified for the **Relational database (RDB)** parameter, and the database is on a system that is not a System i, *SQL must be specified as the naming convention.

Second-level message text options:

***NOSECLVL**
Second-level text descriptions are not added to the listing.

***SECLVL**
Second-level text with replacement data is added for all messages on the listing.

NUL required options:

***NOCNULRQD**
For output character and graphic host variables, the NUL-terminator is not returned when the host variable is exactly the same length as the data. Input character and graphic host variables do not require a NUL-terminator.

***CNULRQD**
Output character and graphic host variables always contain the NUL-terminator. If there is not enough space for the NUL-terminator, the data is truncated and the NUL-terminator is added. Input character and graphic host variables require a NUL-terminator.

LOB optimization for DRDA options:

***OPTLOB**
The first FETCH for a cursor determines how the cursor will be used for LOBs (Large Objects) on all subsequent FETCHes. This option remains in effect until the cursor is closed.

If the first FETCH uses a LOB locator to access a LOB column, no subsequent FETCH for that cursor can fetch that LOB column into a LOB host variable.

If the first FETCH places the LOB column into a LOB host variable, no subsequent FETCH for that cursor can use a LOB locator for that column.

***NOOPTLOB**
There is no restriction on whether a column is retrieved into a LOB locator or into a LOB host variable. This option can cause performance to degrade.

Event file creation options:

***NOEVENTF**
The compiler will not produce an event file for use by CoOperative Development Environment (CODE).

***EVENTF**
The compiler produces an event file for use by CoOperative Development Environment (CODE). The event file will be created as a member in the file EVFEVENT in your object library. CODE uses this file to offer error feedback integrated with the CODE editor. This option is normally specified by CODE on your behalf.

Extended indicators options:

***NOEXTIND**
Extended indicator support is not enabled.

***EXTIND**
Extended indicator support is enabled.

Top

Target release (TGTRLS)

Specifies the release of the operating system on which you intend to use the object being created.

When specifying the **target-release** value, the format VxRxMx is used to specify the release, where Vx is the version, Rx is the release, and Mx is the modification level. For example, V5R3M0 is version 5, release 3, modification 0.

Valid values depend on the current version, release, and modification level of the operating system, and they change with each new release. You can press F4 while prompting this command parameter to see a list of valid target release values.

***CURRENT**
The object is to be used on the release of the operating system currently running on your system. The object can also be used on a system with any subsequent release of the operating system installed.

***PRV** The object is to be used on the previous release with modification level 0 of the operating system. The object can also be used on a system with any subsequent release of the operating system installed.

target-release
Specify the release in the format VxRxMx. The object can be used on a system with the specified release or with any subsequent release of the operating system installed.

Top

INCLUDE file (INCFILE)

Specifies the source file that contains members to be included in the program with the SQL INCLUDE statement.

Single values

***SRCFILE**
The qualified source file you specify for the **Source file (SRCFILE)** parameter contains the source file members specified on any SQL INCLUDE statements.

Qualifier 1: INCLUDE file

name Specify the name of the source file that contains the source file members specified on any SQL INCLUDE statements.

The record length of the source file you specify here must be no less than the record length of the source file you specify for the **Source file (SRCFILE)** parameter.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the source file is located.

Top

SQL INCLUDE directory (INCDIR)

Specifies the path name of the directory containing any files to be included in the program with the SQL INCLUDE statement.

Specifying a value for this parameter does not pass an INCDIR value from the SQL precompiler to the compiler. If you want to pass an INCDIR value through to the compiler, this can be done using the **Compiler options (COMPILEOPT)** parameter.

***NONE**
The current directory and the source directory will be searched.

path-name
The path name of the directory that contains the files specified on any SQL INCLUDE statement. The current directory will be searched first, then the specified path name, then the source directory.

Top

Allow copy of data (ALWCPYDTA)

Specifies whether a copy of the data can be used in a SELECT statement.

***OPTIMIZE**
The system determines whether to use the data retrieved directly from the database or to use a copy of the data. The decision is based on which method provides the best performance. If the **Commitment control (COMMIT)** parameter is not *NONE, the **Allow blocking (ALWBLK)** parameter should be set to *ALLREAD, when possible, for best performance.

***YES** A copy of the data is used only when necessary.

***NO** A copy of the data is not used. If a temporary copy of the data is required to perform the query, an error message is returned.

Top

Close SQL cursor (CLOSQLCSR)

Specifies when SQL cursors are implicitly closed, SQL prepared statements are implicitly discarded, and LOCK TABLE locks are released. SQL cursors are explicitly closed when the user issues the CLOSE, COMMIT, or ROLLBACK (without HOLD) SQL statements.

***ENDACTGRP**
SQL cursors are closed and SQL prepared statements are implicitly discarded, and LOCK TABLE locks are released when the activation group ends.

