

The UnTools

A Microsoft® Access Add-In
by Steve Tessler

Tools to help you Explore, Repair, and Improve your data structures and data

The UnTools (UnClear, UnField, UnPivot, UnList, UnMatch, UnCode, UnDaTe) and all associated documentation are Copyrighted
© 1999-2003 Steve Tessler. All Rights Reserved.

Email: UnTools@email.com
URL: <http://www.CleanDataSystems.com>

This file contains the text of the Help button for each tab of the UnTools.
The tool tab images here are low-quality to keep this file as small as possible.
Valid for version 1.01, build 53 2003-03-30

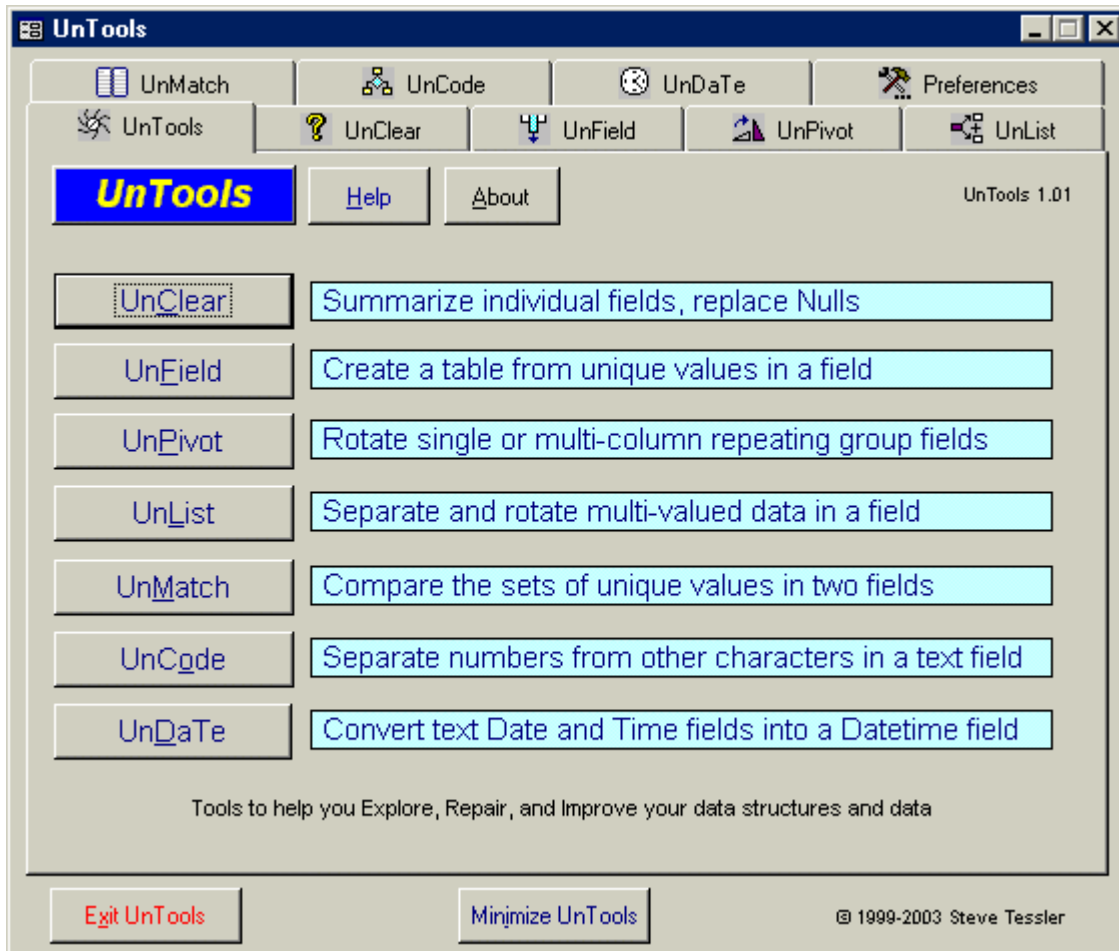
THE UNTOOLS	4
PURPOSE:	5
LICENSE:	5
REQUIREMENTS:	5
KNOWN ISSUES:	6
SUPPORT:	7
DISCLAIMER:	8
GENERAL UNTOOLS CONSIDERATIONS:	8
UNCLEAR	11
PURPOSE:	11
HOW TO USE UNCLEAR:	12
1. <i>Select Source:</i>	12
2. <i>Select Field to Summarize</i>	12
3. <i>View Summary of Field Contents</i>	12
4. <i>Replace Nulls</i>	13
5. <i>Include Zero-length strings?</i>	13
6. <i>Include just spaces (any #)?</i>	14
7. <i>Confirm</i>	14
<i>DO IT!</i>	14
UNFIELD	15
PURPOSE:	15
HOW TO USE UNFIELD:	16
1. <i>Select Source:</i>	16
2. <i>Select Field for new domain table:</i>	16
3. <i>Name the New Table:</i>	17
4. <i>Name the Content Field:</i>	17
5. <i>Make Duplicate Field?</i>	17
6. <i>Keep Null in Domain?</i>	17
7. <i>Add a Numbered PK (Primary Key) Field?</i>	18
<i>DO IT!</i>	18
USAGE NOTES:	18

