

DXUtil v2.70 for Win32

Program Description

A common task with DataExchange is the manipulation of files with system data. Actian PSQL/Zen provides a way to rebuild database files from one version to another while preserving the system data, and the Rebuild Utility handles this task quite well. However, if a file is damaged to the point that Rebuild cannot handle it, the user must often jump through a lot of hoops to rebuild the DataExchange environment around that file.

The DXUtil tool operates similarly to Actian's own BUTIL. A RECOVER operation will export the data (possibly in two directions) along with the system data values. A subsequent LOAD can then preserve the system data values in the new file, and the file can be placed in the original directory. This process can save a substantial amount of time when dealing with the repair of large corrupted files, as a majority of the data will NOT need to be resynchronized.

For this tool, an expanded version of the UNF file format was defined that includes system data. If the first character of a UNF record is "S", then the next 16 characters are expected to be the ASCII/hexadecimal value for the System Data value. This data is immediately followed by a comma, and then the regular UNF record structure follows. A fragment of a sample file is shown here:

S08C414E3EC1A736F,333,±u4♣	James	Ogelvie
S08C414E3EC1A7370,333,ûπ7♣	Kanagae	Yoko

DXUtil supports reading and writing both this expanded format as well as the original UNF format. In addition to the "S" value, files with UpdateSysData (a.k.a. System Data v2) will have a second 8-byte hex value starting with a "U" that represents the last Update Timestamp of the record in question. Of course, DXUtil can read any of the three UNF formats when loading data.

DXUtil also supports a forced UPDATE of all records in a file, eliminating the need for a separate tool like UPDTRECS. This forced update tells the engine that the records are changing (although they are not) so that all records are sent with the next replication cycle. Finally, DXUtil can also update the PDC table directly with the current timestamp (the FORCE option), cutting down the I/O requirements of this operation. Related operations, PURGE and PURGEALL, allow you to quickly clear a PDC table of deleted row tracking records, also called obituary records.

The DXUTIL tool also has a CLONE option which can clone a Btrieve file (similar to BUTIL) with two additional features. First, it supports cloning a file on top of itself. While potentially a dangerous operation that can destroy data, it can be particularly helpful when blanking PDC tables. Secondly, DXUTIL supports the /V option that allows you to specify the file format to be used for the new file.

Finally, DXUtil provides a number of special-purpose functions, designed explicitly for managing DataExchange in your environment.

Platform and Package

Win32; GSLic

Pricing

\$200 for Single-DX License, \$800 for LAN Site License

Download is available for 30-day full-featured trial.

Command Line Syntax and Help Screen

DXUtil Version 2.70: 07/18 (C)2025 Goldstar Software Inc.
Registered to Goldstar Software Inc. (Site License GS)

The command syntax is as follows:

```
DXUTIL -BACKUP <LicFile> [/options]
    Extracts your PSQL license from the Primary Server for backup purposes.
DXUTIL -CHECK <FileSpec> <DXFoldername>
    Checks FileSpec to verify all files are included in the DX database.
DXUTIL -CHECKSTATS <FileSpec> <PartnerDataFolder>
    Checks database file statistics from primary server against a partner folder.
DXUTIL -CLONE <NewFile> <SourceFile> [/options]
    Clones (Creates) a new file from the source MKDE file.
DXUTIL -COPY <SourceFile> <TargetFile> [/options]
    Copies records from the source MKDE file to the target.
DXUTIL -COPYBIND <SourceFile> <TargetFile> [/options]
    Copies the MiscData block (binding) from the source MKDE file to the target.
DXUTIL -DELETE <DataFile>
    Deletes a Btrieve file via the Btrieve Interface with NO confirmation.
DXUTIL -FORCE <PDC-Table> [/options]
    Forces new timestamps into the PDC Table to replicate all records.
DXUTIL -LOAD <UNFFile> <DataFile> [/options]
    Imports records and system data into the specified data file.
DXUTIL -PARSE <TimeStamp>
    Displays the timestamp value as a time/date value.
DXUTIL -PURGE <PDC-Table> [/options]
    Purges obituaries older than 30 days old from the PDC table.
DXUTIL -PURGEALL <PDC-Table> [/options]
    Purges ALL outstanding obituaries from the PDC table.
DXUTIL -RECOVER <DataFile> <UNFFile> [/options]
    Recovers the database records and system data in physical order.
DXUTIL -RENAME <DataFile> <NewFileName>
    Renames a Btrieve file via the Btrieve Interface.
DXUTIL -RESTORE <LicFile> [/options]
    Restores your PSQL license to the Backup Server after a failure.
DXUTIL -SAVE <DataFile> <UNFFile> <KeyNumber> [/options]
    Exports the database records and system data in the given key order.
DXUTIL -SAVEONE <DataFile> <UNFFile> <SysDataValueInHex> [/options]
    Exports ONE database record by its system data value.
DXUTIL -SHOWBIND <Filename> [/options]
    Displays the MiscData block (binding) from the given file.
DXUTIL -UNBIND <Filename> [/options]
    Removes the MiscData block (binding) from the given file.
DXUTIL -UPDATE <DataFile> [/options]
    Updates all records in the datafile to replicate all records.
DXUTIL -ZAPPDC <DataFile> <PDC-Table> <KeyNumber> <Value>
    Deletes a PDC Record for a given record in the primary file.
```

