INTERNAL
SAP HANA Client 2.4
Document Version: 1.1 – 2023-06-14

What's New in the SAP HANA Client

THE BEST RUN SAP

Content

1 New and Changed Features in the SAP HANA Client

> **i** Note
>
> Information about earlier versions of the SAP HANA Client (prior to version 2.4) is included in the SAP HANA Platform documentation.

Client Version	Type	Description	
2.17	New	**User-switching Node.js Connection Pools**	Explicit connection pooling allows you to create pools that support different users, making it different than implicit connection pooling.. The new `allowSwitchUser` parameter determines whether or not the pooled connection will allow the pool to contain connections from more than one user. For more information see: Explicit Connection Pooling API createPool({String | Object}[, Object]) Method getConnection([uid], [pwd], [Function]) Method
2.17	New	**.NET Entity Framework Core 7 Support**	The .NET Entity Framework Core version 7 driver supports columns that contain JSON documents. Entity Framework Core Support (SAP HANA Data Provider for Microsoft ADO.NET)

Client Version	Type	Description	
2.17	New	**LOB Streaming for INSERT, UPDATE, DELETE, and SELECT Statements**	The Go driver supports LOB streaming from statements that are prepared from a sql.Db* or sql.Tx* object. This allows for the following when sending data to or retrieving data from LOB columns: • More control over the driver's internal memory usage • Greater flexibility in sources of data for input/output other than directly in the application's memory The `lobChunkSize` connection property specifies the chunk size of the data that is retrieved and sent by the driver. LOB Streaming for INSERT, UPDATE, DELETE, and SELECT Statements Go Connection Properties
2.16	New	**Node.js TypeScript Support**	Use the Node.js driver with TypeScript to support optional static typing. Node.js TypeScript Support
2.16	New	**.NET Entity Framework Core Support**	SAP HANA supports .NET Entity Framework Core so that you can develop applications. Entity Framework Core Support (SAP HANA Data Provider for Microsoft ADO.NET) Example: Use the SAP HANA Entity Framework Core Database Provider

Client Version	Type	Description	
2.16	New	**Node.js queryTimeout Option for exec[ute](String[, Array][,Object][, Function]) Method**	The queryTimeout option has been added to the Node.js exec[ute](String[, Array][,Object][, Function]) Method. When set to true, it ensures that the server will time out within a reasonable period of time when a connection is queried. exec[ute](String[, Array][,Object][, Function]) Method
2.16	New	**SQLDBC TIMING Trace Category**	The new TIMING trace category summarizes the trace time for each SQL statement SQLDBC Trace Options
2.16	New	**Python setcommandinfo(command-info, line-number) Method**	Sets an arbitrary string and integer as command information to be sent to the database along with the next command, `execute`. Cursor Class setcommandinfo(command-info, line-number) Method
2.16	New	**The Go Driver Supports Obtaining Specific Ping() Errors**	The GO Driver now supports the retrieval of SAP HANA protocol-specific error messages sql.DB.PingContext() and sql.Conn.PingContext() by providing a driver-specific context.Context object. Obtaining Specific Ping() Errors

Client Version	Type	Description
2.16	New	**Upserting Spatial Data in WKT Format** All SQLDBC-based SAP HANA client components now support directly upserting spatial data in WKT format to SAP HANA without using the ST_GeomFromWKT conversion function in the UPSERT statement. The spatial data can be given in UTF8, CESU8, UCS2, or UCS4 format. However, in order to use this feature, the connection property `spatialTypes` must be set to 1 through the connection string for applications that use the SAP HANA Node.js, Python, golong, or Ruby driver. For ODBC applications, setting `spatialTypes` to 1 is not required, but the application should not set this property to 0 through the connection string or through SQLSetConnectAttr with the attribute of SQL_SQLDBC_SPATIALTYPES. SQLDBC Library
2.16	New	**SQLDBC Connection Properties** The new `allowReconnectOnSelect` connection property allows reconnects to happen for SELECT queries that fail during a receive. SQLDBC Connection Properties Reconnecting to the Server