UNPIVOT.....	20
PURPOSE:	20
HOW TO USE UNPIVOT:	21
1. <i>Select Source:</i>	21
2. <i>Select the first Field in the set of Fields you want to UnPivot:</i>	21
3. <i>GroupSize:</i>	22
4. <i>Name the New Table:</i>	22
5. <i>Name the Field to hold the column Labels:</i>	22
6. <i>Name the Field to hold the column Values:</i>	23
7. <i>Create GroupNum field?</i>	23
8. <i>Skip Null Records?</i>	24
<i>DO IT!</i>	24
REPEATING GROUP TERMINOLOGY USED FOR UNPIVOT:	25
** <i>Headers, Labels, Values, and Companions</i> **	25
** <i>GroupSize - Single vs. Multi-Field Repeating Groups</i> **	25
TROUBLE-SHOOTING TIPS	26
UNLIST	28
PURPOSE:	28
HOW TO USE UNLIST:.....	29
1. <i>Select Source:</i>	29
2. <i>Select Field containing multi-valued (list) data:</i>	29
3. <i>Define Type of List within the Field:</i>	29
4. <i>Name the new Table:</i>	30
5. <i>Name the new individual Item field:</i>	31
6. <i>Make 'Item Order' Field?</i>	31
7. <i>Trim separated Items?</i>	31
8. <i>Keep Null and Zero-Length String Items?</i>	32
<i>DO IT!</i>	32
USAGE NOTES:	33
UNMATCH.....	34
PURPOSE:	34
HOW TO USE UNMATCH:.....	35
1. <i>Select Source:</i>	35
2. <i>Select Source Matching Field:</i>	35
3. <i>Select Target:</i>	35
4. <i>Select Target Matching Field:</i>	36
5. <i>Merge the Fields?</i>	36
6. <i>Name the New Table (or accept default):</i>	36
<i>DO IT!</i>	37
USING THE OUTPUT TABLE:	37
UNCODE.....	39
PURPOSE:	39
HOW TO USE UNCODE:.....	40
1. <i>Select Source Table:</i>	40
2. <i>Select Field to UnCode:</i>	40
3. <i>Name the New Field to hold Number Part:</i>	40
4. <i>Name the New Field to hold Text Part:</i>	40
<i>Options</i>	40
<i>DO IT!</i>	42
USAGE NOTES:	42

UNDATE	44
PURPOSE:	44
DO YOU REALLY NEED TO USE UNDATE?.....	44
HOW TO USE UNDATE:	45
1. <i>Select Source Table:</i>	45
2. <i>Select text Field containing DATE string:</i>	45
3. <i>Date-only Option</i>	46
4. <i>Select text Field containing TIME string:</i>	46
5. <i>Name the new Field to hold Datetime values:</i>	46
6. <i>Skip Null 'Time' records during conversion?</i>	47
<i>DO IT!</i>	47
USAGE NOTES:	47
UNTOOLS PREFERENCES	49
DISPLAY NOTICES	49
INDIVIDUAL TOOL DEFAULTS	49
<i>UnClear</i>	50
<i>UnField</i>	50
<i>UnPivot</i>	50
<i>UnList</i>	51
<i>UnMatch</i>	52

The UnTools

UnTools@email.com
www.CleanDataSystems.com



COPYRIGHT NOTICE:

UnTools (UnClear, UnField, UnPivot, UnList, UnMatch, UnCode, UnDaTe, and all documentation)
© 1999-2003 Steve Tessler. All Rights Reserved.
Email: UnTools@email.com

Purpose
License
Requirements
Known Issues
Support
Disclaimer
General UnTools Considerations

Purpose:

The UnTools are 7 separate Tools in a Microsoft Access® Add-In that assist with exploring, cleaning up and normalizing data tables and data. They can be used alone or in various sequential combinations to do lots of work for you. Each UnTool is presented on its own tab and is like having a single-page Wizard.

** **UnClear** lets you view a summary of the contents of a field (frequency distribution) and optionally replace Nulls or empty string values with a new value of your own choosing.

** **UnField** creates a new table based on unique values in a field and optionally adds a primary key or a second copy of the selected field. It helps you make lookup and translation tables.

** **UnPivot** rotates a horizontal arrangement of related fields (a repeating group) from a flatfile table or query to a vertical structure in a new table; it rotates simple multi-field repeating groups, too.

** **UnList** takes a field containing delimited or fixed-width multi-valued (list) data and separates/rotates the items into a new table, optionally storing the original item position. It makes an itemized list of the original field contents.

