Converting Actian Zen Data to Postgres

A White Paper From

For more information, see our web site at

http://www.goldstarsoftware.com

Converting Actian Zen Data to Postgres

Last Updated: March 2026

Converting Actian Zen data to Postgres can be challenging due to the many differences between the two environments. We have created this document in order to help streamline the data migration process as much as possible. This procedure leverages some powerful tools from Goldstar Software, including *DDF2XML*, *GSSync*, *FPrint* and *MakeBat*.

You may not need to execute every section of this procedure, as some of these tasks may already be done, but the process essentially looks like this:

1. Download Postgres
2. Install Postgres
3. Install the 64-bit and 32-bit ODBC Drivers
4. Create a New Database in Postgres
5. Export the Schema from Actian Zen
6. Create the Tables in Postgres
7. Create a New 32-bit ODBC DSN
8. Migrate the Data

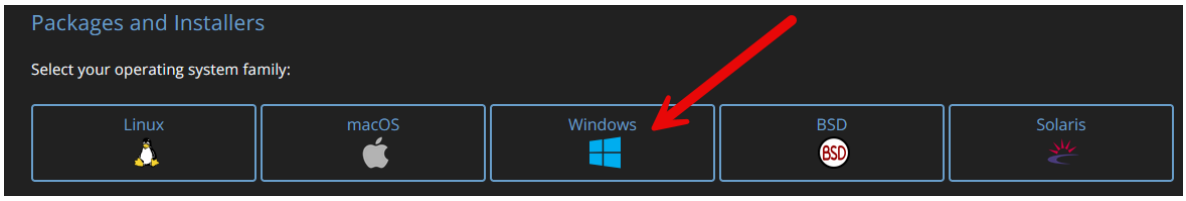
Download Postgres

At the time of this writing, the current release is v18.3.2.

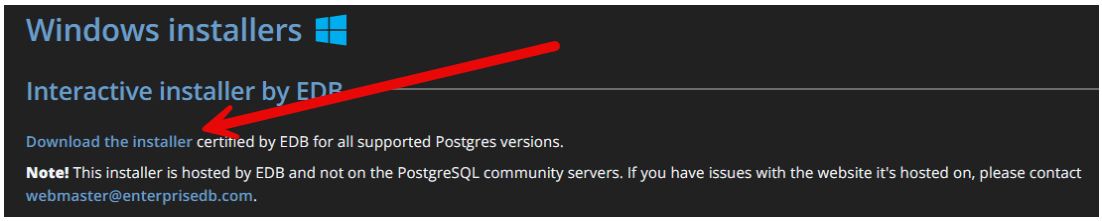
Use your browser to go to this link:

https://www.postgresql.org/download/

Click on the Windows download button:



Click on the download link here:

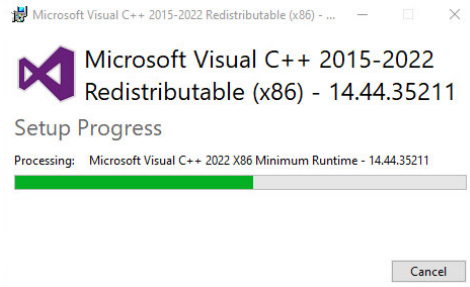


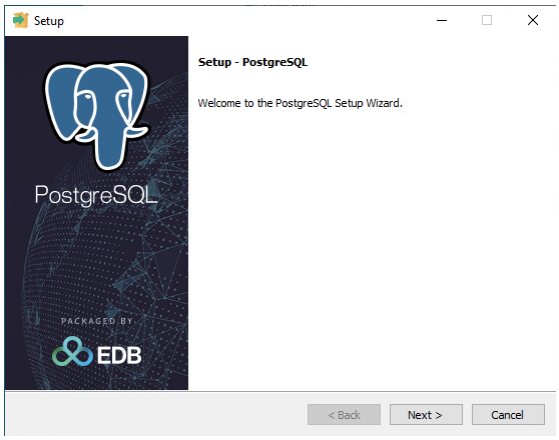
Download the file from here:

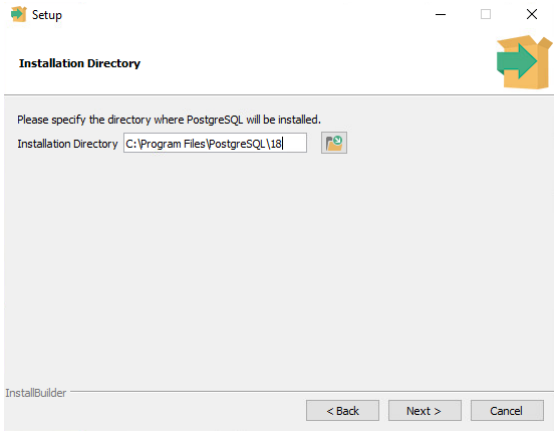
Wait for the download to complete.

