

Zeos 7.2.6 Release Notes

Jan Baumgarten

October 23, 2019

Zeos 7.2

The Zeos Team is proud to announce the availability of Zeos 7.2.6 as a stable release. This is the newest stable version of Zeos. It deprecates Zeos 7.1, and all prior versions of Zeos 7.2 which will not be supported anymore. Zeos 7.2 has seen tons of bug fixes and improvements. We urge all people still using older versions of Zeos to upgrade. If you have any problems with Zeos 7.2, please get in contact with us on the forums (<http://zeoslib.sourceforge.net>) or on the bugtracker (<https://sourceforge.net/p/zeoslib/tickets/>).

Chapter 1

Zeos 7.2.6 Release Notes

1.1 General

1.1.1 New C++ Builder Packages

This release of Zeos has new packages for C++ Builder versions 2006, 2009, XE2 and XE 10.2. Each package subdirectory contains a Manual-ZeosDbo-*.txt file that has detailed installation instructions. For C++ Builder 2006 the necessary files are generated using the Delphi 2006 package files. See Manual-ZeosDbo-2006.txt for more information.

1.1.2 Fixed Bugs

- 253: Zeos had an issue that prevented it from filtering on computed columns which was fixed.
- 252: TZConnection.AutoEncodeStrings was not restored correctly on ANSI Delphis (anything until Delphi 2007) and on Lazarus.
- A bug about metadata caching was fixed that could lead to the TZSQL-Metadata not giving correct results if meta data for the same object was queried a second time. See <http://zeoslib.sourceforge.net/viewtopic.php?f=40&t=73973> for more information.
- 369: A bug has been fixed, where a filter didn't work correctly on the last letter of a character column.

- 356: When a select contains multiple tables and fields were selected using the table alias name and when the "first M skip N" option is present, the first field may be missing in the result set.

1.1.3 Changes

binary connection strings are deprecated

Use of character sets that move the correct interpretation of strings to the client side is deprecated with this release. We will discontinue support for these connection strings with the Zeos 7.3 release. In Zeos 7.3 usage of these character sets as connection character sets will lead to an exception. Background: Zeos tries to preserve your data. The database systems usually are better at character set conversions than we are. So we decided to fully rely on them. On Firebird and Interbase these are the character sets "none" and "octetts". On PostgreSQL this is the character set SQL_ASCII. On MySQL and MariaDB this is the character set "BINARY".

Limited support for TGraphicField

Zeos now supports the use of persistent TGraphicField components with drivers that support BLOB fields.

1.1.4 Known problems

Filtering on computed columns or lookup columns does not work

Filtering on computed columns or lookup columns will generate an exception: "Column with index -1 is not accessible". This will not be fixed in Zeos 7.2 because it requires a major rewrite of the filtering mechanism.

No support for BCD type columns

Zeos currently doesn't support the BCD type columns of Delphi. NUMERIC and DECIMAL columns still get mapped to floating point types. This will be addressed in Zeos 7.3 because it requires a lot of changes in the Zeos core.

Metadata Caching

Zeos will usually aggressively cache metadata of your database. Because of this scenarios where your database structure is changing constantly, are not well supported. You can call `ZConnnection.DbcConnection.GetMetadata.-ClearCache` to clear out the caches. You can also disable the `UseMetadata` property of the `TZConnection` object. All data sets will become readonly in that case. Use `TZUpdateSQL` if you need them to be writable again.

Blob fields are read as a whole

As soon as you touch a blob field for reading, Zeos will fetch the whole blob contents from the database server. This may lead to high memory consumption.

1.2 Firebird / Interbase

1.2.1 Fixed Bugs

- 249: In previous versions of Zeos the Firebird driver silently only copied parts of string parameters to the server if the client library didn't provide enough buffer space for the whole string. This has been rectified. Zeos will now throw an exception in these cases. As a workaround you might want to explicitly cast your parameters - i.e. where `:param = 'x'` becomes where `cast(:param as varchar(50)) = 'x'`.
- 351: Firebird: A bug has been fixed where NUMERIC and DECIMAL columns could be converted to different field types depending on the way they were selected.