Client Version	Type	Description
2.16	Changed	**SQLDBC Supports ON/OFF Boolean Property Settings** SQLDBC and SQLDBC-based client components now support ON/OFF Boolean property settings. Previous versions of the SAP HANA client have treated the ON/OFF settings of Boolean properties as a FALSE setting. SQLDBC Connection Properties
2.15	New	**macOS ARM 64-bit Platform Support** All drivers (except Ruby) now support macOS ARM 64-bit. The Python driver has also been posted to pypi.org . Available Client Interfaces on UNIX, macOS, or Linux
2.15	New	**Node.js Connection Properties** The new `passphrase` property specifies the alias for `sslKeyStorePassword`. Node.js Connection Properties
2.15	New	**SQLDBC Connection Properties** The `authenticationX509Password` specifies the PIN for the PSE or the password for the encrypted key, when used with CommonCryptoLib. When used with OpenSSL, it specifies the password for the encrypted key. The `sslKeyStorePassword` specifies the PIN for the keystore PSE or the password for the encrypted key, when used with CommonCryptoLib. When used with OpenSSL, it specifies the password for the encrypted key. SQLDBC Connection Properties

Client Version	Type	Description	
2.15	Changed	**hdbuserstore Commands**	The -u <USER> option is only for Microsoft Windows administrators. The -H <HOST> option is only for non-Microsoft Windows users. The COMPACT command allows you to permanently delete entries already marked as deleted. hdbuserstore Commands
2.14	New	**Node.js Explicit Connection Pooling API**	Create multiple customized connection pools and obtain the connection from the pool directly. Node.js Connection Pooling Explicit Connection Pooling API ConnectionPool Class Hana Class
2.14	New	**Linux ARM 64-bit Platform Support**	Except for Ruby, all SAP HANA client drivers are now available for Linux ARM 64-bit architectures. The install format is provided as .zip on SAP Software Downloads. Supported Platforms Available Client Interfaces on UNIX, macOS, or Linux
2.14	New	**macOS ARM 64-bit Platform Support**	The Node.js driver now supports macOS ARM 64-bit. Supported Platforms Available Client Interfaces on UNIX, macOS, or Linux

Client Version	Type	Description	
2.14	New	**macOS Signed and Notarized DMG Client Install**	The macOS Client install is now provided in DMG format and is signed and notarized by SAP.

Client Version	Type	Description	
2.14	Changed	**Network Compression Now Compresses Same-Machine Connections**	When network compression is enabled using the client `compress` connection property or server `indexserver.ini` > *session* > *compression* configuration, all SQLDBC and JDBC same-machine connections are compressed. Previously, same-machine connections were not compressed if `compression` was enabled. This change has been made to handle connections which use a proxy, where the end point of the proxy connection was on the same machine as the server. **i Note** If there is no proxy in use, then compressing same-machine connections can cause reduced performance. In this case, the new `allowLocalCompress`=FALSE SQLDBC and JDBC connection property can be used to restore the old behaviour of not compressing same-machine connections. Network Compression Configure Network Compression SQLDBC Connection Properties JDBC Connection Properties

Client Version	Type	Description	
2.14	New	**Node.js spatialTypes Connection Property**	Determines the native type returned by Statement.getColumnInfo and Statement.getParameterInfo for ST_POINT and ST_GEOMETRY columns or parameters. Node.js Connection Properties
2.14	New	**Node.js getOpenConnectionCount() Method**	Returns the current number of open connections. getOpenConnectionCount() Method
2.13	New	**SAP HANA HDBSQL Supports Positional Input Parameters**	SAP HANA HDBSQL supports the use of positional input parameters in interactive mode when prepared statements are toggled on. Run Commands
2.13	New	**X.509 Authentication in the Secure Store**	The connection property authenticationX509 can now be stored as part of a user key in the secure store. SQLDBC-based drivers using such the key now performs X.509 authentication. hdbuserstore Commands
2.13	New	**Go Driver Bulk Fetching**	The Go driver supports fetching a specified number of rows at a time from a rows object. Perform Bulk Fetching

