

Zeos 7.2.4 Release Notes

Jan Baumgarten

March 25, 2018

1 Zeos 7.2

The Zeos Team is proud to announce the availability of Zeos 7.2.4 as a stable release. This is the newest stable version of Zeos. It deprecates Zeos 7.1, 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/>).

2 General Changes

2.1 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.

2.2 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 4.6 ADO, Page 8)

2.3 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.

3 Behaviour changes

3.1 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'))`

3.2 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>.

3.3 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.

3.4 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.

3.5 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.

4 Driver specific changes

4.1 PostgreSQL

- The postgresql driver now uses the native autocommit of PostgreSQL. OID columns can no longer be written to in autocommit mode. See 3.2 PostgreSQL autocommit and OID columns, Page 5.

- GUID columns are now supported.
- New parameters, so PostgreSQL can connect using SSL (sslmode, requiressl, 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 2.2 Date and Time format settings, Page 1.
- 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.

4.2 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 2.3 Batch Loading, Page 2.

- 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.

4.3 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 `MySQLFieldType_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 sting filed in any case. This parameter will be enabled by default in Zeos 7.3 if the server version is `>= 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 paramaters 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 2.2 Date and Time format settings, Page 1.
- 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.

4.4 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 2.2 Date and Time format settings, Page 1.
- If you still use the mssql protocol using ntwdblib.dll to connect to MS SQL Server, we urge you to move on to either use ADO or FreeTDS. The mssql protocol using ntwdblib.dll is not supported by Microsoft for ages now and we will discontinue its support with Zeos 7.3.

4.5 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 2.3 Batch Loading, Page 2.

4.6 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 2.3 Batch Loading, Page 2.
- 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 2.2 Date and Time format settings, Page 1.

- 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.

4.7 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.

5 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.

6 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 ntwdblib.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.