1.2.2 Breaking Changes

field types for NUMERIC and DECIMAL

There was a bug in the interbase / firebird driver where NUMERIC and DECIMAL fields could be assigned different field types. This bug has been fixed but can lead to problems because now on persistent fields and typecasts in your source code because Zeos might assign a different field type to your fields than it assigned before.

Value	Description
inet	This will create a connection string to be used with TCP/IP. If the server name is empty, localhost will be used as the server name.
wnet	This will create a connection string to be used with named pipes. If the server name is empty, a connection to a local server will be created.
xnet	On Firebird 3 this will force the client library to use an xnet (shared memory) connection to a local server. On Interbase and Firebird 2.5 and below the embedded server cannot be forced to use xnet, while the regular client library will always use xnet in that case. The server name will be ignored with this setting.
local	This value will try to do a local connection. This will only work with embedded servers. Regular client libraries will usually fall back into using xnet.

Table 1.1: values for fb`protocol

1.2.3 Changes

new connection parameter fb`protocol

The new parameter "fb`protocol" was introduced as a connection parameter. With Firebird 3 this parameter switches Zeos to use url style connection strings. This enables you to force the library to use xnet for connecting to the server, bypassing a local engine12. This parameter also allows you to use the wnet (Named Pipes) protocol on older versions of Firebird and on Interbase. For possible values of fb`protocol see table 1.1 on Page 5.

reading the client library version

The firebird/interbase driver now can read the client library version of Firebird 1.5+ and Interbase 7.0+.

Unicode characters in database name

If a firebird 2.5+ library is detected, Zeos passes the database name using UTF8, allowing all Unicode characters to be used.

new default mode for read committed on Firebird 4.0

On Firebird 4.0 Zeos uses the new read committed default mode "READ COMMITTED READ CONSISTENCY" when it detects a Firebird 4.0+ server version.

1.2.4 Known Problems

no support for identity columns

Firebird 3.0 introduced a new feature called "IDENTITY columns". These columns can use an internal generator to automatically create their values. Zeos currently does not support using these columns. It cannot read back values that were assigned to these columns automatically.

The tokenizer does not understand single line comments

The Zeos tokenizer for Firebird and Interbase does not understand single line comments. As a workaround we suggest you use multi line comments, as shown in the following listing:

```
-- this single line comment is not supported

/* use this multi line comment instead */
```

1.3 SQLite

1.3.1 Fixed Bugs

- 248: selecting the rowid in a table made Zeos move a Query into Read-Only mode because it thought that rowid was not part of the table. This has been fixed.

1.3.2 Breaking Changes

Varchar fields without a length specifier

In previous versions of Zeos Varchar fields without a length specifier were mapped as TStringField fields. Zeos now maps them as TMemorField since

Zeos doesn't have any idea on how long you want the contents of these fields to be. Use the parameter `Undefined`Varchar`AsString`Length` to get the old behaviour and specify a length that you are comfortable with. I.e. use `Undefined`Varchar`AsString`Length=40`.

1.3.3 Changes

extended error information

The SQLite driver of Zeos 7.2 now can get extended error information from SQLite and provide much better error descriptions. In Zeos 7.3 this will be the default. In Zeos 7.2 we introduced the new parameter `ExtendedErrorMessage` to enable the behaviour. Setting this parameter to 1 / Yes / true will get you the extended error messages. This parameter will not exist in Zeos 7.3 anymore.

backup API

We added the SQLite backup API to the plain driver. A working example how to use it can be found on <http://zeoslib.sourceforge.net/viewtopic.php?f=40&p=126408>

1.4 MySQL / MariaDB

1.4.1 Fixed Bugs

- 255: The ability to use the `Bit(1)` datatype as a boolean data type is now correctly determined even if the MariaDB client and MySQL server are mixed with each other.
- Reading metadata about stored procedures from MySQL 8.0 now works. This had side effects on the way Metadata is read. For more information see 1.4.2 stored procedures metadata, Page 8.