Client Version	Type	Description
2.13	New	**Python Returns Information on Rows Affected and Errors** The Python driver provides the parameter `batcherrors` for the cursor method executemany(). The new cursor method getrowsaffectedcounts() returns a tuple containing the rows affected for each row that was processed by the executemany() call. Cursor Class
2.12	New	**JDBC Unblocks Server Garbage Collection** The JDBC driver now supports the `respectAutoCommitOnPrepare` connection property which unblocks server garbage collection when `autocommit` is set to TRUE. The default value is FALSE. JDBC Connection Properties
2.12	New	**JDBC Parameter Support** The JDBC driver now supports prepared and callable statements with more than 32767 parameters.
2.12	New	**JDBC Tracing Failures are Now Reported** The JDBC driver can now report failures that occur while creating, opening, or writing to the trace file. JDBC Tracing and Trace Options
2.11	New	**Node.js Adds Promise Support** A new Node.js module exposes Promise-based versions of all asynchronous driver methods. Promise Module

Client Version	Type	Description	
2.11	New	**Callback for ODBC Connection-level Tracing**	ODBC applications provide a callback to receive trace messages for a particular connection. Callback for ODBC Connection-level Tracing ODBC Connection Attributes
2.11	New	**Backoff Timer for JDBC and SQLDBC Drivers**	SQLDBC and JDBC drivers introduce a backoff timer for regular statement routing to minimize the effect of internal connection failures on routed statement executions. Backoff Timer for Regular Statement Routing (SAP HANA Platform) JDBC Connection Properties SQLDBC Connection Properties
2.11	New	**SQLDBC Better Detects Dropped Connections**	SQLDBC is now able to do silent reconnects in more cases when connecting to HANA Cloud or when there is a proxy between the client and the server. SQLDBC Connection Properties

Client Version	Type	Description	
2.11	New	**Removals from Client Installation Packages**	The ODBO (MDX) driver was removed from the HANA Client Windows x86 and x64 installations. The ODBO driver is still available as SAP HANA CLIENT FOR EXCEL 2.0 on SAP Software Downloads. In addition, the Microsoft Visual C++ 2010 SP1 runtime installation packages, `msdev2010_vcredist_x86.exe` and `msdev2010_vcredist_x64.exe`, were also removed from the HANA Client install.
2.10	New	**Python Driver Returns Byte Sequence in UTF-16 Format**	The Python driver now returns a byte sequence of the unconvertible bytes in UTF-16 format. SAP HANA can store partial UTF-16 sequences in NVARCHAR fields.
2.10	New	**Tutorial on How to Connect Using the SAP HANA Go Interface on Windows**	The tutorial shows you how to install Go and to create a Go application that queries an SAP HANA database on Windows. Connect Using the SAP HANA Go Interface on Windows Set Up Your Application to Use the Go Driver Package

Client Version	Type	Description	
2.10	Changed	**Updates to Go Driver Prerequisites**	There are two updates to the prerequisites to build the SAP HANA driver for Go: • On 64-bit Microsoft Windows, you must have a gcc compiler installed and the path to it set in your PATH environment variable. • The Go driver supports versions 1.8–1.16. Set Up Your Application to Use the Go Driver Package 3006307
2.10	New	**Node.js Maximum Result Set Array Limit**	The new function `resultSetArrayLimitMB` allows an application to limit the maximum amount of memory used for result set arrays returned by Connection.exec[ute] and Statement.exec[ute]. Node.js Connection Properties
2.10	New	**Node.js Driver Supports Network Packet Size Per Statement**	The Node.js driver provides two new methods to set and get the network packet size for a statement. • setPacketSize(Integer) Method • getPacketSize() Method Statement Class

Client Version	Type	Description
2.10	New	**Node.js Global Connection Pool Limit** The new function `setConnectionPoolLimit(Integer)` defines the maximum number of connections allowed in the connection pool on a global basis. Node.js Connection Properties Node.js Connection Pooling
2.10	New	**Python Driver Supports Network Packet Size Per Cursor** The Python driver provides two new methods on the cursor object to set and get the network packet size per cursor. • setPacketSize(value) Method • getPacketSize() Method Cursor Class
2.10	New	**SQLDBC Driver Supports New Set/Get Transaction DDL API** The SQLDBC driver supports a new API that allows setting transaction DDL to keep the driver in the correct state during reconnect.
2.10	New	**SAP HANA HDBSQL Suppress and Switch On/Off Printing of Rows and Execution Time** SAP HANA HDBSQL provides two options to switch on/off printing the number of rows affected and both overall and server time taken when a SQL command is executed. • *-nocmdstats* • *\cmdstats [ON|OFF]* SAP HANA HDBSQL Options

