

# DDF2XML v2.81 for Win32

## Program Description

DDF2XML provides a way to extract database definitions from data dictionary files (\*.DDF) in the v1 (old) metadata format. The first output format supported is an XML file (in the COBOLSCHEMAEXEC format). The second format is a Python structure string, useful for Python developers. The third option is to create a Btrieve file for the SQL table definition, which can be handy if you have DDF's but need to create the underlying database files for testing purposes.

The final output format supported by DDF2XML is a text file containing CREATE statements based on the data dictionary definition. These statements can optionally include the IN DICTIONARY clause, and they can be created for multiple engines -- Scalable SQL (from PSQL7 and older), the SRDE (PSQL2000i and newer), Actian's Vector, PostgreSQL, or Microsoft SQL Server. This is very useful for DDF modification, migrating a database structure to a new database environment, or for simply extracting the schema for an existing set of DDF's.

## Platform and Package

Win32, DDF Utility & Management Package; GSLic

## Pricing

\$250 Single-User; \$500 Site License (Eval Limited to 10 Tables)

## Command Line Syntax and Help Screen

DDF2XML Version 2.81: 03/18 (C)2026 Goldstar Software Inc.

Usage: DDF2XML OutFile [/MMasterPwd] [/C[Options]] [/B|P|S]Table [/D]

This program exports DDF data to an output file with several formats.  
Use /B to create a Btrieve file from the specified Table definition.  
Use /C to make CREATE statements from the DDF's.  
    Add 7 to the /C parameter to make statements compatible with PSQL7.  
    Add I to the /C parameter to make CREATE IN DICTIONARY statements.  
    Add U to the /C parameter to include USING clauses on INDEX.  
    Add L to the /C parameter to use LEGACY data types option (v12+).  
    Add # to the /C parameter to use hash marks as statement separators.  
    Add V to the /C parameter to make statements compatible with Vector.  
    Add M to the /C parameter to make statements compatible with MSSQL.  
    Add P to the /C parameter to make statements for PostgreSQL (BETA).  
    -Add O to the /CV-/CM-/CP param to Oversize all unsigned integer columns.  
    -Add S to the /CV-/CM-/CP param to add a new SystemData field to tables.  
Use /L# to specify the maximum length to use for BLOBs (Def=1000).  
Use /M to specify a Master password for the PSQL/Zen Dictionary.  
Use /P to display a Python structure string to the screen for the Table.  
Use /Q to omit quotes from table, column, and other names (for Postgres).  
Use /S to create a COBOL XML File for the specified Table.  
Use /D to enable DEBUG output.

## Examples and Sample Usage

To export unsecured DDF definitions into an XML-formatted file, use a command like this:

```
DDF2XML Output.XML
```

If security is enabled, and the Master password is "PASS", use this:

```
DDF2XML Output.XML /MPASS
```

To generate the CREATE statements for Tables, Views, Procedures, Triggers, Users, and Rights on a secured database, add the /P option and the Master password:

```
DDF2XML Creates.SQL /Mpassword /C
```

By default, this creates DDF's in a format compatible with PSQL 2000i and above. If you need DDF's for Pervasive.SQL 7 (Scalable SQL) or older, use the switch "/C7" instead.

By default, this process uses SRDE data types only. If you have PSQL v12 or above and want to use the LEGACYTYPESALLOWED setting to use the older legacy data types, then use the switch "/CL" instead.

By default, all SQL statements are separated with a semicolon (;). You can add the “#” to the /C option to change this to the pound sign instead (#).

To generate the CREATE statements to build a Vector replica of your database via GSSync, use the /CVS switch, like this:

```
DDF2XML VectorCreates.SQL /CVS
```

Likewise, the /CM switch can be used to generate CREATE statements with Microsoft SQL Server data types and /CP for PostgreSQL. Note that Postgres treats “quoted names” as case sensitive, so if you want case insensitive names in your target database, you should add the /Q option to omit quotes.

To generate the CREATE statements for Vector where the source data uses LONGVARCHAR fields, you can add the /L switch to force the VARCHAR length, which defaults to 1000, and you can add the O option to the /C switch to increase unsigned integers to the next size up:

```
DDF2XML VectorCreates.SQL /L2000 /CVSO
```

### Alternative Output Formats

To generate an Btrieve file from the SQL table definition, use the /B option, as shown in this example:

```
DDF2XML Person.MKD /BPerson
```

This command will read the table definition for the “Person” table, then create a file called Person.MKD based on that definition. If you omit the output file name, then the table name as used by the dictionary will be written. To avoid data loss, this function will return an error if the target file already exists.