The following options are supported:

```
/O<owner> specifies the Btrieve file owner name.
/I Reads Btrieve files in the FORWARD direction.
/J Reads Btrieve files in the BACKWARD direction.
/V## (CLONE) Sets the File Version to 6, 7, 8, 9, 95, or 13 (-1=Existing).
/U (LOAD/COPY) will UPDATE existing recs with System Data (Def=InsertOnly).
/X Opens Btrieve files in EXCLUSIVE mode.
```

Examples and Sample Usage

```
DXUTIL -RECOVER PERSON.MKD PERSON.UNF /I /J
```

The above RECOVER command reads data records in physical order from the Btrieve file PERSON.MKD and writes them to an extended UNF file PERSON.UNF. As described above, an Extended UNF file may contain additional data fields for the System Key value (indicated with an S) and the UpdateSysKey value (indicated with a U) prior to the record length. Just like BUTIL, the /I and /J switches allow for forward and backward processing of the file. If both options are specified, backwards processing is ONLY done if an error is seen during the forward process.

```
DXUTIL -SAVE PERSON.MKD PERSON.UNF 1
```

The SAVE option is similar to the RECOVER option, but it takes the extra Key Number parameter. (The key must exist, or an error will be returned.) The /I and /J are also valid here, if needed.

```
DXUTIL -SAVEONE PERSON.MKD PERSON.UNF 08C414E3EC1A734E
```

The SAVEONE option is identical to the SAVE option, but it exports only one record – the one with the provided System Data Value. This can be used to extract a single record from a backup file after a data recovery.

```
DXUTIL -LOAD PERSON.UNF PERSONNEW.MKD /X
```

This LOAD command reads the PERSON.UNF file (which can be in the original or extended formats) and inserts the data to the data file PERSONNEW.MKD (which should be created as a clone of the original). The MKD file is opened in exclusive mode, so no applications can use the file until the load completes. Like BUTIL, it will skip any duplicate key errors. Records with no system data are processed with regular inserts, so it may be necessary to run DXSYNCTABLES on a file after such a repair, just to verify all records are present in the PDC.

The LOAD command also provides an additional function if System Data is available in the UNF file AND the data file. Using the /U option will tell DXUTIL to attempt to Update existing Btrieve records based on the system data value. This can allow a user to roll back to a valid backup, perform a data recovery on the original file using GSRECOVER /R or DXUTIL -RECOVER, and then load the recovered data on top of the backup file. The net result is that if a more current record was able to be recovered from the corrupted file, it will be reloaded on top of its original record, thus affecting any changes to the records correctly. Any files which are NOT in the original file are inserted, as expected. The only thing NOT handled is a record which was previously deleted, which will now suddenly reappear. This can be used in a case to recover MORE data than would be normally allowable through any other unique key (i.e. you'll capture changes that may have been missed by a BUTIL -LOAD), and it will also allow you to avoid creating duplicate records when no unique key exists on the file.

```
DXUTIL -COPY PERSON.MKD PERSONNEW.MKD
```

The COPY option is identical to the RECOVER and LOAD options running together, without the intervening UNF file. This is similar to BUTIL -COPY, but it supports the preservation of System Data. Add the /U option to update records based on the System Data value as described above.

```
DXUTIL -UPDATE PERSON.MKD /Osecret
```

This UPDATE command reads all records of the table PERSON.MKD, which is protected by an owner name of "secret", and then updates them with no changes. This refreshes the LastChanged timestamp in the PDC table and forces the entire table to be replicated on the next replication cycle.

```
DXUTIL -FORCE PDCPERSON.MKD
```