1.4.2 Breaking Changes

stored procedures metadata

Until Zeos 7.2.4 when querying for procedure columns (parameters) the database name was returned in the PROCEDURE`CAT column. PROCEDURE`CAT now always returns "def" while the database is returned in the PROCEDURE`SCHEM column. This is more in line with other MySQL metadata queries and also with MySQL / MariaDb themselves. Also the scale for NUMERIC and DECIMAL now is returned in the SCALE column and not in the LENGTH column anymore. If you relied on this behavior your code might break!

1.4.3 Known Problems

using TZUpdateSQL

The TZUpdateSQL component will raise an error if a record is posted to the connected Dataset where no values were changed. This is because MySQL / MariaDB returns the number of changed records by default. If a record didn't change, 0 is returned instead. See <https://dev.mysql.com/doc/refman/8.0/en/mysql-affected-rows.html> for more information. This can be mitigated by adding "CLIENT`FOUND`ROWS=1" to the TZConnection.Properties property.

1.5 SAP / Sybase ASE

1.5.1 Known Problems

unichar / univarchar / unitext doesn't work

Using unichar / univarchar / unitext columns with the sybase driver doesn't work. Use freetds if you need this to work.

1.6 Sybase / SAP ASA

We did a large amount of improvements that have no ticket number assigned. Please let us know if you find problems in the ASA driver.

1.7 PostgreSQL

1.7.1 Fixed Bugs

- 246: Zeos had a problem where it couldn't clean up its internal state correctly in the event of a connection loss to the server. This has been fixed.
- 274: The TZPGEventAlerter component now works correctly in a 64 bits environment.
- 350: PostgreSQL: It wasn't possible to correctly execute stored functions on PostgreSQL without unpreparing them first.

1.7.2 Changes

support for citext

We added support for the citext module. See <https://www.postgresql.org/docs/current/citext.html> for more information on citext.

1.7.3 Known problems

using stored procedure is not possible

Recent versions of PostgreSQL introduced the new stored procedures - as opposed to stored functions, which have been part of PostgreSQL for quite some time now. Zeos 7.2.6 does not support using PostgreSQL stored procedures with the TZStoredProc component. Stored functions continue to work as before.

Chapter 2

Zeos 7.2.4 Release Notes

2.1 General Changes

2.1.1 Supported compilers

Zeos 7.2 supports Delphi versions from Delphi 7 to XE 10.2 Tokyo. Only the Win32 and Win64 compilers are supported. Other platforms utilizing the NextGen compiler are not (yet) supported. The Free Pascal compiler is supported from version 2.6 to version 3.0. Older versions might work but don't get official support by us anymore.

2.1.2 Date and Time format settings

Zeos now supports specifying date and time format settings that will be used if Zeos doesn't know how to correctly format date and time settings for the DBMS to understand them. This feature gets used with emulated parameters in ADO - if the FoxPro driver is used for example. These new parameters can be set in the `TZConnection.Properties` property. The format of these parameters conforms to the usual Delphi standards. The following new parameters are supported:

- `DateReadFormat`
- `DateWriteFormat`
- `DateDisplayFormat`

- TimeReadFormat
- TimeWriteFormat
- TimeDisplayFormat
- DateTimeReadFormat
- DateTimeWriteFormat
- DateTimeDisplayFormat

The ReadFormat parameters describe the date and time formats as the database sends them to the application. The WriteFormat parameters describe the date and time formats as the application should send them to the database. The DisplayFormat settings are used for strings that are supplied by the application to Zeos. These settings can be used to solve problems on the following drivers:

- dblib (FreeTDS, mssql, sybase)
- postgresql
- mysql (depending on parameter preferprepared, used if preferprepared is not set)
- ado (emulated parameters, see 2.3.6 ADO, Page 17)