** **UnMatch** compares unique values in two fields, showing matched and unmatched values and their record counts. It can also construct a 'Master List' of the values in the two fields.

** **UnCode** separates mixed alpha/numeric-characters within a single field (e.g., "<2.5") and places the results in two fields appended to the original table.

** **UnDaTe** converts Date and Time text strings stored in separate fields (e.g., "19990601" and "083000") into a single, true Datetime field appended to the original table.

License:

The comments here are not a substitute for the full License agreement. The complete License agreement can be viewed from the 'About' button on the main tab of the UnTools, and should also have accompanied your copy of the UnTools as a text file. Your installation or use of the UnTools constitutes your acceptance of the terms of the License. You assume all risks associated with the use or misuse of the UnTools on your system, files, and data.

The UnTools Add-In is provided to you as freeware. The UnTools and its documentation are copyrighted. You are not charged for your copy, can't charge anyone else for a copy, and no registration of any kind is required. Read the full license agreement for other information.

Requirements:

Minimum Requirements: You need Microsoft Access to use a version of the UnTools as an Add-In. There are separate versions for Access 97, Access 2000, and Access 2002 (XP). You also need at least one table containing data in your database. UnTools will reject attempts to be opened in read-only databases, or mdb's where no tables are present.

Performance: The UnTools have been developed and tested on a Pentium 133 with 32 or 96 MB RAM running Win95, a Pentium III 866 with 256 MB running Win Me, an Athlon 1700 with 256MB running Win Me, and are used regularly on Pentium 450's with 128 or 256 MB running NT4 and Pentium 4's up to

2GHz and 1 GB RAM. The UnTools definitely perform better on faster machines and with more memory, particularly for larger tables. I have used all of the UnTools on tables from 100K to 3.8M records, and larger more complex tables take longer to process; linked tables across a network can be slower, too. If you are able to quickly manipulate your routine table sizes on your machine using Access, the UnTools should work fine. If your complex queries run slow, simply plan what you are going to do and give the UnTools time to do what you want. On the P133 machine, creating an UnMatch table comparing fields with 600 vs. 205,000 records has taken up to five minutes while running other programs; it takes only a *fraction* of that time if I give it full access to processor time (turn off your MP3's for more speed!). Tables having only a few hundred records are processed nearly instantly.

Known Issues:

The following items are the only known problems with using the UnTools. Some limitations are built into each UnTool. Please see the Tools' own help for more information about correct input formats and how to use each UnTool. See the next section on Support about reporting bugs.

1) Even though Access will let you and the UnTools are very robust with respect to odd naming conventions, try to avoid wild names or excessive punctuation in table and field names. UnTools also may not process tables or fields having names longer than 60 characters. Odd behavior or failure of UnTools to perform certain tasks can mean that you have strange names for your database objects, and these can confound SQL or other internal operators. As a general rule of thumb, numerals are fine, but you shouldn't begin table or field names with them, CapitalizingWordParts can be very useful, and if you must have punctuation, readability is fine when using `only_an_underscore_between_parts`. You should know that MS Access is not fully SQL-92 compliant, and if you want your tables and fields to be accessible by other software (now or in the future) use good judgment in naming your database objects.

In case *you* never use bad names, here are some examples of things I've seen people put (or try to put) into field and table names (in addition to writing descriptions as names): various spaces; quotes and double-quotes; colons and semicolons; parentheses and brackets; slashes (/ and \); all sorts of higher ASCII characters (like ², ±, and others); asterisks (*); etc. Please use the table and field description areas for these sorts of things, keep the names you create simple yet meaningful, and don't report problems with your wild names as bugs in the program. ..thanks for listening..

2) If a table or query is open in Design View and then selected as the Source for an UnTool, you may get an error about the object being "in use by another user"--that user is you. Close the object and try again. If you still get the error even though you've closed it, Exit and reopen UnTools to release any lock it may have internally. Read the next issue, too.

3) Opening and closing the UnTools within your database *many times* without closing your own mdb may eventually eat up all of your available memory. The memory appears to clear when you close your mdb, but if you eat it up first you can freeze your machine. I am looking into this. In the meantime, open the UnTools once and if you think you might use it again before closing Access, minimize it (there is a button at the bottom center of the form to make it easy for you); bring it back up by double-clicking or using the Windows menu item in your database. You should close and make a named version copy of your own mdb periodically anyway, and that will help avoid this problem.

4) It may take a while for an UnTool to complete the task you give it, due to the speed of your machine, the size of the input, and its underlying complexity (i.e., complex query sources generally take longer). Please be patient.

5) If you have Nulls, zero-length strings, and values consisting only of various numbers of spaces in a text field, UnClear, UnField, UnPivot, and UnList will distinguish them if you allow Null data. UnMatch loses the distinction between these value types, and they may be blended together into a single value. This is

being investigated. In the meantime, before using UnMatch use UnClear to see if you have a combination of Nulls, zero-length strings, or space-only values in the fields.