This FORCE command bypasses the table updates and touches the PDC table directly, updating the LastChanged field to the current timestamp. This also forces the entire table to be replicated on the next replication cycle, but it is faster than the FORCE option (since it requires half the number of I/O operations), making it ideal for large tables. The FORCE function should be followed by at least one update to the data file to set the dTblChg flag and ensure that the data gets replicated. (This may be incorporated in a future version.)

```
DXUTIL -PURGE PDCPERSON.MKD  
DXUTIL -PURGEALL PDCPERSON.MKD
```

This PURGE and PURGEALL commands allow DXUtil to run through the specified PDC table, deleting obituaries (i.e. deleted record markers) from the file. The PURGE function will delete obituaries over 30 days old, and PURGEALL

will delete ALL obituary records. It can be used to help clean up an environment taking lots of deletes that should not be replicated, or in a case where the obituary cleanup process has not run in some time. It can also be used after a file has been replaced or rebuilt and DXSyncTables has been run, in order to clear the obituaries from the PDC's on both sides to save disk space. The PURGEALL option should only be run on a quiet system when no actual deletes need to be captured, as it can delete the obituaries before the Sync process has seen them.

```
DXUTIL -CLONE PDCPERSON.MKD PDCPERSON.MKD /V8
```

This CLONE command creates a new 8.x PDCPERSON table on top of the existing, effectively blanking the table. This is superior to the BUTIL CLONE as you do not need to have a different source file when blanking the PDC. For the /V switch, valid values are 5, 6, 7, 8, 9, 95, or 13 to specify the needed file version. (Although 5.x files cannot be created on PSQlv9 and above, and although 5.x and 6.x files do not support System Data, these options have been added for completeness and for alternative uses.) Also valid is the value -1, which forces the new file to be the same file version as the original file. If the /V switch is left off (or is an invalid value), then the new file will be created in the default file format specified by the engines "Create File Version" setting.

The PARSE option is a separate troubleshooting tool that can take a septasecond Timestamp value and parse it out into the corresponding date value. You can provide either a decimal value or a hexadecimal value (preceded with "0x"). Note that earlier versions of the engine use an incorrect SystemKey value interpretation – it is off by one month. Actian has fixed this in the newer v15 engine, so newer files will use data that represent the timestamp correctly.

The BACKUP and RESTORE options are provided to allow you to follow the letter of the Pervasive License Agreement regarding the installation of license keys. Use the BACKUP option to create a flat file with your license data from the primary server. If the primary server ever fails, you can immediately load this data onto the secondary server (as allowed in the license agreement) using the RESTORE option. **Using this tool to violate your Pervasive license agreement is not allowed.**

The CHECK function is designed to be used by system administrators to verify that all database files are activated, which can be very useful after an application update which may have created new database files. This command must be run from your database directory, and it takes two parameters: a filespec indicating the files that should be checked, and a directory name, which should be your DX folder. When you run the command, a list of activated files is loaded from your DX folder's FILE.DDF file. Then, each file that matches the filespec is compared against this list, and any files that are NOT yet activated are reported to the output. Additionally, the application returns a status code equal to the number of files that were NOT activated, allowing you to write a simple batch file to verify your DX database.

The CHECKSTATS function is another check feature that compares the file statistics of the primary server files to the partner server database. CheckStats is designed to rapidly scan the data files on both servers and identify any files in which the file structure may have changed. Specifically, it looks at the number of indices, the record size, the presence of system data, the system data version, the system data length, the file version, and the record count. (Note that file version is a non-critical difference, and DX will frequently vary the record count on any production system. As such, the record count should only be used to indicate

problems when a system is taken off-line, a replication is run, and then the CheckStats is run, all with no users actively updating either database.) The Filespec provided on the command line can use wildcards, and *.* is valid in this case. DXUtil will only compare valid Btrieve files to the secondary server.

The ZAPPDC function is a special-purpose function that deletes the related PDC record for a given database key value. This serves to “hide” a given record from the database replication, and may be needed to allow replication of auto-increment key fields with a zero value, which is not typically allowed by the database. At this time, the key value *must* be a 4-byte integer. If sufficient demand exists, we may make this more flexible in a future release, so please contact us if you need something in particular.