Client Version	Type	Description	
2.10	New	**SAP HANA HDBSQL Print Overall and Server Execution Time**	SAP HANA HDBSQL provides two options to print the overall and server time taken by SQL commands executed in batch mode using option *-I*. • *-timePerBatch* • *-totalBatchTime* SAP HANA HDBSQL Options
2.10	New	**JDBC Support for java.time Classes**	The JDBC driver now supports java.time classes introduced in JDK 8.
2.10	New	**JDBC Improved Secondary Session Fallback**	The JDBC driver now supports fallback to the anchor connection when a secondary connection is added to a write transaction.
2.9	New	**SQLDBC and JDBC Connections (SAP HANA Cloud)**	Enhanced support for SAP HANA Cloud redirection to an appropriate internal database server. This improves resource use and query latency.
2.9	New	**Node.js Trace Callback**	Provides a callback to receive trace messages. onTrace(traceOptions, callback) Method
2.9	New	**ODBC Statement Attribute**	The HANA ODBC driver provides a new statement attribute, `SQL_ATTR_STMT_PACKET_SIZE`, to set the network packet size for a statement. ODBC Statement Attributes

Client Version	Type	Description
2.9	New	**JDBC Driver Extensions to Set/Get Packet Size** The HANA JDBC driver provides two new driver-spe get a packet size per statement: • `com.sap.db.jdbc.Driver.setPacke statement, int packetSize )` • `com.sap.db.jdbc.Driver.getPacke statement )` getPacketSize(java.sql.Statement) Method setPacketSize(java.sql.Statement, int) Method
2.9	New	**golang Driver Extension for Bulk Inserts/ Updates/ Deletes** The golang driver supports bulk inserts, updates, and deletes via prepared statements. Perform Bulk INSERT, UPDATE, and DELETE Statements
2.9	New	**Check Node.js Connection Liveness** The Node.js driver allows you to specify that connections in the connection pool should be tested for viability before being reused via the `poolingCheck=true` connection property. Node.js Connection Pooling
2.9	New	**SAP HANA Explorer Microsoft Visual Studio 2017 Plug-in Has Been Removed** The `HanaExplorer.vsix` has been removed from the HANA Client installation.

Client Version	Type	Description	
2.9	New	**Remove Node.js Platform Binaries During npm Install Phase**	Specify which binaries should be kept during installation using the environment variable `HDB_NODEJS_INSTALL_PLATFORMS`. Environment Variables
2.9	New	**HDBCLI Python Driver Supports Explicitly Prepared Statements**	Three new Cursor methods allow for the separation of prepares and executes. These allow for the explicit creation of prepared statement handles without immediate execution. • prepare(operation[, newcursor=False]) Method • executeprepared([parameters]) Method • executemanyprepared(list_of_parameters) Method Cursor Class
2.8	New	**Windows Client Installation Includes ADO.NET Examples**	The .NET Framework examples are now included with all Microsoft Windows client installations in the `examples\ado.net` directory.
2.8	New	**SQLDBC Connection Properties**	The `statementRoutingWarnings` property returns a warning that gives information about the routing failure of a prepared statement. SQLDBC Connection Properties

Client Version	Type	Description	
2.7	New	**SQLTransactionRollbackException Thrown for SQL States 129-133**	The SAP HANA JDBC driver now throws a `SQLTransactionRollbackException` when an error with SQL State 129-133 is received from the server. Previously, the driver would throw a `SQLException`.
2.7	New	**hdbalm Support for Python 3**	The hdbalm tool is now supported for Python 3.6 runtime and later. The tool is still compatible with older Python 2.6 and 2.7 runtimes. For further information, see SAP Note 2990848.
2.7	New	**The Python Driver Can be Installed Under 32-bit Python on Microsoft Windows**	The following limitations apply: • The maximum length of a LOB column for the 32-bit Python driver on Microsoft Windows is 2147483647. • The maximum rowcount that can be returned for the 32-bit Python driver on Microsoft Windows is 2147483647.