***ENDMOD**
SQL cursors are closed and SQL prepared statements are implicitly discarded when the module is exited. LOCK TABLE locks are released when the first SQL program on the call stack ends.

Top

Allow blocking (ALWBLK)

Specifies whether the database manager can use record blocking and the extent to which blocking can be used for read-only cursors.

***ALLREAD**
Rows are blocked for read-only cursors. All cursors in a program that are not explicitly able to be changed are opened for read-only processing even though there may be EXECUTE or EXECUTE IMMEDIATE statements in the program.

Specifying *ALLREAD:

- Allows record blocking for all read-only cursors.
- Can improve the performance of almost all read-only cursors in programs, but limits queries in the following ways:
  - The Rollback (ROLLBACK) command, a ROLLBACK statement in host languages, or the ROLLBACK HOLD SQL statement does not reposition a read-only cursor when *ALLREAD is specified.

– Dynamic running of a positioned UPDATE or DELETE statement (for example, using EXECUTE IMMEDIATE), can not be used to update a row in a cursor unless the DECLARE statement for the cursor includes the FOR UPDATE clause.

***NONE**

Rows are not blocked for retrieval of data for cursors.

Specifying *NONE:

- Guarantees that the data retrieved is current.
- May reduce the amount of time required to retrieve the first row of data for a query.
- Stops the database manager from retrieving a block of data rows that is not used by the program when only the first few rows of a query are retrieved before the query is closed.
- Can degrade the overall performance of a query that retrieves a large number of rows.

***READ**

Records are blocked for read-only retrieval of data for cursors when:

- *NONE is specified for the **Commitment control (COMMIT)** parameter, which indicates that commitment control is not used.
- The cursor is declared with a FOR READ ONLY clause or there are no dynamic statements that could run a positioned UPDATE or DELETE statement for the cursor.

Top

Delay PREPARE (DLYPRP)

Specifies whether the dynamic statement validation for a PREPARE statement is delayed until an OPEN, EXECUTE, or DESCRIBE statement is run. Delaying validation improves performance by eliminating redundant validation.

***NO** Dynamic statement validation is not delayed. When the dynamic statement is prepared, the access plan is validated. When the dynamic statement is used in an OPEN or EXECUTE statement, the access plan is revalidated. Because the authority or the existence of objects referred to by the dynamic statement may change, you must still check the SQLCODE or SQLSTATE after issuing the OPEN or EXECUTE statement to ensure that the dynamic statement is still valid.

***YES** Dynamic statement validation is delayed until the dynamic statement is used in an OPEN, EXECUTE, or DESCRIBE SQL statement. When the dynamic statement is used, the validation is completed and an access plan is built. If you specify *YES for this parameter for precompiled programs, you should check the SQLCODE and SQLSTATE after running an OPEN, EXECUTE, or DESCRIBE statement to ensure that the dynamic statement is valid.

Note: If you specify *YES, performance is not improved if the INTO clause is used on the PREPARE statement or if a DESCRIBE statement uses the dynamic statement before an OPEN is issued for the statement.

Top

Severity level (GENLVL)

Specifies the severity level at which the convert operation fails. If errors occur that have a severity level greater than this value, the operation ends.

10 The convert operation fails if SQL precompiler messages with a message severity greater than 10 are generated.

0-40 Specify the maximum SQL precompiler message severity level to be used to control whether the convert operation fails.

Top

Source margins (MARGINS)

Specifies the part of the precompiler input record that contains source text.

If the **Source stream file (SRCSTMF)** parameter is specified, margins are ignored.

Element 1: Left margin

***SRCFILE**
The left margin will be set to 1 and the right margin will be set to the record length of the input source file.

1-32754
Specify the beginning position to be used for each input record.

Element 2: Right margin

1-32754
Specify the ending position to be used for each input record.

Top

Date format (DATFMT)

Specifies the format used when accessing date result columns. All output date fields are returned in the format you specify. For input date strings, the value you specify is used to determine whether the date is a valid format.

Note: An input date string that uses the format *USA, *ISO, *EUR, or *JIS is always valid.

If you connect to a relational database that is on a system that is not a System i, the format must be *USA, *ISO, *EUR, or *JIS.

***JOB** The format specified for the job at precompile time or when a new interactive SQL session is created is used.

Use the Display Job (DSPJOB) command to determine the current date format for the job.

***USA** The United States date format **mm/dd/yyyy** is used.

***ISO** The International Organization for Standardization (ISO) date format **yyyy-mm-dd** is used.

***EUR** The European date format **dd.mm.yyyy** is used.

***JIS** The Japanese Industrial Standard date format **yyyy-mm-dd** is used.

***MDY** The date format **mm/dd/yy** is used.

***DMY** The date format **dd/mm/yy** is used.

***YMD** The date format **yy/mm/dd** is used.

***JUL** The Julian date format **yy/ddd** is used.

Top

Date separator character (DATSEP)

Specifies the separator to be used when accessing date result columns.

