

# New Features Summary

## Sybase® IQ 15.1

Document ID: DC01087-01-1510-01

Last revised: July 2009

Topic	Page
Scalar and aggregate external C/C++ user-defined functions	1
Time series and forecasting functions	3
Enhanced scalar date and time functions	5
New aggregate functions	5
Integration with Sybase WorkSpace, Sybase PowerDesigner and Sybase ETL	5
New licensable option	7
FIPS supported on Windows x64 [CR 564764]	7
Behavior changes	8
File and directory name changes	8
IQ Small Business Edition licensed by number of sockets	8
IQ Small Business Edition: maximum IP connection limit increased	8
IQ Enterprise Edition: VLDB license changes	8

## Scalar and aggregate external C/C++ user-defined functions

Sybase IQ now supports high-performance user-defined functions (UDFs). The UDFs are written in C or C++ code adhering to the interfaces described in the *User-Defined Functions Guide*. The UDF definitions must be compiled and linked into a shareable library which can then be loaded into a running Sybase IQ server, allowing the UDFs to be used in queries or other SQL statements.

Scalar and aggregate UDFs are only available with the In-database Analytics-Partner Solutions option. Installing the IQ\_UDF license enables the scalar and aggregate UDFs.

---

**Note** Sybase supports the scalar and aggregate UDF infrastructure only when:

- Used with the Sybase-provided time series and forecasting functions (see “Time series and forecasting functions” on page 3), or
- Used with other Sybase-certified third party libraries, or
- Used in functions written by customers who have purchased RAP – The Trading Edition™ Enterprise.

The use of external C/C++ UDFs requires the IQ\_UDF license.

---

Sybase IQ 15.1 includes two types of UDFs—scalar and aggregate:

**Scalar** Sybase IQ supports simple scalar UDFs that can be used anywhere the SQRT function can be used. These scalar UDFs can be deterministic, which means that for a given set of argument values, the function always returns the same result value, or they can be nondeterministic scalar functions, which means that the same arguments can return different results.

**Aggregate** Sybase IQ supports external C/C++ user-defined aggregate functions (UDAFs). The SUM function is an example of a built-in aggregate function. You can write UDAFs that can be used anywhere the SUM aggregate can be used. A simple aggregate function produces a single result value from a set of argument values. Many UDAFs can also be used as OLAP-style aggregates with an OVER clause that allows you to specify a static or moving window, within which Sybase IQ computes the aggregate. See Chapter 2, “Using OLAP,” in *System Administration Guide: Volume 2*.

Note that these new external C/C++ UDFs are different than the Interactive SQL UDFs, introduced in earlier versions of Sybase IQ. Interactive SQL UDFs are unchanged in version 15.1, and do not require a special license.

Documentation on creating UDFs using Interactive SQL remains in its original location: Chapter 1, “Using Procedures and Batches,” in *System Administration Guide: Volume 2*.

For more information on scalar and aggregate UDFs, see the *User-Defined Functions Guide*.

## Time series and forecasting functions

Sybase IQ 15.1 includes new functions for financial time series and forecasting analysis. These functions are available only with RAP – The Trading Edition Enterprise.

The new SQL functions call two third-party integrated libraries. The IMSL C Stat and C Math libraries, provided by Visual Numerics, contain C functions for time series and forecasting. Sybase IQ 15.1 automatically loads the IMSL libraries as needed when you call a valid function for time series and forecasting analysis. See “Loading the IMSL libraries for time series and forecasting functions” in Chapter 4, “SQL Functions,” in *Reference: Building Blocks, Tables, and Procedures*.

For information on how Sybase IQ handles and logs errors if a runtime error occurs when invoking IMSL library functions, see “IMSL library time series function error-handling” in Chapter 4, “SQL Functions,” in *Reference: Building Blocks, Tables, and Procedures*.

The new time series and forecasting SQL functions are:

**TS\_ARMA\_AR** Calculates the least-square estimates of parameters for an autoregressive moving average (ARMA) model and returns the requested autoregressive estimate.

**TS\_ARMA\_CONST** Calculates the least-square estimates of parameters for an autoregressive moving average (ARMA) model and returns an estimated constant.

**TS\_ARMA\_MA** Calculates the least-square estimates of parameters for an autoregressive moving average (ARMA) model, and returns the requested moving average estimate.