Client Version	Type	Description	
2.7	New	**X.509 Authentication Support for SQLDBC-based Clients**	SQLDBC-based clients now support x509 certificate authentication. By default, the certificate information is stored in the same `sslKeystore` location, but can also be specified by a new connection property, `authenticationX509`, pointing to a PEM string or file location for a separate keystore. The new authentication mode is enabled by default when no username or password is provided and overrides the previous Kerberos default if both are configured. To re-enable the default Kerberos authentication, use the new connection property `authenticationMethods`. SQLDBC Connection Properties
2.7	New	**SQLDBC Tracing Now Supports PACKET Summary Tracing**	SQLDBC tracing now supports PACKET summary tracing, enabled by `hdbsqldbc_cons TRACE PACKET 0` or `PACKET=0`. SQLDBC Tracing

Client Version	Type	Description
2.7	New	**SQLDBC Tracing to stdout and stderr** Tracing can now be sent to the application console by specifying "stdout" or "stderr" as the trace file name. This is supported through the trace configuration tools, hdbsqldbc_cons and hdbodbc_cons, and through the HDB_SQLDBC_TRACEFILE environment variable. Tracing to the console is also supported on a per-connection basis using two new connection properties `traceFile` and `traceOptions`. SQLDBC Tracing Environment Variables SQLDBC Connection Properties
2.7	New	**JDBC Connection Properties** The new `authenticationMethods` property provides a comma-separated list of authentication methods to be used by the client. JDBC Connection Properties
2.7	New	**Optional Node.js Client Parameters for the ResultSet.getValues Method** The `options` parameter specifies the optional object containing the options for defining the representation of rows in the result. The `callback` parameter specifies the optional callback function. getValues([Options][, Callback]) Method

Client Version	Type	Description	
2.7	Changed	**The Node.js Client Throws Errors as JavaScript Errors**	The old Node.js client threw errors as JavaScript Objects. Now, the Node.js client throws errors as JavaScript Errors. The properties remain the same (message, code, sqlState, stack). However, some Node.js functions (for example, “console.log(err)” and “JSON.stringfy(err)”) return different results when using the new Node.js client. Due to this, you may need to make some minor changes in the application code.
2.7	New	**Node.js Client Added Suffix to Duplicated Column Names**	The Node.js client now adds a unique suffix identifier to duplicate column names in a given SQL statement to prevent data loss. For example, `connection.exec(“SELECT 1 A, 2 A FROM DUMMY”)` returns `{‘A’: 1, ‘A:2’: 2}`. However, since data loss is still possible using this format, it is highly recommended that you specify unique aliases for each column in a SQL statement.

Client Version	Type	Description
2.7	New	**JDBC Tracing to stdout and stderr** Tracing can now be sent to the application console by specifying "stdout" or "stderr" as the trace file name. This is supported through the trace configuration tool (java -jar ngdbc.jar), and through the HDB_JDBC_TRACEFILE environment variable. Tracing to the console is also supported on a per-connection basis using the existing `traceFile` and `traceOptions` connection properties. Environment Variables JDBC Connection Properties JDBC Tracing and Trace Options
2.7	New	**Node.js Client 2.7 Supports Node.js 8+ Versions** The Node.js client is implemented as a native add-on using N-API, independent from the underlying JavaScript runtime. It is Application Binary Interface (ABI) stable across versions of Node.js.
2.6	New	**Node.js Connection Properties** The new `allowFetchWarnings` connection property controls whether server warnings from retrieving result set of a query are returned to a JavaScript when it calls the next function of the ResultSet class. Node.js Connection Properties

Client Version	Type	Description
2.6	New	**Node.js Environment Variables** The new `HDB_NODEJS_THREADPOOL_SIZE` environment variable specifies the number of worker threads in the thread pool. The formerly used `UV_THREADPOOL_SIZE` environment variable is still respected, but is overridden by `HDB_NODEJS_THREADPOOL_SIZE`. If neither is set, a default value of 4 worker threads is used. Environment Variables Tuning an Application
2.6	New	**ODBC Connection Properties** The new `allowFetchWarnings` controls whether server warnings from retrieving the result set of a query is returned to an ODBC application when it calls ODBC APIs SQLFetch or SQLExtendedFetch. ODBC Connection Properties