Note: This parameter applies only when *JOB, *MDY, *DMY, *YMD, or *JUL is specified for the **Date format (DATFMT)** parameter.

***JOB** The date separator specified for the job at precompile time, when a new interactive SQL session is created, or when Run SQL Statement (RUNSQLSTM) command is run.

Use the Display Job (DSPJOB) command to determine the current date separator value for the job.

'/' A slash is used as the date separator.

'.' A period is used as the date separator.

'-' A dash is used as the date separator.

',' A comma is used as the date separator.

' ' *or* *BLANK
A blank is used as the date separator.

Top

Time format (TIMFMT)

Specifies the format used when accessing time result columns. All output time fields are returned in the format you specify. For input time strings, the value you specify is used to determine whether the time is specified in a valid format.

Note: An input time string that uses the format *USA, *ISO, *EUR, or *JIS is always valid.

If you connect to a relational database that is on a system that is not another System i, the time format must be *USA, *ISO, *EUR, *JIS, or *HMS with a time separator of a colon or period.

***HMS** The **hh:mm:ss** format is used.

***USA** The United States time format **hh:mmxx** is used, where **xx** is AM or PM.

***ISO** The International Organization for Standardization (ISO) time format **hh.mm.ss** is used.

***EUR** The European time format **hh.mm.ss** is used.

***JIS** The Japanese Industrial Standard time format **hh:mm:ss** is used.

Top

Time separator character (TIMSEP)

Specifies the separator used when accessing time result columns.

Note: This parameter applies only when *HMS is specified for the **Time format (TIMFMT)** parameter.

***JOB** The time separator specified for the job at precompile time, when a new interactive SQL session is created, or when RUNSQLSTM is run is used.

Use the Display Job (DSPJOB) command to determine the current time separator value for the job.

':' A colon is used as the time separator.

'.' A period is used as the time separator.

',' A comma is used as the time separator.

' ' *or* ***BLANK**
A blank is used as the time separator.

Top

RDB connect method (RDBCNNMTH)

Specifies the semantics used for CONNECT statements.

***DUW**
CONNECT (Type 2) semantics are used to support distributed unit of work. Consecutive CONNECT statements to additional relational databases do not result in disconnection of previous connections.

***RUW** CONNECT (Type 1) semantics are used to support remote unit of work. Consecutive CONNECT statements result in the previous connection being disconnected before a new connection is established.

Top

Default collection (DFTRDBCOL)

Specifies the name of the schema identifier used for the unqualified names of the tables, views, indexes, SQL packages, aliases, constraints, external programs, node groups, and triggers. This parameter applies only to static SQL statements.

***NONE**
The naming convention specified for the **Precompiler options (OPTION)** parameter is used.

name Specify the name of the schema identifier to be used instead of the naming convention specified for the **Precompiler options (OPTION)** parameter.

Top

Dynamic default collection (DYNDFTCOL)

Specifies whether the default schema name specified for the **Default collection (DFTRDBCOL)** parameter is also used for dynamic statements.

***NO** Do not use the value specified for the **Default collection (DFTRDBCOL)** parameter for unqualified names of tables, views, indexes, SQL packages, aliases, constraints, external programs, node groups, and triggers for dynamic SQL statements. The naming convention specified for the **Precompiler options (OPTION)** parameter is used.

***YES** The schema name specified for the **Default collection (DFTRDBCOL)** parameter will be used for the unqualified names of the tables, views, indexes, SQL packages, aliases, constraints, external programs, node groups, and triggers in dynamic SQL statements.

Top

Package (SQLPKG)

Specifies the qualified name of the SQL package created on the relational database specified for the **Relational database (RDB)** parameter of this command.

Qualifier 1: Package

***OBJ** The name of the SQL package is the same as the object name specified for the **Object (OBJ)** parameter.

name Specify the name of the SQL package. If the remote system is not a System i, no more than 8 characters can be specified.

Qualifier 2: Library

***OBJLIB**
The package is created in the library with the same name as the library specified for the **Object (OBJ)** parameter.

name Specify the name of the library where the package is created.

Top

SQL path (SQLPATH)

Specifies the path to be used to find procedures, functions, and user defined types in static SQL statements.

***NAMING**
The path used depends on the naming convention specified for the **Precompiler options (OPTION)** parameter.

For *SYS naming, the path used is *LIBL, the current library list at runtime.

For *SQL naming, the path used is ″QSYS″, ″QSYS2″, ″userid″, where ″userid″ is the value of the USER special register. If a schema name is specified for the **Default collection (DFTRDBCOL)** parameter, the schema name takes the place of userid.

***LIBL** The path used is the library list at runtime.

name Specify one or more schema names. A maximum of 268 schema names may be specified.