6) Here's something I've experienced on a few machines when using UnTools or other add-ins. When passing data from Access to Excel using the 'Office Links/Analyze It with MS Excel' menu item, you may get an error from Excel about not being able to open UnTools because of the ldb locking file. Apparently Excel detects the open add-in and wants it, too. This happens with other open add-in's as well. It may be a machine setting. Just ignore the error in Excel by pressing Cancel.

7) Watch your disk space and keep an eye out for the 2GB limit if you're working on a large database and creating large output tables. Compact your database periodically AND KEEP SEPARATE VERSIONS ALONG THE WAY--TO RECOVER FROM MISTAKES.

Support:

Because the UnTools are provided to you without charge, the License should not imply and does not entitle any user to support for solving data problems using the UnTools. Once you understand the problem you are trying to solve and which UnTool to use, experiment and you might not need help anyway. That said, feel free to let me know if a tool is confusing to use or a different kind of tool would have worked better for you. I use the UnTools all the time and want to know of any bugs or other problems you have that are not covered in the Help button information.

You can contact me via email at: UnTools@email.com.
Check the UnTools website for updates at: www.CleanDataSystems.com.

To offer you built-in support, UnTools has error handling code to provide some feedback, so please check any messages for information to help you understand the problem. The UnTools were designed with messy data in mind, so it is likely that some users will experience an error as a result of using the wrong type or structure of data with a Tool. I've tried to trap many of those kinds of errors to guide you, but certainly haven't anticipated every situation. Please check to make sure that your data meet the criteria for using a Tool before reporting a problem. In particular, if you use a query source for an UnTool, make sure it really works before using it as input.

If you have difficulties with the UnTools that are not covered in the Help screens or resolved through the error messages, please email me. To report problems, first make sure that the error is repeatable, and then provide a description of what you are trying to do along with any error messages that were displayed (a screen capture would be great) , and the version of UnTools you are using (click the About button). DO NOT SEND ME YOUR DATA FILES, PLEASE!!

The UnTools were created to assist with correcting common problems found in data files. Each data set presents unique challenges, however, and usually requires special data handling techniques and multi-step strategies for repair. The UnTools have become part of my toolkit, but if they can't help you, sorry. If you have a 'special' data problem, send an email and describe it to me. I like those kinds of things.

I would also like to hear from anyone who finds the UnTools useful. Even if you used it just once to restructure your data for analysis (I assume many people fit that category). Let me know how things should work if you wished they worked differently. I only develop UnTools and other freeware in my mythical spare time, so a little encouragement wouldn't hurt either.

Disclaimer:

MAKE BACKUPS! AND DON'T FORGET TO MAKE BACKUPS!

CAUTION: Always WORK ON A *NEW* COPY of your data, NOT YOUR *ONLY* COPY of your data. You should ALWAYS BACKUP YOUR DATA before doing anything radical.

The UnTools are not warranted to be free of bugs. In fact, there is no warranty of any kind. Unlike the incredible diversity of invertebrates in nature, any 'bugs' in UnTools are not to be admired. I've tested and used the UnTools extensively, but can't anticipate all of the situations you may encounter with your own data, software and hardware. Please check the Help button within each Tool for some known (and planned) limitations. See the SUPPORT section above to report problems.

CAUTION: The UnTools are designed to help you manipulate and correct data and data structural problems. Because some UnTools allow you to write over tables or fields--essentially deleting what is being overwritten--be cautious and wise about how you use the UnTools. If you are unsure about overwriting data, create new tables and fields instead. I take no responsibility for your use or misuse of any of the UnTools. Practice good version control when either experimenting or actively reconstructing tables or fields. Make frequent copies of your work to ensure you can 'undo' something that proves to be wrong at a later time. Treat the UnTools as potentially dangerous power tools; don't hack at your data without a way to go back.

CAUTION: By default the UnTools should not change or delete any of your original data. You have that option in different situations, and may choose to do so. A warning message should appear if you try to overwrite a table (UnField, UnPivot, UnList, UnMatch) or field (UnCode, UnDaTe); UnClear requires specific confirmation before replacing (overwriting) Nulls. If you choose to overwrite existing data, be sure you understand the consequences of that action -- it is not reversible.

CAUTION: Plan what you are doing and why. The UnTools may prove very helpful in rearranging or decoding your flatfiles--or any data--but you can also create new problems if you haven't thoroughly evaluated your procedural steps during data conversions. That's when your backup database can save your butt; did I mention that you should always have a recent backup? Act recklessly with the copy.

General UnTools Considerations:

** UnTools will report a problem and then close (your database stays open) if you try to open the Add-In inside of a read-only database, or a database that has no tables. UnTools can't do anything for you in either of those situations.