Client Version	Type	Description	
2.6	New	**SQLDBC Connection Properties**	SQLDBC-based connections using OpenSSL now trust the Certificate Authorities (CAs) provided with your operating system or OpenSSL installation by default. For example, connections to SAP HANA Cloud no longer require configuration to provide the "DigiCert Global Root CA" in the `sslTrustStore` connection property and can take advantage of the certificate already provided with the OpenSSL installation. This feature does not apply to CommonCryptoLib connections. This feature is similar to how JDBC connections and SQLDBC connections using the MSCrypto provider function. A new connection property `sslUseDefaultTrustStore`, when set to FALSE, can be used to revert to the original behaviour of not trusting the default CAs. SQLDBC Connection Properties
2.6	New	**SQLDBC and JDBC Connections (SAP HANA Cloud)**	Connections to SAP HANA Cloud are automatically redirected to an appropriate internal database server to improve resource use and query latency.

Client Version	Type	Description	
2.6	New	**README.html is Included with SAP HANA Client Installation**	The SAP HANA client software package now comes with a README.html file that includes helpful information and links regarding the client installation.
2.6	New	**Connections to Port 443 (SAP HANA Cloud) Are Encrypted by Default**	For SQLDBC and JDBC, connections to port 443 (SAP HANA Cloud) are encrypted by default and no longer need to specify `encrypt`=TRUE in the connection properties. To disable this behaviour, use the `encrypt`=FALSE connection property.
2.6	New	**The SAP HANA Calculation View API for SAP HANA Cloud has been Updated to Version 1.4.1**	The following enhancements are included: • Improved Javadoc • Improved tracing • Moved certain classes to the correct package • Relaxed the regex pattern for identifier • Added missing fields and methods • Removed unneeded classes, fields, and methods

Client Version	Type	Description	
2.6	New	**The SAP HANA Calculation View API for SAP HANA Platform has been Updated to Version 1.4.2**	The following enhancements are included: • Improved Javadoc • Improved tracing • Moved certain classes to the correct package • Relaxed the regex pattern for identifier • Removed unneeded classes, fields, and methods
2.5	New	**.NET Core Enhancements**	A new environment variable, HDBDOTNETCORE, and examples have been added for .NET Core. Run the .NET Core Examples
2.5	New	**.NET Core Supports Linux and macOS**	You can now develop .NET Core applications on Linux and macOS with the SAP HANA client. The .NET Core Provider Available Client Interfaces on UNIX, macOS, or Linux
2.5	New	**SAP HANA Calculation View API**	The new SAP HANA Calculation View API is a Java-based Application Programming Interface (API) which allows you to create SAP HANA calculation views programmatically.SAP HANA Calculation View API

Client Version	Type	Description	
2.5	New	**Go Connection Property**	The `utcTimeConversion` connection property controls whether or not time.Time objects used as input parameters should be converted to UTC before upserts are executed. Go Connection Properties
2.5	Changed	**SAP HANA HDBSQL Output Command Separator**	The separator for the `output` command at the end of a specified line in an SAP HANA HDBSQL script is now optional.
2.5	New	**SAP HANA HDBSQL Options**	The new *-stdin* option reads SAP HANA HDBSQL options from the standard input. The new *-optionsfile* `<file>` option specifies that command-line options must be listed in a one option per line format in the file. The new *-h2* option displays all help options, as opposed to -h, which only displays the most common help options. SAP HANA HDBSQL Options
2.5	Changed	**SAP HANA HDBSQL -b Option**	The argument value is now case insensitive. SAP HANA HDBSQL Options

Client Version	Type	Description	
2.5	Changed	**Updates to the SAP HANA HDBSQL Exit Code**	The SAP HANA HDBSQL exit code that is returned when an error occurs has been updated: • When SAP HANA HDBSQL fails to open a file, it exits with '1' instead of the system error code. • When SAP HANA HDBSQL encounters an error processing substitution variables, it exits with '4' instead of '1'. • When SAP HANA HDBSQL fails to parse the provided command line options, it exits with '2' instead of '0'. • When SAP HANA HDBSQL exits with a SQL error it exits with '3' instead of the SQL error number truncated to the platform-specific exit code size. SAP HANA HDBSQL exits with an SQL error if the final statement executed returned an error, or on the first statement to raise an error if the *-e* option is provided. If the *-e* option is followed by a valid integer argument, that is used as the exit code instead of '3'.
2.5	New	**JDK Support Update**	For the SAP HANA JDBC driver versions 2.5 and greater, the minimum supported version of the Java Development Kit (JDK) is 1.8.