Install Postgres

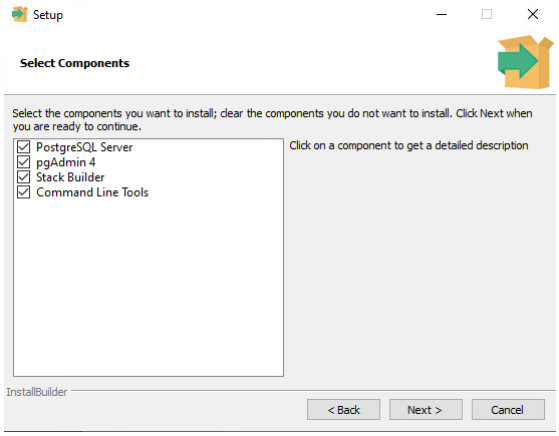
After the download has been finished, run the downloaded file to start the installation process. You will see screens that look like the following:

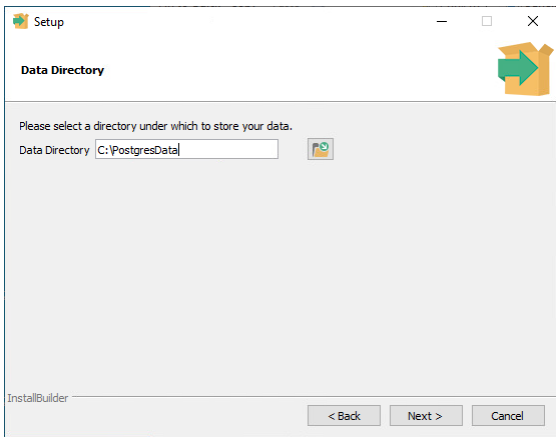




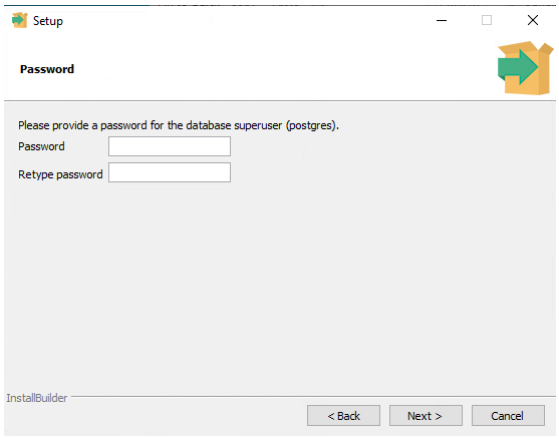


This is the default location, which should be OK for most installations.

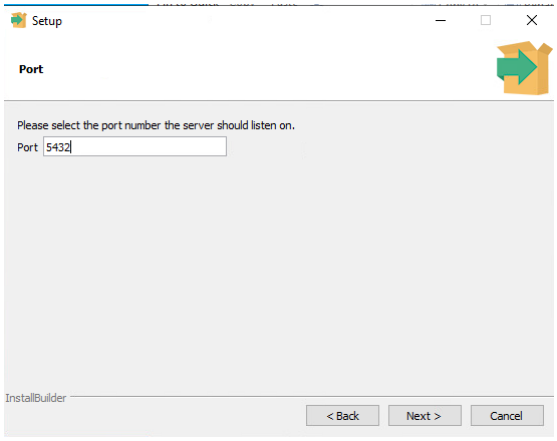




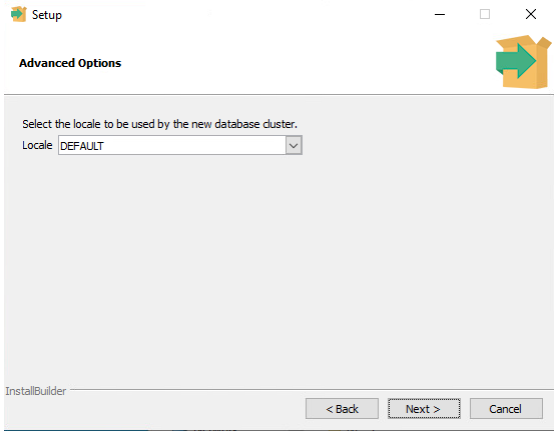
The default location for your database is a DATA folder located inside the install folder (from a few screens back), but this puts it in the C:\Program Files folder, which may not always have sufficient disk space or offer the expected security. We changed it here to make it obvious, but consider putting it onto a separate data volume with sufficient disk space.