Top

SQL rules (SQLCURRULE)

Specifies the semantics used for SQL statements.

***DB2** The semantics of all SQL statements will default to the rules established for DB2. The following semantics are controlled by this option:

Hexadecimal constants are treated as character data.

***STD** The semantics of all SQL statements will default to the rules established by the ISO and ANSI SQL standards. The following semantics are controlled by this option:

Hexadecimal constants are treated as binary data.

Top

IBM SQL flagging (SAAFLAG)

Specifies the IBM SQL flagging function. This parameter allows you to flag SQL statements to verify whether they conform to IBM SQL syntax.

***NOFLAG**
No checks are made to see whether SQL statements conform to IBM SQL syntax.

***FLAG**
Checks are made to see whether SQL statements conform to IBM SQL syntax.

Top

ANS flagging (FLAGSTD)

Specifies whether non-standard statements are flagged. This parameter allows you to flag SQL statements to verify whether they conform to the Core level of the ISO/IEC 9075-2003 standards.

***NONE**
No checks are made to see whether SQL statements conform to ANSI standards.

***ANS** Checks are made to see whether SQL statements conform to standards.

Top

Print file (PRTFILE)

Specifies the printer device file to be used for the precompiler output listing.

Qualifier 1: Print file

QSYSPRT
The precompiler output file is directed to the IBM-supplied printer file, QSYSPRT. The file QSYSPRT has a record length of 132. If you specify a file whose record length is less than 132, information is lost.

name Specify the name of the printer device file to which the precompiler output is directed.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library where the printer file is located.

Top

Debugging view (DBGVIEW)

Specifies the type of source debug information to be provided by the SQL precompiler.

***NONE**
The source view will not be generated.

***SOURCE**
The SQL precompiler will provide the source views for the root and if necessary, SQL INCLUDE statements. A view is to be provided which contains the statements generated by the precompiler.

Top

User profile (USRPRF)

Specifies the user profile that is used when the compiled program object and SQL package object is run, including the authority that the program object or SQL package has for each object in static SQL statements. The profile of either the owner or the user is used to control access to objects.

***NAMING**
The user profile is determined by the naming convention. If the naming convention is *SQL, USRPRF(*OWNER) is used. If the naming convention is *SYS, USRPRF(*USER) is used.

***USER**
The profile of the user running the program or SQL package is used.

***OWNER**
The user profiles of both the owner and the user are used when the program or SQL package is run.

Top

Dynamic user profile (DYNUSRPRF)

Specifies the user profile used for dynamic SQL statements.

***USER**
Local dynamic SQL statements are run under the profile of the program's user. Distributed dynamic SQL statements are run under the profile of the application server job.

***OWNER**
Local dynamic SQL statements are run under the profile of the program's owner. Distributed dynamic SQL statements are run under the profile of the SQL package's owner.

Top

Sort sequence (SRTSEQ)

Specifies the sort sequence table to be used for string comparisons in SQL statements.

Note: *HEX must be specified for this parameter on distributed applications where the application server is not on a System i.

Single values

***JOB** The SRTSEQ value for the job is used.

***JOBRUN**
The SRTSEQ value for the job is retrieved when the program is run. For distributed applications, SRTSEQ(*JOBRUN) is valid only when LANGID(*JOBRUN) is also specified.

***LANGIDUNQ**
The unique-weight sort table for the language specified for the **Language id (LANGID)** parameter is used.

***LANGIDSHR**
The shared-weight sort table for the language specified for the LANGID parameter is used.

***HEX** A sort sequence table is not used. The hexadecimal values of the characters are used to determine the sort sequence.

Qualifier 1: Sort sequence

name Specify the name of the sort sequence table to be used with this program.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library to be searched.

Top

Language id (LANGID)

Specifies the language identifier to be used when SRTSEQ(*LANGIDUNQ) or SRTSEQ(*LANGIDSHR) is specified.

***JOB** The LANGID value for the job is retrieved during the precompile.

***JOBRUN**
The LANGID value for the job is retrieved when the program is run. For distributed applications, LANGID(*JOBRUN) is valid only when SRTSEQ(*JOBRUN) is also specified.

language-id
Specify the language identifier to be used by the program.

Top

Decimal result options (DECRESULT)

Specifies the maximum precision, maximum scale and minimum divide scale that should be returned for result data types. The specified limit only applies to numeric (zoned) and decimal (packed) data types used in arithmetic expressions and in SQL column functions AVG and SUM.

Element 1: Maximum precision

31 The maximum precision (length) that should be returned for the result data types is 31 digits.

63 The maximum precision (length) that should be returned for the result data types is 63 digits.

Element 2: Maximum scale

31 The maximum scale (number of decimal positions to the right of the decimal point) that should be returned for the result data types is 31 digits.