** You can set your own UnTools default behavior for messages and naming conventions on the Preferences tab.

** All UnTools are used by first selecting a Source followed by one or more fields. System tables, tables starting with "~TMP", and queries starting with "~" are not displayed in the selection list, as these are prefixes associated with internal temporary and deleted objects. Empty tables and queries are useless to the UnTools and will be rejected.

** The combo boxes used to select a Source table, query, or field show only about 30 characters. You can double-click on any of the UnTools combo boxes to get the dropdown list to display at 2 times the control's width. The change is persistent for that control for the current UnTools session. When you reopen UnTools at a later time the default width is used. This feature was added to accommodate those users with very long table and query names, where differences between names are only apparent *after* 30 characters, so use it when you need it. (You may want to consider renaming to shorter names!)

** Numbers on the individual UnTool forms show you the best sequence to follow for filling in information when using any UnTool for the first time. Tabbing between the fields will follow that order. You can also click your way around the form to repeat or vary actions after changing an item (for example, to change the field you are working on).

** Most controls have tool tips associated with them to provide a little help as you go. Hold your mouse over a control to see the tip. The same information appears in the status bar as you tab or click through the controls on a form.

** Activity of an UnTool is indicated by showing a small message while a Tool is working; there may also be hourglass activity (or your theme equivalent) during some behind-the-scenes queries. Progress during operation of the UnTools modules can usually also be monitored in the status bar, where you can see progress bar activity (UnClear, UnPivot, UnList, and UnMatch) or numbers being counted off for records being processed (UnCode and UnDaTe). Where appropriate, a report appears when a Tool is done. Be patient when processing large tables; instead of thinking about how long it takes, think of all the work you didn't have to do by hand. Take a break.

** Input and output requirements and formats are specified and described within each module. Don't expect UnTools to work correctly with the wrong input.

** Some UnTools create a table, field, or query using a Source name plus a Prefix or Suffix designated on the Preferences tab. In Access, names of tables, fields, and queries are limited to 64 characters. Any names that will be longer than that after applying the preferred Prefix or Suffix will result in a warning message and return to the form without doing anything. Your options will be to rename the invalid item manually, adjust the preferred Prefix or Suffix to something smaller, or create a query on the Source where the items are renamed to something smaller. The best solution is to always limit the names of your items to 20-30 characters and use the Description property for any additional information needed to interpret the item.

** Using a Query as input to an UnTool.

Please make sure your query actually works before using it as input to an UnTool !!

Queries can be used as the Source for several of the UnTools. This is limited to only Select or Crosstab queries (action queries are excluded because they don't return records to work with). No record count for the Source is displayed for a selected query source, because the query would have to be run first to know how many records it returns. Also, when a field is selected from a query source, it may not be possible to determine the data type and display that on the form.

If your selected query source is complex--particularly Crosstab queries--there may be a delay before the dropdown list of fields appears after you click on it. This is due to Access having to first evaluate the query to determine the fields to display. Be patient.

Parameter queries -- those that require input from the user -- cannot be used with the UnTools. Although parameter queries will appear in the source selection list, they are rejected when selected and you are instructed to pick another source. Make a plain Select query with criteria for the subset you want to work with and use an UnTool on that.

If the table behind a query changes (e.g., renamed fields), your query may no longer work--prompting you for 'values' as if you had created a parameter query. Test your queries before using them with UnTools.

UnTools may not work with all Select or Crosstab queries--with the result that nothing happens when you run the Tool. Usually the cause is one of three things: (1) the query source depends upon other queries for *its* source (nested queries), (2) the field you selected is created as a concatenation of mixed data types, or (3) the field's values are calculated using functions (e.g., IIF) or custom routines. Many query sources

having these properties will work, but some may not. You are certainly encouraged to try using a query Source for an UnTool, as it shouldn't hurt anything if it fails (produce no output, or give an error). Minimal testing on queries has shown that first-level queries (those directly against tables) work quite well, but some nested queries (queries based on other queries) return an empty set. Symptoms of an UnTool not being able to work with a particular query include no fields to select or no output generated. I've been successful with simple Select queries nested up to six levels deep. If your query doesn't work, make a table from it and use UnTools on that.

Rarely, an UnTool operation will fail if using a Query source that contains complex IIF statements or uses other functions--an error is given indicating that the user 'cancelled' the operation. Again, if your query doesn't work, make a table from it and use UnTools on that.

** The UnTools handle *Linked* tables differently in some respects than resident tables. UnClear has no limitations with regard to linked tables; summarization and Null replacement work fine. UnField, UnPivot, UnList and UnMatch make new tables from your data, and these new tables are created in the mdb file where you are working, rather than in the source database for linked tables. UnCode and UnDaTe append new fields to the original table and won't work on linked tables (you will get a warning message if you try). If you want to use either UnCode or UnDaTe on those tables, do so within the database where those tables actually reside, or import the tables rather than link to them.