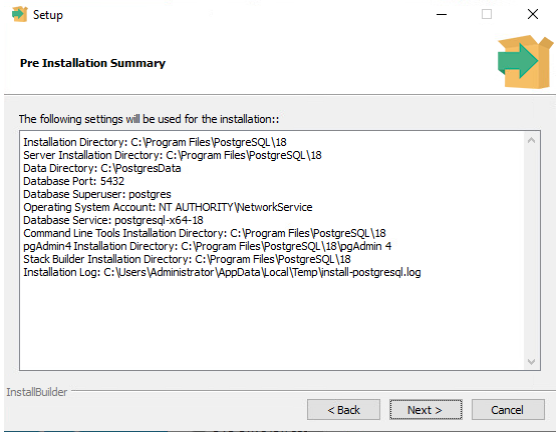


The text you enter here will be your administrative password for the user *postgres*. Remember the password you enter, as you **will** need it later!

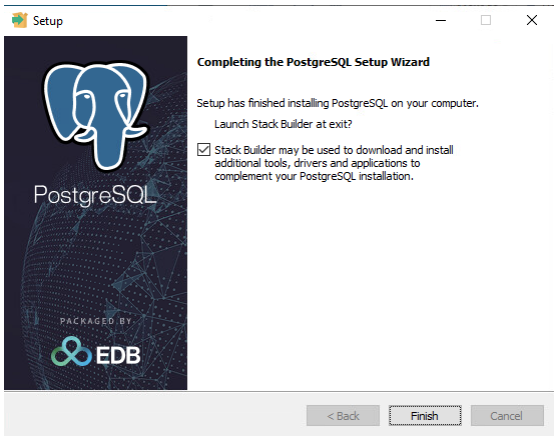


The default port is 5432, and there should be no reason to change this for a standard install. If you are installing multiple versions on the same server, then you may need to give each its own port.

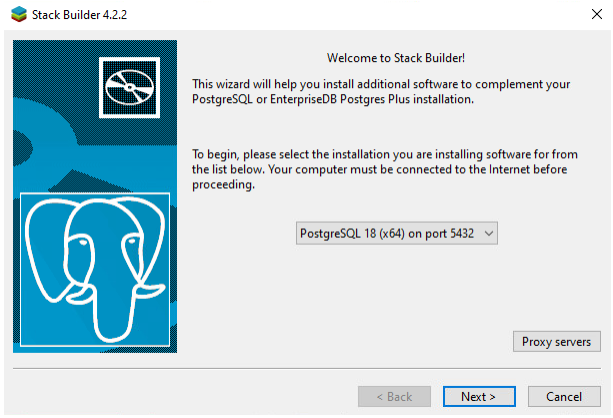




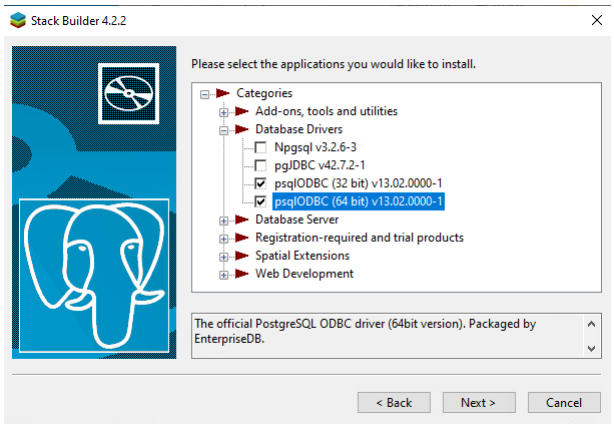
This is your last change to look over these values!