2.1.3 Batch Loading

We added API support for batch loading of data into databases that support this feature. Currently this feature is only available at the DBC layer. Assume the following example:

```
uses Types, ZVariant, ZDbcIntfs;

procedure Example.InsertBatchIntoForBar(const Connection:
    IZConnection);
var
    Value_NullArray: TBooleanDynArray;
    ID_IntArray: TIntegerDynArray;
    Value_StringArray: TStringDynArray;
    I: Integer;
    Stmt: IZPreparedStatement;
begin
```

```

Connection.SetAutoCommit(False);
try
  SetLength(Value_NullArray, 10);
  SetLength(ID_IntArray, 10);
  SetLength(Value_StringArray, 10);
  for i := 0 to 9 do begin
    ID_IntArray[i] := i;
    {every other string is null in our example}
    if i mod 2 = 0 then begin
      Value_NullArray[i] := False;
      Value_StringArray[i] := SysUtils.IntToStr(i);
    end else begin
      Value_NullArray[i] := True;
    end;
  end;
  { assume the id field is not nullable, so we don't need a
    null indicator array }
  Stmt := Connection.PrepareStatement(
    'insert into FooBar(ID, AValue), values
    (?,?)');
  Stmt.SetDataArray(FirstDbcIndex, ID_IntArray, stInteger);
  {$IFDEF UNICODE}
  Stmt.SetDataArray(FirstDbcIndex+1, Value_StringArray,
    stString, ZVariant.vtUnicodeString);
  {$ELSE}
  Stmt.SetDataArray(FirstDbcIndex+1, Value_StringArray,
    stString, ZVariant.vtRawByteString);
  {$ENDIF}
  Stmt.SetNullArray(FirstDbcIndex+1, stBoolean,
    Value_NullArray);
  I := Stmt.ExecuteUpdatePrepared;
  Assert(i = 10, 'WrongUpdateCount');
  Connection.Commit;
except
  Connection.Rollback
end;
end;

```

First we start a new transaction using `SetAutoCommit(False)`. This is for several reasons:

- we get better performance, which always should be good ;-)
- For some databases Zeos will generate more than one insert statement. If we were in autocommit mode, one wouldn't be able to roll back the complete insert if a problem arises at the end of the batch.

Afterwards we prepare array that contain the values to be inserted. In a real world example you most probably already have read them from a file. Since typed arrays can't contain null values, a separate array can be prepared that tells Zeos which values are to be set to null (`Value`NullArray`). So you can pass two arrays per column: One containing the actual values (`ID`IntArray`, `Value`StringArray`) and one containing rows that are to be set to null (`Value`NullArray`). We don't create an `ID`NullArray` because in our example we assume the the ID field is not nullable. Please note: You may not delete or modify the arrays until you have imported them because Zeos will only store pointers to the arrays and not the actual arrays. Finally we call `ExecuteUpdatePrepared` to do the actual import.

Please note that we use `FirstDbcIndex` for generating column indexes. This is because Zeos can be compiled in two modes: One where the column index starts with 1 and one where the column index starts with 0. `FirstDbcIndex` will be set accordingly.

2.2 Behaviour changes

2.2.1 Case sensitivity of the TZMetadata object and DBC layer meta data functions

If you call Metadata functions on the DBC layer or use the TZMetadata object be sure that the case of the object name that you retrieve information for is in the correct case. Zeos will not do any guesswork - it will simply query the underlying database for the identifier that you supply. Example: In former versions of Zeos the call `GetColumns('PEOPLE')` might have returned information for the table people. This will not happen anymore. To query information about the table people you will have to use `GetColumns('people')`. If you want the former behavior restored, your call has to be like this: `GetColumns(IZConnection.GetMetadata.NormalizePatternCase('PEOPLE'))`

2.2.2 PostgreSQL autocommit and OID columns

The postgresql driver now uses the native autocommit of PostgreSQL. Writing to OID BLOBs only works in explicit transactions because of this. This is a limitation in PostgreSQL and cannot be fixed in Zeos. The proposed workaround is to use the bytea data type. For more information see <https://>

www.postgresql.org/message-id/002701c49d7e%240f059240%24d604460a%40zaphod.