Client Version	Type	Description	
2.5	New	**JDBC Static Methods**	The new suspendTrace(java.sql.Connection) method suspends JDBC tracing on a per connection basis while the new resumeTrace(java.sql.Connection) method resumes JDBC tracing on a per connection basis. suspendTracing(java.sql.Connection) Method resumeTracing(java.sql.Connection) Method

Client Version	Type	Description
2.5	New	**JDBC Connection Properties** The following new TCP connection properties are supported when running under JVM 11 and later for the Linux, macOS, and Microsoft Windows client: • The `tcpKeepAliveCount` connection property controls the number of KeepAlive packets sent to the server before the driver assumes that the connection is closed. • The `tcpKeepAliveIdle` connection property controls the number of specified seconds that the connection is idle before the driver sends a KeepAlive packet. • The `tcpKeepAliveInterval` connection property controls the number of specified seconds before the driver sends an additional KeepAlive packet if there was no response to the previous KeepAlive packet. The following new connection properties are supported when running SAP HANA 2.4 Rev 41: JDBC 2.4.54 or later or SAP HANA 2.5: JDBC 2.5.0 or later: • The `traceFile` and `traceOptions` connection properties, when used together, implicitly enable tracing for the current JDBC connection only. The new `proxyHttp` connection property enables HTTP proxy authentication. The new `bindAddress` connection property specifies that the provided address is used as the source IP/host for the socket connection.

Client Version	Type	Description
		The new `nonBlockingIO` connection property specifies whether to block I/O when sending and receiving packets: TRUE/FALSE. The default is TRUE, which specifies that I/O should not be blocked. Setting this property to FALSE blocks the I/O The new `sslSNIRequest` connection property specifies whether SNI requests are enabled for TLS connections: TRUE/FALSE. JDBC Connection Properties
2.5	New	**JDBC Environment Variables** The following new environment variables are supported when running SAP HANA 2.4 Rev 41: JDBC 2.4.54 or later or SAP HANA 2.5: JDBC 2.5.0 or later. • The HDB_JDBC_TRACEFILE and HDB_JDBC_TRACEOPTIONS environment variables, when used together, implicitly enable tracing for all JDBC connections established in the current JVM. Environment Variables

Client Version	Type	Description	
2.5	Changed	**JDBC Tracing Enhancements**	JDBC trace files now include the trace settings in the trace file header. Additionally, the settings are written to the trace file if they change while tracing is enabled. When JDBC connection tracing is enabled, calls to the JDBC static methods are logged in the trace file. Trace a JDBC Connection Using a Connection String Trace a JDBC Connection Using the Command Line Trace a JDBC Connection Using the GUI
2.5	New	**JDBC Trace Option**	The new "Show elapsed times" tracing option enables/disables elapsed times for each JDBC API call trace record and each packet send/receive record. JDBC Tracing and Trace Options
2.5	New	**Node.js on Linux Alpine**	You can now build applications on Linux Alpine that make use of the SAP HANA Node.js client interface. Configure the Node.js Driver (Client Install)
2.5	New	**Node.js Support**	The Node.js driver now supports node version 12.

Client Version	Type	Description	
2.5	New	**Node.js Connection.getWarnings Method**	This method returns the warnings for a connection. getWarnings() Method
2.5	New	**Node.js Connection Properties**	The new `dataTruncationError` property, when set to TRUE, fails the stored procedure call whenever a data truncation error occurs. The new `getDriverVersion` property retrieves the version of the node.js binaries. The `maxPoolSize` connection property defines the maximum number of connections that are allowed in the connection pool. The `connectionLifetime` specifies the maximum time, in seconds, that the connection is cached in the pool. A value of 0 causes pooled connections to be cached permanently. Node.js Connection Properties

Client Version	Type	Description	
2.5	New	**Node.js Environment Variable**	The new `HDB_NODE_PLATFORM_CLEAN` environment variable removes, at npm install time, unnecessary native platform libraries that will never be used after the Node driver is installed. To invoke this cleaning behavior, specify `HDB_NODE_PLATFORM_CLEAN=1`. This is not recommended if you are vendoring dependencies. Environment Variables
2.5	Changed	**Microsoft Windows ODBC Manager Enables Server Certificate Validation by Default**	The *Validate the TLS/SSL* certificate option has been removed so that, by default, the server certificate is validated when using TLS/SSL. Disable this validation clicking the *Advanced* button and setting the connection property `sslValidateCertificate` to FALSE. Connect to SAP HANA via ODBC
2.5	Changed	**ODBC Tracing**	ODBC trace files output enabled trace settings at the start of every new file, even when the trace settings have changed while the application is already running.