To use the ZAPPDC function, you need to know the Btrieve Key Number of the file’s key, as well as the key value that you want to remove. These values are provided on the command line, along with both the original database file AND the PDC file name. DXUtil will then find the record in the base data file, locate it’s system data value, then locate the matching record in the PDC and delete it. Please note that there is NO confirmation dialog here – specifying the wrong filenames may result in a damaged file. Luckily, this can be “fixed” by running DXSYNCTABLES on the damaged file, so it is not fatal.

The RENAME function allows you to rename a file using the Btrieve interface. This can be used when you have multiple extension files for a given database table, or when you do not have access to the files at the operating system level.

The DELETE function allows you to delete a file using the Btrieve interface. This can be used when you have multiple extension files for a given database table, or when you do not have access to the files at the operating system level. Note that there is no confirmation on this operation.

The SHOWBIND function will display the contents of the MiscData block from the specified file. This block is populated when RI constraints or triggers are enabled on a file, which can potentially be lost in a DX deployment.

The UNBIND option displays the current MiscData block and then clears that block, essentially unbinding the file from the database. This may be needed when you need to move a file off of one server to another one. As binding is linked to the data file name itself, the proper process is to make a backup of the file, UNBIND the file, and after moving the unbound copy, put the backup file right back in place.

The COPYBIND option can be used to re-establish the MiscData block (i.e. RI Constraints and Trigger Flags) in a file by copying it from another file. Using a normal DX deployment, these blocks are lost, which will disable RI and triggers on the data files on the secondary server. You can use COPYBIND to read the binding block from the source server and write that to the target server to re-establish this functionality.

Other Information

DXUtil is a standalone tool.

For more information on these utilities contact us at www.goldstarsoftware.com

Version History

Version 1.0: First documented version in Win32.

Version 1.1: Added ability to parse timestamp values.

Version 1.2: Added PURGE option, completed FORCE option.

Version 1.3: Added /U option.

Version 2.0: Enabled Licensing and Trial capabilities.

Version 2.1: Added ability to Backup and Restore PSQL license data.

Version 2.2: Added the CHECK feature.

Version 2.21: Added Btrieve file detection to the CHECK feature.

Version 2.30: Added ZAPPDC function to clear out a PDC record for a single Btrieve record.

Version 2.31: Added reporting on Status 5 (duplicate keys) during LOAD function. Display record number and first 24 bytes of data block.

Version 2.40: Added CHECKSTATS function.

Version 2.41: Added filter to skip any records larger than 64129 bytes on a DXUTIL –LOAD command.

Version 2.42: Added –SAVEONE option.

Version 2.50: Added RENAME and DELETE functions.

Version 2.51: Changed FORCE option to use Local instead of UTC time.

Version 2.52: Removed check for System Data on the target file for the LOAD process. DXUtil now replaces BUTIL when UNF files have 0-byte records.

Version 2.53: Updated licensing code.

Version 2.54: Changed PURGE to only delete records older than 30 days, and added new PURGEALL function to delete all obituaries.

Version 2.55: Extended Owner Name support to 32 bytes, as well as hex.

Version 2.56: Added ability to force file format to v13 format on CLONE; Use BTRVEX call for CLONE option; added proper support for more than 4B records where needed; added support for 32-byte owner names.

Version 2.57: Fixed bug with CLONE on the v13.30 and above engine.

Version 2.58: Changed logic to allow some operations (Recover, Save) to continue even when there is no system data in the source database file.

Version 2.60: Added support for SysData2, as well as the new COPY operation. Allowed the PARSE function to work with hex values.

Version 2.61: Fixed 01/01 date conversion error in some years.

Version 2.62: Improved performance of PURGE/PURGEALL through using the Extended operations. Users on v14.20 and higher will use the DeleteExtended for even better performance.

Version 2.63: Fixed status 8 issue on PSQL v13.

Version 2.64: Fixed issue with UpdateSysData and v13 files.

Version 2.65: Updated functions LOAD, SAVE, SAVEONE, RECOVER, and COPY to work with ANY record size (up to 4GB) through chunking.

Version 2.70: Added functions SHOWBIND, COPYBIND and UNBIND to manage the MiscData blocks with RI or triggers are used.

Known Problems

The FORCE option currently updates ONLY the PDC file, and does not touch the dTblChg file. (The flags in this file are used to optimize replication by only checking the tables that have actually been updated since the last replication cycle.)

Until this is addressed, using the FORCE option should be accompanied by at least one manual update on the file in question. (WBEXEC32 works well for this.)