2.2.3 FreeTDS default library name

The FreeTDS drivers now uses sybdb.dll / sybdb.so as the default library to load. Watch out if your program doesn't set the LibraryPath property in TZConnection.

2.2.4 Automatic opening of connections

In the past the drivers for MySQL, ADO and SQLite automatically opened a connection as soon as the connection object was created. This is changed. No driver will connect automatically anymore. Call the connect method to open the connection. This change only affects users of the DBC layer.

2.2.5 IZConnection.GetMetadata will automatically connect now

This change is on the DBC layer. Calling IZConnection.GetMetadata will connect to the database now in all cases. In previous versions of Zeos this wasn't guaranteed, leading to undefined behaviour, depending on whether the IZConnection was already connected or not.

2.3 Driver specific changes

2.3.1 PostgreSQL

- The postgresql driver now uses the native autocommit of PostgreSQL. OID columns can no longer be written to in autocommit mode. See 2.2.2 PostgreSQL autocommit and OID columns, Page 13.
- GUID columns are now supported.
- New parameters, so PostgreSQL can connect using SSL (sslmode, requiresssl, sslcompression, sslcert, sslkey, sslrootcert, sslcrl). Take a look at the PostgreSQL documentation on how to use these parameters.

- Zeos now can use SSPI, Kerberos and Windows Authentication with PostgreSQL. Just leave the username and password empty for this.
- The PostgreSQL driver now maps the transaction isolation levels `tiNone` and `tiReadUncommitted` to `tiReadCommitted`. It is no longer valid to use your own transaction handling code. Please use the `TZConnection` methods `StartTransaction`, `Commit` and `Rollback`.
- The PostgreSQL driver now supports read only transactions.
- The PostgreSQL driver now supports `+Infinity`, `-Infinity` and `NaN` for floating point values.
- Using date and time format settings with PostgreSQL is now supported. Since Zeos sends date and time parameters to the server as strings, problems with formatting can arise. If you have problems, try changing the date and time format settings. For more information see 2.1.2 Date and Time format settings, Page 10.
- If you still use PostgreSQL 7 databases, we urge you to move on to a newer version. PostgreSQL 7 is deprecated with this version and will be removed with Zeos 7.3.

2.3.2 Firebird / Interbase

- We added support for Firebird 3.0.
- We added a new parameter to enable Firebird 3 wire compression: `wirecompression`. Setting it to `1/Yes/True` will enable Firebird 3 wire compression.
- Zeos now supports Firebird 3.0 boolean fields.
- We reenabled the use of Firebird and Interbase autocommit.
- We added support for the new DBC layer batch loading API to the Firebird / Interbase driver. For more information see 2.1.3 Batch Loading, Page 11.
- If you still use Interbase 5 databases, we urge you to move on to a newer version. Interbase 5 is deprecated with this version and will be removed with Zeos 7.3.

2.3.3 MySQL / MariaDB

- TZQuery and TZReadOnlyQuery now support the use of multiple statements in the query. The first result that is returned by the server will be the result that gets loaded into the Dataset.
- The MySQL driver should now be thread safe. This still means that threads are not allowed to share a connection.
- New connection level parameter MySQL'FieldType'Bit'1'IsBoolean. If this parameter is enabled (set to 1/Yes/True), fields declared as BIT(1) will be treated as boolean fields. The old assumption that an enum('Y','N') is a boolean field is disabled if this parameter is enabled. If this parameter is enabled, enum('Y','N') will be a string field. Other enums behave as before, they will be mapped to a string field in any case. This parameter will be enabled by default in Zeos 7.3 if the server version is $\geq 5.0.3$.
- Using date and time format settings with MySQL is now supported. This can be used if the parameter preferprepared is disabled (default setting). In this mode Zeos sends date and time parameters to the server as strings. Problems with formatting can arise. If you have problems, try changing the date and time format settings. For more information see 2.1.2 Date and Time format settings, Page 10.
- The MySQL driver will not automatically connect to the database when the DBC layer connection object is created anymore. This only affects users of the DBC layer.