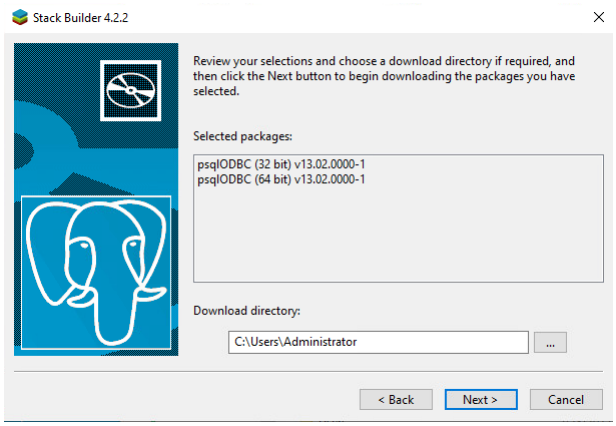
The installation will run for quite a long time (compared with Actian Zen, anyway). When it completes, you will run StackBuilder to install the ODBC drivers:

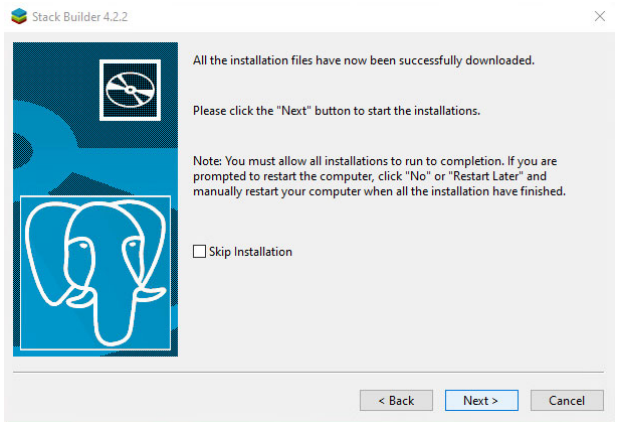


Select the Postgres instance you are modifying:



Select the ODBC drivers here for both 32-bit and 64-bit.

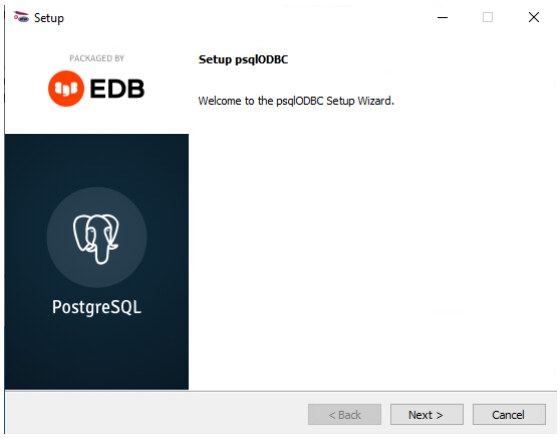


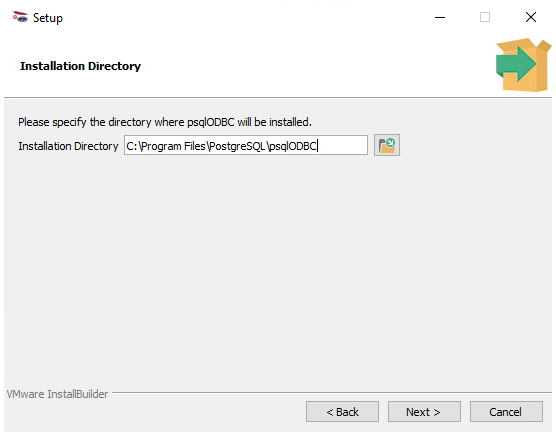


When this finishes, you will have both 32-bit and 64-bit ODBC drivers downloaded.

Install the 64-bit and 32-bit ODBC Drivers

Now that the Postgres engine is installed, you can install the ODBC drivers. We did the 64-bit drivers first, which looked like this:





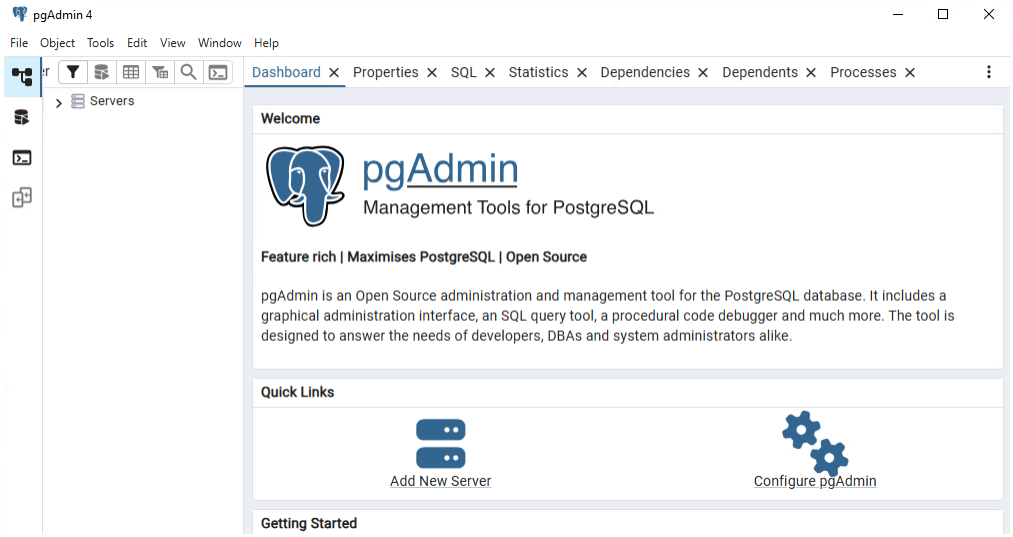
Notice how this is going into the **C:\Program Files** folder, which is correct for the 64-0bit ODBC Drivers. Click Next and Finish to complete the install process.

Since we ALSO need 32-bit ODBC drivers GSSync, go to your Users folder, locate the second install image that was downloaded, rename it to **edb_psqlodbc32.exe** and run it in the same way. (You can tell it is the right one because it installs to the **C:\Program Files (x86)** folder.) Again, use a default install as depicted above.

Create a New Database in Postgres

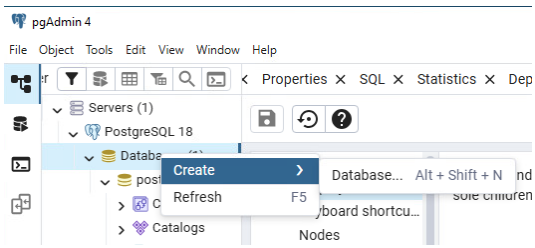
Now that we have everything installed, it is time to create the new database in Postgres. Note that database names are case sensitive, so now is the ideal time to have that discussion with anyone else using the system to select an appropriate name.

Launch the *pgAdmin* tool from the Start menu:

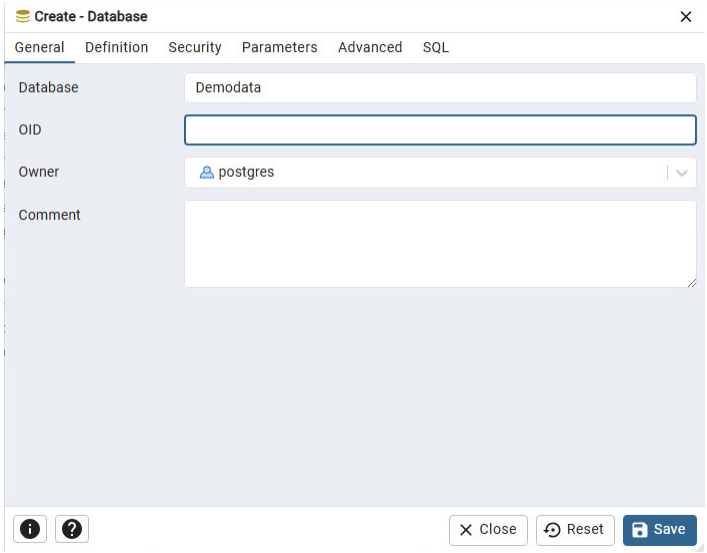


Open up the *Servers* item and provide the logon credentials (the password provided during installation):

Right-click on *Database*, then select *Create*, then *Database…*:



Specify the database name, along with any other details:



Click Save to create the new database! That was easy.