** Don't use an UnTool on a Memo field. The UnTools' activity with fields (parsing, rearranging, and recombining) were not designed to be used on Memo fields. Convert your Memo fields to Text fields before processing. Memo fields are typically rejected if chosen, with the exception of UnPivot which will process them as part of a repeating group but might also truncate values to 255 characters without saying it did. You were just warned about this, so don't tell me if it happens to you!

** The Default behavior of each UnTool form is to *not* overwrite existing tables, fields, or data.

UnField, UnPivot, UnList, UnMatch, UnCode and UnDaTe should not inadvertently replace existing fields or tables with new ones of the same name. These should be detected and give you some options. Be especially cautious about replacing any fields in your table with a new one of the same name; it's your choice. Just remember that this is not reversible.

There are good reasons why you might want to overwrite an existing table or field. You can experiment with different operations on converted fields, and keep the same field output name with each run on *different* input fields within a table to use the same queries or reports on different sets of converted results.

One of UnClear's purposes *is* to change your data--replacing Nulls with a new value--so be especially careful when you tell the tool to do so. You must confirm that you are aware of your Null replacement request; you must ask to do it. In any case, think not only about the change you are requesting, but also what the consequences of the change can be to the quality or integrity of your data. In particular, if you change Nulls to a value that already existed in the field, you will no longer be able to distinguish the newly changed values from the originals.

** COMPACT your database after any lengthy process of modifying tables, such as after using any of the UnTools on large tables. You can often get better performance from Access and a smaller mdb file, too.

Experiment, explore and have fun. I hope the UnTools save you time in your work.

UnClear

The screenshot shows the UnTools application window with the UnClear tool selected. The interface includes a menu bar with options like UnMatch, UnCode, UnDaTe, Preferences, UnTools, UnClear, UnField, UnPivot, and UnList. A help box at the top states: "UnClear lets you summarize the contents of individual fields, and optionally replace Nulls".

Source Information

Select Source: Table Query

1. [Dropdown menu]

Number of Records:

Select Field to Summarize:

2. [Dropdown menu]

Data type:

View Summary of Field Contents

Sort Order	Ascending	Value	# Recs
<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

3. [Using a Form] [Using a Query] [X]

Replace Nulls

Type the Value that you want to use to REPLACE NULL VALUES in the selected Field:

4. [Text input field]

5. Include zero-length strings? No Yes

6. Include just spaces (any #)? No Yes

Check YES to Confirm Your Replacement Request, then Click 'Do It'

7. Yes

[Do It!]

View the Summary again after Null Replacement to see the results

[Return to Main Menu] [Exit UnTools] [Minimize UnTools] © 1999-2003 Steve Tessler

Purpose:

UnClear has two functions. 1) UnClear can summarize a field from a table or query, providing a list of unique Values in a field, along with a record count and the percentage of all records. 2) UnClear can replace Nulls in a field with a Value of your choice, and replace zero-length strings or values consisting purely of one or more 'space' characters within Text fields. The name "UnClear" is used for this UnTool because you are either not sure (unclear) about the contents of a field, or you want to 'unclear' the 'clear' values represented by Nulls and empty strings.

You can test UnClear using the 'TestUnClear' table or any query in the demo database, UnToolsDemo.mdb.

How To Use UnClear:

*Preferences: You can set your own default summary query Prefix for UnClear on the Preferences tab.

Items below relate to numbered positions on the UnClear form.

1. Select Source:

Pick the option button for either a Table or Query as the type of source you want. This fills the combo box with the appropriate selections. Select your source from the dropdown list.

Tables: Linked tables can be used as the field source. A message is displayed if the link appears to be broken. Once a Table is selected, the number of records it holds is determined and displayed below the combo box. For very large or linked tables, it may take a moment for this to complete.

Queries: Only Select and Crosstab queries are allowed and displayed for selection. Queries that are also Parameter queries are rejected when selected. Because a Query must be run to know how many records it returns, "Not known" is displayed for the record count with a Query source. If you want to use a Query source, please test your query to make sure it works before using it as a source. Not all queries will work as input to an UnTool (see the general Help on the UnTools tab for more information about query limitations).

Null replacement in UnClear is disabled when using a query source because not all fields in all queries are 'updateable.'

2. Select Field to Summarize

A Field in the selected Source is chosen from the dropdown list. When a field is selected its data type is determined and displayed. The data type is not always able to be determined from some Query fields. For Text fields, the size setting is also displayed in parentheses (e.g., "Text(20)" for a 20 character Text field).

Memo, Hyperlink and OLEObject fields cannot be used with UnClear. UnClear groups values in a field to create the frequency summary, and these data types cannot be grouped. Replacing Nulls on these data types could also be problematic, so UnClear simply rejects fields with these data types.

3. View Summary of Field Contents

You can choose the initial sort order of the summary display, either Ascending or Descending by the Value or Frequency of records holding a particular value. Once the data are displayed, however, you can re-sort in any way you want. The default sort order is Ascending order by Value and shows any Nulls followed by zero-length strings on top. Another useful sort order is to use Descending by #Recs to show the most frequent Values first, and then sorted alphabetically within each frequency group.