0-63 Specify the maximum scale (number of decimal positions to the right of the decimal point) that should be returned for the result data types. The value can range from 0 to the maximum precision.

Element 3: Minimum divide scale

0 The minimum divide scale (number of decimal positions to the right of the decimal point) that should be returned for both intermediate and result data types is 0.

0-9 Specify the minimum divide scale (number of decimal positions to the right of the decimal point) that should be returned for both intermediate and result data types. The value cannot to exceed the maximum scale. If 0 is specified for the maximum scale, minimum divide scale is not used.

Top

Decimal float rounding mode (DECFLTRND)

Specifies the decimal floating point rounding mode used for static SQL statements.

***HALFEVEN**
Round to nearest; if equidistant, round so that the final digit is even. If the discarded digits represent greater than half (0.5) the value of a one in the next left position then the result coefficient should be incremented by 1 (rounded up). If they represent less than half, then the result coefficient is not adjusted (that is, the discarded digits are ignored). Otherwise (they represent exactly half) the result coefficient is unaltered if its rightmost digit is even, or incremented by 1 (rounded up) if its rightmost digit is odd (to make an even digit).

***HALFUP**
Round to nearest; if equidistant, round up. If the discarded digits represent greater than or equal to half (0.5) of the value of a one in the next left position then the result coefficient should be incremented by 1 (rounded up). Otherwise the discarded digits are ignored.

***DOWN**
Round towards 0 (truncation). The discarded digits are ignored.

***CEILING**
Round towards +infinity. If all of the discarded digits are zero or if the sign is negative the result is unchanged other than the removal of discarded digits. Otherwise, the result coefficient should be incremented by 1 (rounded up).

***FLOOR**
Round towards -infinity. If all of the discarded digits are zero or if the sign is positive the result is unchanged other than the removal of discarded digits. Otherwise, the sign is negative and the result coefficient should be incremented by 1.

***HALFDOWN**
Round to nearest; if equidistant, round down. If the discarded digits represent greater than half (0.5) of the value of a one in the next left position then the result coefficient should be incremented by 1 (rounded up). Otherwise (the discarded digits are 0.5 or less) the discarded digits are ignored.

***UP** Round away from 0. Of all of the discarded digits are zero the result is unchanged other than the removal of discarded digits. Otherwise, the result coefficient should be incremented by 1 (rounded up).

Top

Examples

```
CVTSQLCPP   SRCFILE(PAYLIB/QCSRC)  SRCMBR(PAYROLL)
            TOSRCFILE(PAYLIB/QSQLSRC)
```

This command runs the SQL precompiler which precompiles the C++ source in member PAYROLL of source file QCSRC in library PAYLIB, and stores the changed source in member PAYROLL of source file QSQLSRC in library PAYLIB.

Top

Error messages

***ESCAPE Messages**

SQL9001
SQL precompile failed.

SQL9004
Create of SQL package failed.

SQL9006
DB2 Query Mgr and SQL DevKit not at same install level as the operating system.

Top

Start DB2 UDB Query Manager (STRQM)

Where allowed to run: Interactive environments (*INTERACT *IPGM *IREXX *EXEC)
Threadsafe: No

The Start Query Manager (STRQM) command displays the DB2 UDB Query Manager main menu. From this menu, you can select options to work with Query Manager queries, report forms, tables, and profiles.

There are no parameters for this command.

Top

Parameters

None

Top

Examples

```
STRQM
```

This command displays the DB2 UDB Query Manager main menu. From this menu, you can select options to work with Query Manager queries, report forms, tables, and profiles.

Top

Error messages

***ESCAPE Messages**

QWM2701
&1 command failed.

QWM2703
&1 command ended.

QWM2707
*LIBL not allowed when SQL naming applied.

QWM2709
User or password not valid with relational database value.

QWM2710
Password value *NONE only valid with user value *CURRENT.

QWM2712
Character in user name not valid.

Top

Start SQL Interactive Session (STRSQL)

Where allowed to run: Interactive environments (*INTERACT *IPGM *IREXX *EXEC)
Threadsafe: No

Parameters
Examples
Error messages

The Start Structured Query Language (STRSQL) command starts the interactive SQL program. The program starts the statement entry of the interactive SQL program which immediately shows the Enter SQL Statements display. This display allows you to build, edit, enter, and run a SQL statement in an interactive environment. Any messages during the running of the program are shown on this display.