Export the Schema from Actian Zen

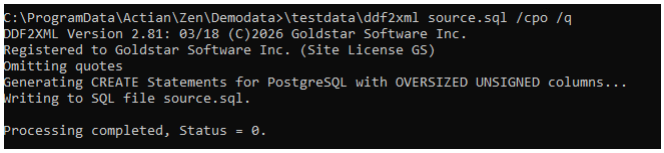
For the purposes of this discussion, we want to translate the data exactly as-is. To make this easier, we are going to use *DDF2XML* to create a schema file (a set of CREATE statements) from the Zen Database. Because *DDF2XML* is a command-line tool, you need to first download it from www.goldstarsoftware.com/tools.asp and apply your license. You can drop **DDF2XML.EXE** into your search path, or simply put it into your Zen database directory (where the DDF files reside). If you have v2 Metadata (i.e. PVFILE.DDF instead of FILE.DDF), then you need to stage **DDF2XML2.EXE** instead.

From a command line, run DDF2XML with the needed parameters. To create a schema for Postgres, you will want to use the /CP option. (If your database leverages large unsigned integer values, then considering using the /CPO switch instead, which will make sure that any of these fields are bumped to the next larger size to prevent value overruns.)

Also, it is important to note that most Postgres databases are case insensitive, but if you include quotes in the CREATE statements, then they will be case sensitive instead. If you want your Postgres tables and columns to be case insensitive, therefore, be sure to add the /Q option to DDF2XML to omit the extra quotes from the schema file. This will give you a command line like this:

```
DDF2XML Source.SQL /CPO /Q
```

When done, it should look something like this:



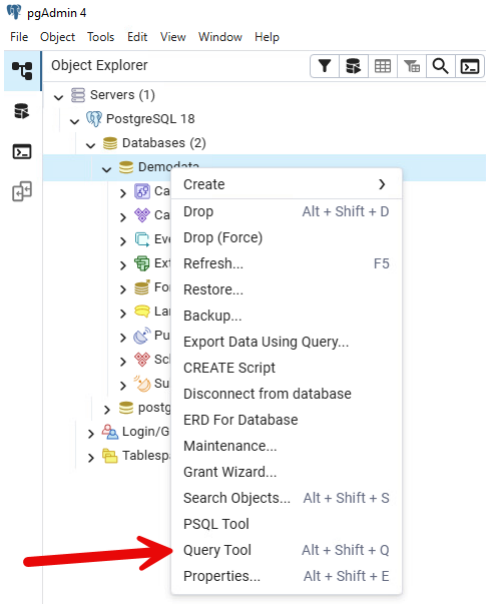
If you are running a trial of DDF2XML, then only 10 tables will be extracted for you, but if you have a licensed version, then everything should be there for you.

Review the schema file (**Source.SQL** in this case) with *Notepad* (or your favorite text editor) and determine if any changes are needed, as this is the ideal time to fix anything that is needed with respect to data types, indices, or other issues.

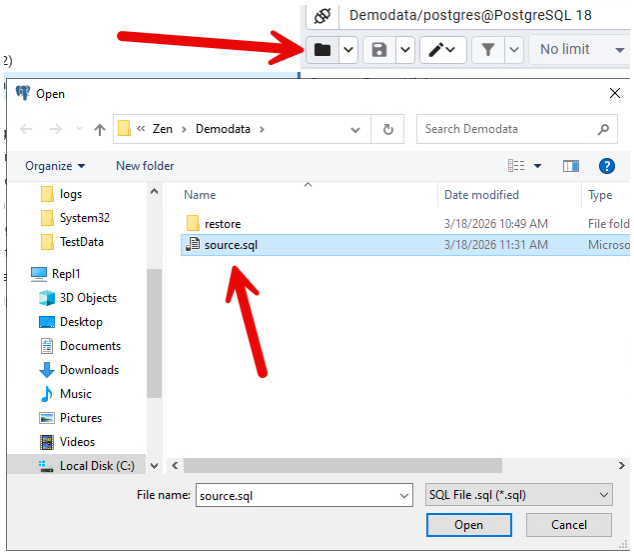
Create the Tables in Postgres

With the schema file generated, we are now going to use it to generate the Postgres tables.