The Field Summary can be viewed using a simple browsing Form or using a Query. You click the appropriate button for the view option you want.

Using a Form: The Form view option is good for quick evaluation of the field contents, and should be your usual selection. It has no overhead and there is nothing that needs to be done when you close the form window. The record count is shown formatted with commas as needed. The form always opens with a

default size and equal column widths, but is resizable for the viewing session. *Rarely, the Form view option will fail if a Query source is used that contains complex IIF statements or uses other functions—the form opens and then indicates the user ‘cancelled’ the operation. If this happens, use the Query view option instead.*

Using a Query: The Query option should only be used when you want to save the summary as a query that you can reuse and modify for your own purposes. However, the Query option has some caveats. The record count is an unformatted Integer. You can resize, sort, filter or manipulate the query results just as you would any other query. However, any manipulations will ultimately cause you to be prompted to 'Save' the query when you try to close it. While viewing, if the query is worth saving you can save it under a new name by using the "File/Save As" menu item. The default name is always constructed as "-- Summary of yourfieldname". By default, the query is NOT deleted when you are done viewing. To delete a Field Summary query when you are done viewing, click the enabled red 'X' button next to the "Using a Query" view option button on the UnClear form. If you do not delete the query using the red 'X' button while it is enabled (until you click or tab to something else in UnTools), you will have to delete it manually later. Of course, not deleting the query with the red 'X' button leaves the query in place for later use--which is the only advantage of using the Query option for the field summary view instead of the Form option.

Progress during operation of UnClear can be monitored by watching the status bar, where the query that creates the summary runs a progress meter. If you have a query source, the underlying query may also show its own progress meter prior to the query generating the summary.

The displayed summary will list unique Values and their record counts. It will also show the percent of the total records for each unique Value ("00.00%" is displayed if less than 0.01%). In both the Form and Query displays you can sort and filter by using the right-click menu.

4. Replace Nulls

Null and empty-string replacement is **only** available for fields in Tables, not Query sources.

Type the Value you want to use to replace Nulls (or empty-strings) in the selected Field.

UnClear will reject attempts to replace Nulls in AutoNumber, Boolean, Memo, Hyperlink, and OLEObject fields.

UnClear will evaluate the Null replacement Value and attempt to block requests that are an inappropriate data type for the field you selected. Note that some inappropriate data **can** be allowed in a field--but with unwanted results. For example, Access will allow "1,1" to be considered the Integer "11" (the comma is ignored). Watch your requests and results!!

UnClear will reject attempts to replace Nulls in a Text field with a new Value longer than the fields' Size setting.

5. Include Zero-length strings?

The Zero-length string option is available only for Text fields. If you need to replace **both** Nulls and any zero-length strings masquerading as Nulls, use the checkbox option provided on the form. The default choice is 'No' -- do not replace zero-length strings. Zero-length strings can look like Nulls in your data, but sort separately in the summary display. If your attempt at Null replacement fails, the entries are probably zero-length strings; check the option for zero-length strings and try again. Also check your summary display to detect 'spaces' in an otherwise empty-looking field. INTERESTING NOTE: Zero-length strings

can sneak into your tables during import, even if you have set the "Allow Zero Length" property to No for the receiving field. It has happened to me. UnClear distinguishes Nulls, zero-length strings, and spaces in its summary query and lets you check for them.

6. Include just spaces (any #)?

The 'spaces' option is available only for Text fields. If you need to replace values that consist of one or more 'space' characters masquerading as Nulls, use the checkbox option provided on the form. The default choice is 'No' -- do not replace 'space-only' string values. Space-only strings can look like Nulls in your data, but sort separately in the summary display, and each variation on the number of spaces is distinguishable because they have a different length (you can check them by clicking in the empty-looking value). If your attempt at Null and zero-length string replacement fails, the entries may be 'space-only' string values; check the 'spaces' option and try again. INTERESTING NOTE: If a text fields' Allow Zero-length string option is set to yes, you cannot directly enter only one or more space characters--they are immediately converted to a Zero-length string!

7. Confirm.

Replacing Nulls with a selected Value can be very useful. But it can also be dangerous if you haven't thought about the consequences. You must CONFIRM your desire to replace Nulls (or empty strings) by checking the box.

UnClear does NOT check to warn you if you are replacing Nulls with a Value that already exists in a field. You should always use the "View Summary of Field Contents" to evaluate a fields' contents *before* requesting Null replacement, and consider the consequences of your selection. If you 'UnClear' a Null by replacing it with a Value that already existed in the field, it is not reversible and you will not be able to distinguish the newly changed Values from those records that already had that Value.

DO IT!

Click the DO IT! button to replace the Nulls or empty strings.

UnClear will test to see whether any updates were actually made based on the Value you entered for Null replacement. If no replacements were made, a message appears explaining the situation (no Nulls to begin with, or wrong Value for the data type of that field).