Top

Parameters

Keyword	Description	Choices	Notes
COMMIT	Commitment control	***NONE**, *CHG, *CS, *ALL, *RR, *NC, *UR, *RS	Optional, Positional 1
NAMING	Naming convention	***SYS**, *SQL	Optional, Positional 2
PROCESS	Statement processing	***RUN**, *VLD, *SYN	Optional, Positional 3
LIBOPT	Library option	*Name*, ***LIBL**, *USRLIBL, *ALLUSR, *ALL, *CURLIB	Optional, Positional 4
LISTTYPE	List type	***ALL**, *SQL	Optional, Positional 5
REFRESH	Data refresh	***ALWAYS**, *FORWARD	Optional, Positional 6
ALWCPYDTA	Allow copy data	***YES**, *OPTIMIZE, *NO	Optional
DATFMT	Date format	***JOB**, *USA, *ISO, *EUR, *JIS, *MDY, *DMY, *YMD, *JUL	Optional
DATSEP	Date separator character	***JOB**, '/', '.', ',', '-', ' ', *BLANK	Optional
TIMFMT	Time format	***HMS**, *USA, *ISO, *EUR, *JIS	Optional
TIMSEP	Time separator character	***JOB**, ':', '.', ',', ' ', *BLANK	Optional
DECPNT	Decimal point	***JOB**, *PERIOD, *COMMA, *SYSVAL	Optional
SRTSEQ	Sort sequence	Single values: *HEX, ***JOB**, *JOBRUN, *LANGIDUNQ, *LANGIDSHR Other values: *Qualified object name*	Optional
	Qualifier 1: Sort sequence	*Name*	
	Qualifier 2: Library	*Name*, ***LIBL**, *CURLIB	
LANGID	Language identifier	*Character value*, ***JOB**, *JOBRUN	Optional
PGMLNG	Program language	***NONE**, *C, *CBL, *PLI, *RPG, *FTN	Optional
SQLSTRDLM	SQL string delimiter	***QUOTESQL**, *APOSTSQL	Optional

Top

Commitment control (COMMIT)

Specifies whether SQL statements are run under commitment control.

***NONE** *or* ***NC**
Specifies that commitment control is not used. Uncommitted changes in other jobs can be seen. If the SQL DROP SCHEMA statement is included in the program, *NONE or *NC must be used.

***CHG** *or* ***UR**
Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs can be seen.

***CS** Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows updated, deleted, and inserted are locked until the end of the unit of work (transaction). A row that is selected, but not updated, is locked until the next row is selected. Uncommitted changes in other jobs cannot be seen.

***ALL** *or* ***RS**
Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows selected, updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs cannot be seen.

***RR** Specifies the objects referred to in SQL ALTER, CALL, COMMENT ON, CREATE, DROP, GRANT, LABEL ON, RENAME, and REVOKE statements and the rows selected, updated, deleted, and inserted are locked until the end of the unit of work (transaction). Uncommitted changes in other jobs cannot be seen. All tables referred to in SELECT, UPDATE, DELETE, and INSERT statements are locked exclusively until the end of the unit of work (transaction).

Top

Naming convention (NAMING)

Specifies the naming convention used for objects in SQL statements.

***SYS** The system naming convention (library-name/file-name) is used.

***SQL** The SQL naming convention (schema-name.table-name) is used.

Top

Statement processing (PROCESS)

Specifies what values are used to process the statements.

***RUN** The statements are syntax checked, data checked, and then run.

***VLD** The statements are syntax checked and data checked but not run.

***SYN** The statements are syntax checked only.

Top

Library option (LIBOPT)

Specifies the list of libraries that are shown when you request list support from the Enter SQL Statements display (F4, F16, F17, F18). If you do not have *EXECUTE authority for a specified library, that library is not shown.

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The job's current library is shown.

***USRLIBL**
Only the libraries in the user portion of the job's library list are shown.

***ALL** All of the libraries in the system, including QSYS, are shown.

***ALLUSR**
All user libraries are searched. All libraries with names that do not begin with the letter Q are searched except for the following:

```
#CGULIB     #DSULIB     #SEULIB
#COBLIB     #RPGLIB
#DFULIB     #SDALIB
```

Although the following Qxxx libraries are provided by IBM, they typically contain user data that changes frequently. Therefore, these libraries are considered ″user libraries″, and are also searched:

```
QDSNX       QRCL        QUSRSYS
QGPL        QS36F       QUSRVxRxMx
QGPL38      QUSER38
QPFRDATA    QUSRINFSKR
```

Note: A different library name, of the form QUSRVxRxMx, can be created by the user for each release. VxRxMx is the version, release, and modification level of the library.

name Specify the name of the library to be shown.

Top

List type (LISTTYPE)

Specifies what types of objects are displayed when you request list support from the Enter SQL Statements display (F16, F17, and F18) or from a prompt (F4) display.

***ALL** All objects are displayed.

***SQL** Only SQL created objects are displayed.

Top

Data refresh (REFRESH)

Specifies when the display select output data is refreshed. This parameter does not apply to any SQL SELECT statements that require the creation of a temporary result.

***ALWAYS**
The data is always refreshed. You cannot specify *ALWAYS for this parameter and *OPTIMIZE for the **Allow copy of data (ALWCPYDTA)** parameter.