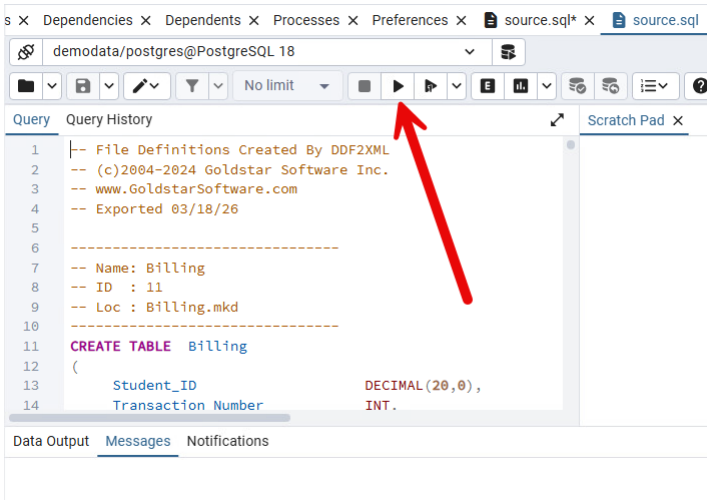
Go back to *psAdmin*, right-click on the database you created, and open the *Query Tool*:



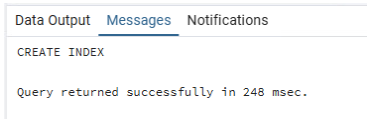
Click the *File Open* button and select the **SOURCE.SQL** file that was created by *DDF2XML*:



You will see your script opened up in the Query window:

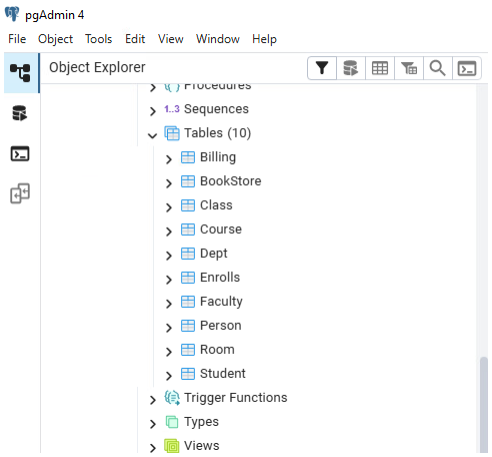


Click *Execute Script* to run all of the SQL statements and create the tables. If there are ANY errors in the script, then the entire process will fail. That is actually OK, as it allows you to fix the issues and re-run the entire script once again instead of having to do each file separately. Fix errors and click *Execute Script* again. Repeat this process until it completes without error:



If you experience a large number of issues specific to your data structures, contact Goldstar Software to see if the DDF2XML tool can be updated to address those issues for you.

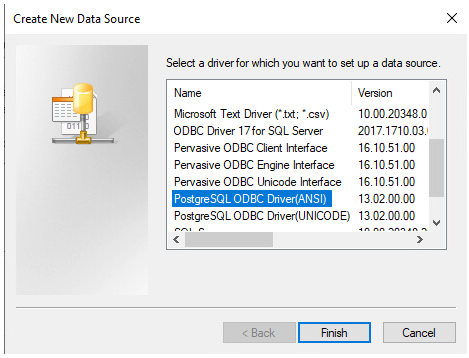
Once this is complete, go back to the *pgAdmin* tool and confirm that all tables now exist:



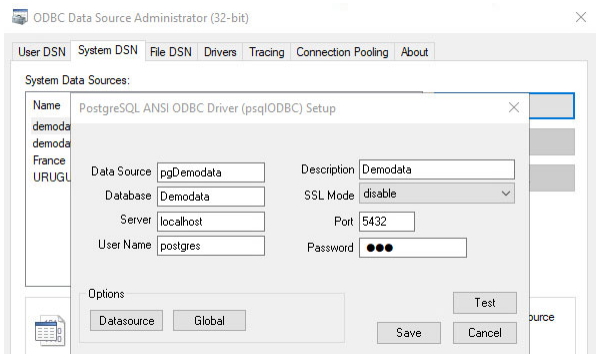
If you are missing anything, now is the time to fix it!

Create a New 32-bit ODBC DSN

Because *GSSync* is a 32-bit tool, you must create a 32-bit DSN for it to access the Postgres environment. Launch the *32-bit ODBC Administrator*, go to the *System DSN* tab, and click on the *Add* button.



Select the *Postgres ODBC Driver* and click *Finish* to get to the configuration screen:



Remember that the database name entered here is case sensitive.

The user name should be *postgres*, and the password is the one you entered at the time of the original installation.

Click *Test* to confirm a valid connection. If it fails, fix what is needed and test again. Once it connects properly, click *Save* to close out.