**TS\_AUTOCORRELATION** Calculates the sample autocorrelation function of a stationary time series.

**TS\_AUTO\_UNI\_AR** Performs automatic selection and fitting of a univariate autoregressive time series model.

**TS\_BOX\_COX\_XFORM** Performs a forward or inverse Box-Cox power transformation.

**TS\_DIFFERENCE** Differences a seasonal or nonseasonal time series.

**TS\_ESTIMATE\_MISSING** Estimates the missing values in a time series and returns them as a new time series interspersed with the original time series.

**TS\_LACK\_OF\_FIT** Performs a lack-of-fit test for a univariate time series or transfer function given the appropriate correlation function.

**TS\_LACK\_OF\_FIT\_P** Same as **TS\_LACK\_OF\_FIT** except that it returns the p-value of  $q$ , rather than returning  $q$ .

**TS\_MAX\_ARMA\_AR** Calculates the exact maximum likelihood estimation of the parameters in a univariate ARMA (autoregressive moving average) time series model, and returns the requested autoregressive estimate.

**TS\_MAX\_ARMA\_CONST** Calculates the exact maximum likelihood estimation of the parameters in a univariate ARMA time series model, and returns the constant estimate.

**TS\_MAX\_ARMA\_LIKELIHOOD** Calculates the exact maximum likelihood estimation of the parameters in a univariate ARMA time series model, and returns likelihood value ( $\ln$ ) for the fitted model.

**TS\_MAX\_ARMA\_MA** Calculates the exact maximum likelihood estimation of the parameters in a univariate ARMA time series model, and returns the requested moving average estimate.

**TS\_OUTLIER\_IDENTIFICATION** Detects and determines outliers and simultaneously estimates the model parameters in a time series where the underlying outlier-free series follows a general seasonal or nonseasonal ARMA model.

**TS\_PARTIAL\_AUTOCORRELATION** Calculates the sample partial autocorrelation function of a stationary time series.

**TS\_VWAP** Calculates the ratio of the value traded to total volume traded over a particular time horizon. VWAP (Volume-Weighted Average Price) is a measure of the average price of a stock over a defined trading horizon.

---

**Note** The **TS\_VWAP** function is used for financial time series calculations, but it does not load the IMSL libraries.

---

For detailed information on each of the above functions, including the mapping between a SQL function and its corresponding IMSL C function, see “Alphabetical list of functions” in Chapter 4, “SQL Functions,” in *Reference: Building Blocks, Tables, and Procedures*.

For reference information on the IMSL C functions, see *IMSL Numerical Library Users Guide: Volume 2 of 2 C Stat Library*.

## Enhanced scalar date and time functions

Three new SQL scalar functions for date and time calculations have been added to support time series and forecasting analysis. These new functions do not require a separate license.

**DATECEILING** Calculates a new date, time, or datetime value by increasing the provided value up to the nearest larger value of the specified multiple with the specified granularity.

**DATEFLOOR** Calculates a new date, time, or datetime value by reducing the provided value down to the nearest lower value of the specified multiple with the specified granularity.

**DATEROUND** Calculates a new date, time, or datetime value by rounding the provided value up or down to the nearest multiple of the specified value with the specified granularity.

For detailed information on each of the functions listed above, see “Alphabetical list of functions” in Chapter 4, “SQL Functions,” in *Reference: Building Blocks, Tables, and Procedures*.

## New aggregate functions

New aggregate functions have been added to support analysis of large amounts of time series data. These new functions provide correlation, linear regression, ranking, and other analysis capabilities. The new functions do not require a separate license. Most of these new aggregates can be used either as simple aggregate functions, or as OLAP-style aggregates. See Chapter 4, “SQL Functions,” in *Reference: Building Blocks, Tables, and Procedures*.

**CORR** Returns the correlation coefficient of a set of number pairs.

**COVAR\_POP** Returns the population covariance of a set of number pairs.

**COVAR\_SAMP** Returns the sample covariance of a set of number pairs.

**CUME\_DIST** Calculates the relative position of one value among a group of rows. It returns a decimal value between 0 and 1.

**EXP\_WEIGHTED\_AVG** Calculates an exponential weighted moving average. Weightings determine the relative importance of each quantity composing the average.