***FORWARD**
The data is refreshed only as the user pages forward to the end of the data for the first time. After that, it is not refreshed.

Top

Allow copy of data (ALWCPYDTA)

Specifies whether a copy of the data can be used when running a SELECT statement.

***YES** A copy of the data can only be used, if necessary, to run a SELECT statement.

***OPTIMIZE**
The system chooses whether to use the data retrieved directly from the database or to use a copy of the data. The decision is based on which choice provides the best performance.

This value decreases the time required for the total query. Since the copy of the data must be made before returning the first row of the result table, the time to retrieve the first row may be increased.

***NO** A copy of the data is not allowed. If the clauses in the SELECT statement require a copy of the data, an error message is returned. If the SELECT statement runs successfully, current data was used.

Top

Date format (DATFMT)

Specifies the format used when accessing date result columns. All output date fields are returned in the format you specify. For input date strings, the value you specify is used to determine whether the date is a valid format.

Note: An input date string that uses the format *USA, *ISO, *EUR, or *JIS is always valid.

If you connect to a relational database that is on a system that is not a System i, the format must be *USA, *ISO, *EUR, or *JIS.

***JOB** The format specified for the job at precompile time or when a new interactive SQL session is created is used.

Use the Display Job (DSPJOB) command to determine the current date format for the job.

***USA** The United States date format **mm/dd/yyyy** is used.

***ISO** The International Organization for Standardization (ISO) date format **yyyy-mm-dd** is used.

***EUR** The European date format **dd.mm.yyyy** is used.

***JIS** The Japanese Industrial Standard date format **yyyy-mm-dd** is used.

***MDY** The date format **mm/dd/yy** is used.

***DMY** The date format **dd/mm/yy** is used.

***YMD** The date format **yy/mm/dd** is used.

***JUL** The Julian date format **yy/ddd** is used.

Top

Date separator character (DATSEP)

Specifies the separator to be used when accessing date result columns.

Note: This parameter applies only when *JOB, *MDY, *DMY, *YMD, or *JUL is specified for the **Date format (DATFMT)** parameter.

***JOB** The date separator specified for the job at precompile time, when a new interactive SQL session is created, or when Run SQL Statement (RUNSQLSTM) command is run.

Use the Display Job (DSPJOB) command to determine the current date separator value for the job.

'/' A slash is used as the date separator.

'.' A period is used as the date separator.

'-' A dash is used as the date separator.

',' A comma is used as the date separator.

' ' *or* *BLANK
A blank is used as the date separator.

Top

Time format (TIMFMT)

Specifies the format used when accessing time result columns. All output time fields are returned in the format you specify. For input time strings, the value you specify is used to determine whether the time is specified in a valid format.

Note: An input time string that uses the format *USA, *ISO, *EUR, or *JIS is always valid.

If you connect to a relational database that is on a system that is not another System i, the time format must be *USA, *ISO, *EUR, *JIS, or *HMS with a time separator of a colon or period.

***HMS** The **hh:mm:ss** format is used.

***USA** The United States time format **hh:mmxx** is used, where **xx** is AM or PM.

***ISO** The International Organization for Standardization (ISO) time format **hh.mm.ss** is used.

***EUR** The European time format **hh.mm.ss** is used.

***JIS** The Japanese Industrial Standard time format **hh:mm:ss** is used.

Top

Time separator character (TIMSEP)

Specifies the separator used when accessing time result columns.

Note: This parameter applies only when *HMS is specified for the **Time format (TIMFMT)** parameter.

***JOB** The time separator specified for the job at precompile time, when a new interactive SQL session is created, or when RUNSQLSTM is run is used.

Use the Display Job (DSPJOB) command to determine the current time separator value for the job.

':' A colon is used as the time separator.

'.' A period is used as the time separator.

',' A comma is used as the time separator.

' ' *or* *BLANK
A blank is used as the time separator.

Top

Decimal Point (DECMPT)

Specifies the decimal point character to be used for numeric constants in SQL statements. This value is also used as the decimal point character when casting between character and numeric values.

***JOB** The representation for the decimal point is the value used by the job running the statement.

***SYSVAL**
The QDECFMT system value is used as the decimal point character.

***PERIOD**
A period represents the decimal point.

***COMMA**
A comma represents the decimal point.

Top

Sort sequence (SRTSEQ)

Specifies the sort sequence table to be used for string comparisons in SQL statements on the Enter SQL Statements display.

Single values

***JOB** The SRTSEQ value is retrieved when the user starts interactive SQL.

***JOBRUN**
The SRTSEQ value for the job is retrieved each time the user starts interactive SQL.

***LANGIDUNQ**
A unique-weight sort table is used.

***LANGIDSHR**
A shared-weight sort table is used.

***HEX** A sort sequence table is not used, and the hexadecimal values of the characters are used to determine the sort sequence.

Qualifier 1: Sort sequence

name Specify the name of the sort sequence table to be used with the interactive SQL session.