To generate a Python structure definition (for pack/unpack) based on the data definition, use the /P option and supply the table name to process. The output will be displayed on the screen only, and not written to any output file. Note that not all data types can be expressed by Python, so you may find lots of “byte” definitions. Use the /Debug option to get a dump of the field names, data types, and lengths, too.

```
DDF2XML /PPerson /D
```

To generate an XML file suitable for use with COBOLSCHEMAEXEC or GSSync for the table “Person”, use the following command:

```
DDF2XML Output.XML /SPerson
```

## Other Information

DDF2XML is part of the DDF Utility & Management Package.

For more information on these utilities contact us at [www.goldstarsoftware.com](http://www.goldstarsoftware.com)

## Version History

Version 1.1: First documented version

Version 1.2: Added ability to create statements in the older PSQL7 syntax.

Version 1.22: Fixed formatting bugs in CREATE statements.

Version 2.0: Added GSLic capabilities; Added ability to create IN DICTIONARY statements.

Version 2.1: Added first-cut ability to export data in the proper format to be imported into the COBOLSCHEMAEXEC tool. Test data is required to complete adequate testing and fix any errors in this function. Increased ability to extract User and Rights information from newer databases. Fixed a few issues relating to PSQL7 export.

Version 2.2: Added length data to NOTE and LVAR field notes.

Version 2.3: Fixed Standards Issue on COBOLSCHEMAEXEC Export.

Version 2.4: Added support for new PSQLv10.20 data types.

Version 2.5: Updated handling of DEFAULT values.

Version 2.51: Added support for NCHAR/NVARCHAR fields for PSQLv12.

Version 2.52: Updated licensing code.

Version 2.53: Fixed issue with NCHAR fields. Changed default CREATE statement separator from <#> to <;>, and added <#> option for use with PSQL7. Added ability to display Index Numbers (new syntax for v12).

Version 2.60: Added ability to detect & handle PRIMARY KEY fields. Swapped order of DEFAULT and NOT NULL clauses to match standard PSQL syntax. Removed extraneous spacing.

Version 2.61: Added ability to generate CREATE scripts for a Vector database.

Version 2.62: Removed DEFAULT clauses from Vector output, added /B switch to control BLOB length in Vector output, and added O switch to increase Unsigned Integer fields to the next size up.

Version 2.70: Added first attempt at generating SQL Server data types.

Version 2.71: Fixed bug introduced with last updates.

Version 2.72: Added support for new data types.

Version 2.73: Added support for TIMESTAMP scale, as well as DEFAULT expressions (in addition to constants).

Version 2.74: Added start of the Python structure dump functionality.

Version 2.75: Added support for LegacyTypesAllowed.

Version 2.76: Added Btrieve File Create Option; Renamed /B to /L to make room for the new /B option.

Version 2.77: Added initial support for PostgreSQL formatting.

Version 2.78: Fixed bug in the /B option creating the wrong data types.

Version 2.79: Fixed length of NUMERICSTS and NUMERICSLs fields.

Version 2.80: Added support for masking fields and UNMASK rights, fixed precision on DECIMAL and MONEY fields.

Version 2.81: Improved support for Postgres target databases, and added the /Q option to omit quotes around table and column names.

## Known Problems

- ❑ When exporting the user and rights definitions on a secured database, it is NOT appropriate to export the passwords for each user. As such, each user will get a generic password of “password” when their GRANT LOGIN statement is generated. This is not a bug, but a security feature. You may need to edit the script file after the export completes.
- ❑ When generating the GRANT statements for tables with owner names, these owner names are NOT generated nor allowed for in the GRANT statements. You may need to modify the statements accordingly to supply the owner names. This also serves to close a security hole.
- ❑ Thorough testing of the /S option has not been completed, pending information from Actian on the COBOLSCHEMAEXEC tool. If you have a

sample table that is not working correctly, please send it to us and we will work with it to address issues.

- ❑ Full functionality for AUTOTIMESTAMP is not yet ready. Let us know if you need this, and we can address it at that time.
- ❑ There is a limit of 10 ACS definitions per file with this tool, but we can expand it if needed.
- ❑ This application works **ONLY** with older DDF's (V1 Metadata). For the newer V2 Metadata, please use DDF2XML2.