Migrate the Data

With all of the above steps completed, you should now have ODBC access to both the Actian Zen source database AND the Postgres target database. You can use just about any ODBC-compliant tool to move the data using simple select and insert operations.

However, we are going to use a faster option here to achieve a bit finer control over the data conversion process, namely the Goldstar Software tool called GSSync. This synchronization tool can be used for data replication as well, so if you need to replicate the data in stages, and then replication recent changes only, then GSSync will be your solution of choice.

The key to getting GSSync working is to set up a configuration file properly. At minimum, you need the following items, but you can certainly change others as needed by your environment. It is best to start with the default **GSSYNC.CFG** file and then the needed changes. While many of these items (except for the ODBC and sql query) can be included on the command line, that will only require extra typing later on. Start with these changes at minimum:

- Since you are pulling data from Zen directly, enter the named database here:

  ```
  <DatabaseName>Demodata</DatabaseName>
  ```

- As we are doing a FULL export only, set for no metadata:

  ```
  <MetadataType>NONE</MetadataType>
  ```

- Set the Export Type to ODBC:

  ```
  <ExportType>ODBC</ExportType>
  ```

- Set the ODBC Target Database to the DSN created above:

  ```
  <ExportODBCConnectString>DSN=pgDemodata</ExportODBCConnectString>
  ```

- Set the SQL INSERT statement as needed. If you are going to be replicating data, then the System Data value may come in handy. However, if this is a one-time data transfer and you don't need the system data value, then you can remove the {{SKEY}} field from the SQL statement. In that case, this is the SQL INSERT query that you need to use:

  ```
  <ExportODBCInsertQuery>INSERT INTO {{TNAM}} VALUES ( {{****}} );
  </ExportODBCInsertQuery>
  ```

- Postgres requires that dates, times, and timestamps be properly formatted, and it also requires using TRUE/FALSE for Boolean data types, so make sure that the following output formats are set as shown here:

```
<DateExportFormat>10</DateExportFormat>
<TimeExportFormat>10</TimeExportFormat>
<TimeStampExportFormat>10</TimeStampExportFormat>
<BooleanExportFormat>1</BooleanExportFormat>
```

You can now run GSSync for each table. If you want to do some test runs, then you can add the /OS command line option and generate a SQL table first, so that you can review the exact SQL statements being generated. (We recommend this the first time, at least.) A command line to generate SQL scripts would look like this:

```
GSSync /CFgssync.cfg /DTbilling /OSbilling.sql
GSSync /CFgssync.cfg /DTbookstore /OSbookstore.sql
etc.
```

After each run, check with the **GSSYNC.LOG** file to make sure that there were no errors.

You can then load the script files into *pgAdmin* and run each one separately, if you wish to have more fine-grained manual control of the import process. This may be helpful when first getting acquainted with this solution.

Once things look good enough to continue, you can now allow *GSSync* to dump the data directly to the Postgres database. Simply omit the /OS option from the command line, and GSSync will use the configuration file options to write directly to the target ODBC database. This looks like:

```
GSSync /CFgssync.cfg /DTBilling
GSSync /CFgssync.cfg /DTBookStore
etc.
```

Note that Table names in the /DT option are NOT case sensitive for Zen, but they ARE case sensitive for Postgres if quotes were used in the initial schema export! If you are getting errors from the direct ODBC INSERT statements, check this first.

The last step is to automate the entire extraction process into a single batch file that can run more unattended. Again, there are many ways to do this, but the Goldstar Software tools *FPrint* and *MakeBat* can make very short work of this work:

```
FPRINT * -n >TableNames.txt
MakeBat "GSSync /CFgssync.cfg /DT" <tablenames.txt >exportdata.bat
```

And then just run the resulting **ExportData.BAT** file to move the data!

When the data transfer is finished, go back to the *pgAdmin* tool and spot-check the table definitions and data records to validate that the data is there and complete. Check record counts as well, to

ensure that records didn't get lost along the way due to data conversion problems. If any discrepancies are found, delete the data from the target table, re-run that one table, and then review the GSSYNC.LOG file for specific details or errors and try to correct what needs to be fixed, then re-run it again. (If needed, add /LL5 to increase the log level on GSSync to its highest value, the debugging log.)

Finding More Help

If you need some additional hand-holding, or if you prefer to outsource this entire project, Goldstar Software may be able to provide you with more direct assistance. You can contact us at 1-708-647-7665 or via the web at http://www.goldstarsoftware.com.