Qualifier 2: Library

***LIBL** All libraries in the job's library list are searched until the first match is found.

***CURLIB**
The current library for the job is searched. If no library is specified as the current library for the job, the QGPL library is used.

name Specify the name of the library to be searched.

Top

Language identifier (LANGID)

Specifies the language identifier to be used when SRTSEQ(*LANGIDUNQ) or SRTSEQ(*LANGIDSHR) is specified.

***JOB** The LANGID value specified for the job is used when the user starts a new interactive SQL.

***JOBRUN**
The LANGID value for the job is retrieved each time the user starts interactive SQL.

language-ID
Specify the language identifier to be used.

Top

Program language (PGMLNG)

Specifies the program language syntax rules to use. To use this parameter, you must specify *SYN for the **Statement processing (PROCESS)** parameter.

***NONE**
You are not using a specific language's syntax check rules.

***C** You are checking syntax using the C language syntax rules.

***CBL** You are checking syntax using the COBOL language syntax rules.

***PLI** You are checking syntax using the PL/I language syntax rules.

***RPG** You are checking syntax using the RPG language syntax rules.

***FTN** You are checking syntax using the FORTRAN language syntax rules.

Top

SQL string delimiter (SQLSTRDLM)

Specifies the SQL string delimiter if *CBL is specified for the **Program language (PGMLNG)** parameter.

***QUOTESQL**
The SQL string delimiter is represented by the quotation mark (").

***APOSTSQL**
The SQL string delimiter is represented by the apostrophe (').

Top

Examples

```
STRSQL
```

This command starts the Interactive SQL program. The program starts the statement entry of the interactive SQL program which immediately shows the Enter SQL Statements display. This display allows you to build, edit, enter, and run a SQL statement in an interactive environment.

Top

Error messages

***ESCAPE Messages**

CPF9801
Object &2 in library &3 not found.

CPF9802
Not authorized to object &2 in &3.

CPF9810
Library &1 not found.

CPF9820
Not authorized to use library &1.

CPF9822
Not authorized to file &1 in library &2.

CPF9830
Cannot assign library &1.

SQL0113
Name &1 not allowed.

SQL6141
Cannot use interactive SQL now.

SQL6145
Interactive SQL session has exceeded system limit.

SQL6332
Authorization list &1 does not exist.

SQL6342
Sort sequence &1 is not valid.

SQL6343
Language identifier &1 is not valid.

SQL9006
DB2 Query Mgr and SQL DevKit not at same install level as the operating system.

SQL9012
DB2 Query Manager and SQL Development Kit for i5/OS not available.

Top

Appendix. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

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

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

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

IBM Corporation

Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

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

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, IBM License Agreement for Machine Code, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Programming interface information

This DB2 Query Manager and SQL Development Kit for i5/OS commands publication documents intended Programming Interfaces that allow the customer to write programs to obtain the services of IBM DB2 Query Manager and SQL Development Kit for i5/OS.

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

Advanced Function Printing
AFP
AS/400
CICS
COBOL/400
C/400
DataPropagator
DB2
Distributed Relational Database Architecture
Domino
DRDA
IBM
Infoprint
InfoWindow
i5/OS
iSeries
Integrated Language Environment
Lotus
LPDA
OfficeVision
Print Services Facility
RPG/400
System i
System x
SystemView
System/36
TCS
Tivoli
WebSphere
z/OS

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

Terms and conditions

Permissions for the use of these publications is granted subject to the following terms and conditions.

Personal Use: You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative works of these publications, or any portion thereof, without the express consent of IBM.

Commercial Use: You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED ″AS-IS″ AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

Code license and disclaimer information

IBM grants you a nonexclusive copyright license to use all programming code examples from which you can generate similar function tailored to your own specific needs.

SUBJECT TO ANY STATUTORY WARRANTIES WHICH CANNOT BE EXCLUDED, IBM, ITS PROGRAM DEVELOPERS AND SUPPLIERS MAKE NO WARRANTIES OR CONDITIONS EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, REGARDING THE PROGRAM OR TECHNICAL SUPPORT, IF ANY.

UNDER NO CIRCUMSTANCES IS IBM, ITS PROGRAM DEVELOPERS OR SUPPLIERS LIABLE FOR ANY OF THE FOLLOWING, EVEN IF INFORMED OF THEIR POSSIBILITY:

1. LOSS OF, OR DAMAGE TO, DATA;
2. DIRECT, SPECIAL, INCIDENTAL, OR INDIRECT DAMAGES, OR FOR ANY ECONOMIC CONSEQUENTIAL DAMAGES; OR
3. LOST PROFITS, BUSINESS, REVENUE, GOODWILL, OR ANTICIPATED SAVINGS.

SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF DIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, SO SOME OR ALL OF THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

Printed in USA