**FIRST\_VALUE** Returns the first value from a set of values.

**LAST\_VALUE** Returns the last value from a set of values.

**MEDIAN** Returns the median of an expression.

**REGR\_AVGX** Computes the average of the independent variable of the regression line.

**REGR\_AVGY** Computes the average of the dependent variable of the regression line.

**REGR\_COUNT** Returns an integer that represents the number of non-NULL number pairs used to fit the regression line.

**REGR\_INTERCEPT** Computes the y-intercept of the linear regression line that best fits the dependent and independent variables.

**REGR\_R2** Computes the coefficient of determination (also referred to as R-squared or the goodness-of-fit statistic) for the regression line.

**REGR\_SLOPE** Computes the slope of the linear regression line fitted to non-NULL pairs.

**REGR\_SXX** Returns the sum of squares of the independent expressions used in a linear regression model. The REGR\_SXX function can be used to evaluate the statistical validity of a regression model.

**REGR\_SXY** Returns the sum of products of the dependent and independent variables. The REGR\_SXY function can be used to evaluate the statistical validity of a regression model.

**REGR\_SYY** Returns values that can evaluate the statistical validity of a regression model.

**WEIGHTED\_AVG** Calculates an arithmetically (or linearly) weighted average. A weighted average is an average in which each quantity to be averaged is assigned a weight. Weightings determine the relative importance of each quantity composing the average.

For detailed information on each of the functions listed above, see “Alphabetical list of functions” in Chapter 4, “SQL Functions,” in *Reference: Building Blocks, Tables, and Procedures*.

For information on these new functions in the context of OLAP window functions, see “Statistical aggregate functions” in Chapter 2, “Using OLAP,” in *System Administration Guide: Volume 2*.

## Integration with Sybase WorkSpace, Sybase PowerDesigner and Sybase ETL

You can integrate Sybase IQ 15.1 can be integrated with supporting Sybase products to build and deploy an analytics infrastructure—an end-to-end information system that delivers business intelligence, advanced analytics, data mining, predictive modeling, reporting, and ETL (extract, transform, and load) capabilities. The supporting products are:

- Sybase WorkSpace provides the enterprise modeling component.
- Sybase PowerDesigner provides the database design and modeling component.
- Sybase ETL provides the extract, transform, and load component.

For more information on these supporting products, including where to find WorkSpace, PowerDesigner, and ETL documentation, see “Supporting Sybase products,” in Chapter 1, “Introducing Sybase IQ,” in *Introduction to Sybase IQ*.

## New licensable option

Sybase IQ 15.1 includes a new licensable option:

**In-database Analytics—Partner Solutions option** This option enhances Sybase IQ with a scalar and aggregate UDF infrastructure. This option includes the IQ\_UDF license (both scalar and aggregate UDFs require the license). See “Scalar and aggregate external C/C++ user-defined functions” on page 1. The In-database Analytics—Partner Solutions option is sold and supported only if it is deployed with partner-certified external libraries.

## FIPS supported on Windows x64 [CR 564764]

Sybase IQ 15.1 supports Federal Information Processing Standards (FIPS) encryption technology on Microsoft Windows for 64-bit systems. FIPS is also supported on all other platforms supported by Sybase IQ.

## Behavior changes

This section describes changes in Sybase IQ 15.1 that are not feature-related.

### File and directory name changes

Table 1 lists files and directories that have been renamed in Sybase IQ 15.1:

**Table 1: File and directory name changes**

15.0 name	15.1 name
IQ-15_0.sh	IQ-15_1.sh (or .csh)
IQ-15_0/	IQ-15_1/
iqsrv15.exe	iqsrv15 or iqsrv15.exe on Windows
IQAgent15.jar	IQAgent1510.jar
IQHelpen15.jar	IQHelpen1510.jar
IQPlugin15.jar	IQPlugin1510.jar

### IQ Small Business Edition licensed by number of sockets

In earlier versions of Sybase IQ, Small Business Edition (SE) was licensed by the number of cores used by the IQ server. In Sybase IQ 15.1 SE, licensing is based on the number of sockets—not cores—used by the IQ server.

### IQ Small Business Edition: maximum IP connection limit increased

In Sybase IQ 15.1 SE, the maximum IP connection limit has been increased from 10 connections to 25 connections.

### IQ Enterprise Edition: VLDB license changes

Each VLDB license now allows for 1TB of storage in the Sybase IQ main store. You will need to purchase one license for each additional TB of storage added to the main store.