Client Version	Type	Description	
2.5	New	**ODBC Connection Attributes**	The SQL_ATTR_BYTES_SENT connection attribute retrieves the total number of the network bytes sent to the server since the database connection was established. The total sent bytes are reset to zero by the driver when the connection is disconnected. The SQL_ATTR_BYTES_RECEIVED connection attribute retrieves the total number of the network bytes received from the server since the database connection was established. The total received bytes are reset to zero by the driver when the connection is disconnected. ODBC Connection Attributes
2.5	New	**SQLDBC Supports Input Table Parameters for SAP S/4HANA**	SQLDBC now supports input table parameters for SAP S/4HANA. Input table parameters are not supported by applications other than SAP S/4HANA, nor is it supported on clients other than SQLDBC.

Client Version	Type	Description	
2.5	New	SQLDBC Connection Properties	The following new connection properties are supported when running under Linux, macOS, and Microsoft Windows clients: • The `tcpKeepAliveCount` connection property controls the number of KeepAlive packets sent to the server before the driver assumes that the connection is closed. • The `tcpKeepAliveIdle` connection property controls the number of specified seconds that the connection is idle before the driver sends a KeepAlive packet. • The `tcpKeepAliveInterval` connection property controls the number of specified seconds before the driver sends an additional KeepAlive packet if there was no response to the previous KeepAlive packet. The new `proxyHttp` connection property enables HTTP proxy authentication. The `isoTimestampOutput` connection property controls whether or not T is used as the separator between the Date and Time parts of a timestamp value. If T is not used, then a space is used as the separator. The new `secondarySessionFallback` property forces the ongoing transaction on a non-anchor connection to fall back to the anchor/primary connection if this connection is dropped by the network or server. The new `bindAddress` connection property specifies that the

Client Version	Type		Description
			provided address is used as the source IP/host for the socket connection. The `nodeConnectTimeout` property now defaults to 60 seconds. The new `sslSNIRequest` connection property specifies whether SNI requests are enabled for TLS connections: TRUE/FALSE. SQLDBC Connection Properties
2.5	New	**SQLDBC Tracing**	Use the HDB_SQLDBC_TRACEFILE and HDB_SQLDBC_TRACEOPTS environment variables to trace an application if you do not have hdbsqldbc_cons installed or if you are using a standalone driver such as Node.js. SQLDBC Tracing
2.5	Changed	**Update to the Client Installation on Microsoft Windows**	On Microsoft Windows, the Client Installer now adds the SAP HANA Client installation directory to the PATH environment variable. The directory is also removed from the PATH environment variable during the SAP HANA Client uninstallation. Install the SAP HANA Client on Microsoft Windows

Client Version	Type	Description	
2.4	New	**Direct TCP and TLS Connections Through an HTTP Proxy**	You can create an HTTP proxy connection without using WebSockets, allowing direct TCP connections via an HTTP proxy. HTTP Proxy Client Connections Implement HTTP Proxy Client Connections JDBC Connection Properties
2.4	New	**SAP HANA Cloud Support**	SAP HANA Cloud supports SNI routing. You must use version 2.4.167 (2.4.67 for the JDBC driver) or later of the SAP HANA client interfaces with SAP HANA Cloud. There are also restrictions on the platforms that support SAP HANA Cloud.

Related Information

SAP HANA Platform
SAP Note 1906576
LOB Streaming for INSERT, UPDATE, DELETE, and SELECT Statements

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:
  - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
  - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.
- Links with the icon : You are leaving the documentation for that particular SAP product or service and are entering an SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Bias-Free Language

SAP supports a culture of diversity and inclusion. Whenever possible, we use unbiased language in our documentation to refer to people of all cultures, ethnicities, genders, and abilities.

www.sap.com/contactsap

THE BEST RUN SAP