2.3.4 MS SQL / SAP ASE (Sybase ASE)

- We reenabled Sybase support. This should allow some basic usage, but your mileage may vary. If you have problems please get in contact.
- The driver now supports GUID-Columns on MS SQL Server.
- We enabled support of TDS 5.0 for newer Sybase Servers.
- With FreeTDS the server port can now be specified.
- The FreeTDS drivers now uses sybdb.dll as the default dll to load. Watch out if your program doesn't set the LibraryPath property in TZConnection.

- Using date and time format settings with Microsoft SQL Server and SAP Adaptive Server Enterprise is now supported. Since Zeos sends date and time parameters to the server as strings, problems with formatting can arise. If you have problems, try changing the date and time format settings. For more information see 2.1.2 Date and Time format settings, Page 10.
- If you still use the mssql protocol using ntwdlib.dll to connect to MS SQL Server, we urge you to move on to either use ADO or FreeTDS. The mssql protocol using ntwdlib.dll is not supported by Microsoft for ages now and we will discontinue its support with Zeos 7.3.

2.3.5 Oracle

- Performance improvement: The oracle driver now supports block cursor reads. This allows to fetch more than one record in one network roundtrip. The parameter for setting the block size is internal 'buffer' size.
- We added support for the new DBC layer batch loading API to the Oracle driver. For more information see 2.1.3 Batch Loading, Page 11.

2.3.6 ADO

- The ADO driver now also supports Free Pascal.
- We added support for the new DBC layer batch loading API to the ADO driver. For more information see 2.1.3 Batch Loading, Page 11.
- Zeos emulates named parameters for ADO drivers that don't support parameters. Unfortunately this means Zeos doesn't know how to correctly format timestamps and similar data types to be correctly recognized by the underlying database. Please use the new connection parameters ReadFormatSettings, WriteFormatSettings, DisplayFormatSettings. For more information see 2.1.2 Date and Time format settings, Page 10.
- The ADO driver will not automatically connect to the database when the DBC layer connection object is created anymore. This only affects users of the DBC layer.

2.3.7 SQLite

- The SQLite driver will not automatically connect to the database when the DBC layer connection object is created anymore. This only affects users of the DBC layer.

2.4 Known Problems

- Zeos currently doesn't support the BCD type columns of Delphi. NUMERIC and DECIMAL columns still get mapped to floating point types. This will be addressed in Zeos 7.3 because it requires a lot of changes in the Zeos core.
- Zeos will usually aggressively cache metadata of your database. Because of this scenarios where your database structure is changing constantly, are not well supported. You can call `ZConnnection.DbcConnection.GetMetadata.ClearCache` to clear out the caches. You can also disable the `UseMetadata` property of the `TZConnection` object. All data sets will become readonly in that case. Use `TZUpdateSQL` if you need them to be writable again.
- As soon as you touch a blob field for reading, Zeos will fetch the whole blob contents from the database server. This may lead to high memory consumption.

Chapter 3

The future development of Zeos

The next version of Zeos will be Zeos 7.3 which currently is in the making. Currently the following changes and features are planned:

- support for OLEDB
- support for ODBC
- support for BCD type columns to allow correct usage of NUMERIC and DECIMAL fields
- support for GUID type columns in Firebird
- Interbase 5 will not be supported officially anymore
- PostgreSQL 7 databases will not be supported officially anymore
- The mssql protocol that uses ntwdlib.dll will not be supported anymore. With Zeos 7.3 there will be plenty of other options to connect to Microsoft SQL Server.
- DBC layer: the use of TZDriverManager.Connect will not be supported with a string anymore, only the use of a TZURL object will be supported.
- Protocols with version numbers will not be used in Zeos 7.3 anymore (i.e. firebird-2.5 will become firebird). Please migrate.
- Zeos 7.3 will have support for the Delphi Nextgen compilers - i.e. Android, iOS, MacOS X and Linux.

If you use any of these new features, please get in contact on the forums (<http://zeoslib.sourceforge.net>).