

# MoveDDF v2.10 for Win32

<b>Program Description</b>	<p>The MoveDDF tool reads information from one DDF set via the direct Btrieve interface, rennumbers the file ID and field ID values, and writes that information into a target DDF set. This is useful for renumbering DDF's that have reached the 65535 Field ID value (and the associated performance issues), for moving a single table definition from one DDF set to another, or for changing the metadata version of the database.</p>
<b>Platform and Package</b>	Win32, DDF Utility & Management Package; GSLic
<b>Pricing</b>	\$100 Single-User; \$250 Site License
<b>Command Line Syntax and Help Screen</b>	<p>MoveDDF Version 2.10: 06/10 (C)2016 Goldstar Software Inc.</p> <p>Usage: MOVEDDF TableName [/D]</p> <p>This command moves the TableName table information from one set of DDF's to another. The Destination set should reside in the current directory, while the Source DDF's should reside in the SOURCE subdir. Use "*" for TableName to process all Tables. Use the /D option to enable Debugging Output.</p> <p>WARNING: SOURCE DDF's are DESTROYED By this process! Have a backup!</p>
<b>Examples and Sample Usage</b>	<p>The command "MOVEDDF *" will read file definitions from the DDF's in the SOURCE subdirectory and write them into the DDF's in the current directory. The end result is that all table definitions can be easily moved to a new set of DDF's, which can be useful when repairing the DDF structure itself.</p> <p>Note that the records are actually recursively handled from the SOURCE DDF's, and they are deleted from the SOURCE as they are moved. This means that all work must be done on a backup copy of the SOURCE DDF's, as they are essentially destroyed in this process.</p> <p>The only DDF records currently processed are File, Field, Index, and Attrib. Data in any other DDF's relating to this table will NOT be transferred. This includes any views, user access rights, comments, relational integrity constraints, etc.</p> <p>As for Attrib, if the source file exists, then the destination ATTRIB.DDF file MUST exist. To prevent data loss, this is currently a fatal error and causes the program to terminate. If the source ATTRIB.DDF file does not exist but the destination DDF set has an ATTRIB.DDF file, this is not an error.</p>
<b>Other Information</b>	<p>MoveDDF is part of the DDF Utility &amp; Management Package.</p> <p>A DOS version is also available.</p> <p>For more information on these utilities contact us at <a href="http://www.goldstarsoftware.com">www.goldstarsoftware.com</a></p>
<b>Version History</b>	<p>Version 1.1: First documented version</p> <p>Version 2.0: Added GSLic capability.</p> <p>Version 2.01: Updated licensing code.</p> <p>Version 2.10: Added ability to handle either V1 or V2 metadata, including the ability to convert data from V1 to V2 and (in special cases) V2 to V1. Simplified and sped up the code.</p>
<b>Known Problems</b>	<p>Warning: The source DDF's are destroyed by the MOVEDDF process. You MUST have a backup of the source DDF's before beginning this process!</p>

Migration from V1 to V2 metadata is usually pretty clean. However, some internal field names (NN\_, PK\_, etc.) on very long field names may not be handled correctly and should be validated after the migration.

Migration from V2 to V1 metadata is fraught with disaster, as you lose both name space AND value space in the process. However, since file and field ID values are renumbered, this SHOULD work fine as long as you have fewer than 65535 fields in your database. However, there is NO special processing or error trapping on Table or Column names! If you migrate a field name with a length of more than 20 characters, it will simply be truncated. In some tables, this can result in duplicate field names – and thus return errors in the move process. You should only use this tool if you know for sure that all table and column names are under 20 characters.