* COMPACT your database after any process of modifying tables that makes or changes a lot of values, such as using UnClear to replace Nulls.

UnField

The screenshot shows the UnTools application window with the UnField dialog box open. The dialog has a title bar 'UnTools' and a menu bar with options: UnMatch, UnCode, UnDaTe, Preferences, UnTools, UnClear, UnField, UnPivot, and UnList. The UnField dialog is titled 'UnField' and has a 'Help' button. A light blue banner at the top of the dialog reads: 'UnField creates a new table from unique values in a field, optionally adding a duplicate or a primary key field'. The dialog is divided into two main sections: 'Source Information' and 'New Table Information'. In 'Source Information', there are radio buttons for 'Table' (selected) and 'Query'. Below this is a dropdown menu labeled '1.' and a 'Number of Records:' label. Another dropdown menu labeled '2.' is for 'Select Field for new domain table:'. Below that is a 'Data type:' label. In 'New Table Information', there is a text input field labeled '3.' for 'Name the New Table:'. Below that is another text input field labeled '4.' for 'Name the New Content Field:'. There are three sets of radio buttons: '5. Make Duplicate Field?' with 'No' (selected) and 'Yes'; '6. Keep Null in Domain?' with 'Keep' (selected) and 'Drop'; and '7. Add a Numbered PK (Primary Key) Field?' with 'No', 'Int', 'Long' (selected), and 'Auto'. Below these is a 'Name:' label and a text input field. At the bottom of the dialog are buttons for 'Return to Main Menu' and 'Do It!'. The footer of the application window contains 'Exit UnTools', 'Minimize UnTools', and '© 1999-2003 Steve Tessler'.

Purpose:

UnField has two functions. 1) UnField can quickly create a domain (lookup) table from a field in a table or query, providing the set of unique Values along with a numeric Primary Key field. 2) UnField can create a translation table by making two copies of the unique Values in the original field; one of the fields can then be altered and the table serves as a translation from the original set of unique Values to the new Values.

You can test UnField using the 'TestUnField' table or any query in the demo database, UnToolsDemo.mdb.

How To Use UnField:

*Preferences: You can set your own default prefix and suffix conventions for UnField on the Preferences tab.

Items below relate to numbered positions on the UnField form.

1. Select Source:

Pick the option button for either a Table or Query as the type of source you want. This fills the combo box with the appropriate selections. Select your source from the dropdown list.

Tables: Linked tables can be used as the field source, but the resulting new table will reside in the database (mdb) you are working in, rather than in the database where the source table resides. If you need to create the new table in the source, open the source database and do the work there. A message is displayed if the link appears to be broken. Once a Table is selected, the number of records it holds is determined and displayed below the combo box. For very large or linked tables, it may take a moment for this to complete.

Queries: Only Select and Crosstab queries are allowed and displayed for selection. Queries that are also Parameter queries are rejected when selected. Because a Query must be run to know how many records it returns, "Not known" is displayed for the record count with a Query source. If you want to use a Query source, please test your query to make sure it works before using it as a source. Not all queries will work as input to an UnTool (see the general Help on the UnTools tab for more information about query limitations).

2. Select Field for new domain table:

A Field in the selected Source is chosen from the dropdown list. When a field is selected its data type is determined and displayed. The data type is not always able to be determined from some Query fields. For Text fields, the size setting is also displayed in parentheses (e.g., "Text(20)" for a 20 character Text field).

Memo, Hyperlink and OLEObject fields cannot be used with UnField. UnField works by creating a 'GroupBy' query, and these data types cannot be grouped. UnField rejects their selection and posts a message.

The data type of the field you choose for UnField will be retained in the new table; i.e., Text is copied with the same Size, Boolean stays Boolean, Currency stays Currency, etc. A complete copy of all field properties is not attempted with this version of UnField, so it is possible that some special 'format' present in the original will be lost; correct that manually after the new table is created. Let me know if preserving all properties is really important to you.

** The name of the selected field is initially used as the root of names in the new table. You can override these defaults by simply typing whatever names you want in the boxes on the UnField form. Default names are automatically generated based on the main output field name for the domain. Some aspects of the naming can be user-defined on the Preferences tab of UnTools. If you change the preferences for UnField naming when you have already selected a source on the form, simply reselect the source field and the new preferences will be applied.

3. Name the New Table:

The New Table name is suggested as the source field name with a prefix attached. After the source field is selected, focus moves to the new table name field and the prefix is selected (highlighted) to allow you to immediately delete or replace it. The default prefix built into UnTools is "tlkp" but you can change it to whatever you want on the Preferences tab (Null is allowed for the preferred prefix, creating no prefix at all). You can also ignore the suggested New Table name and enter anything you like on the form.

4. Name the Content Field:

The Content field is the one that will contain the set of unique Values from the original field. The default name for the Content field is the same as your selected source field, which makes domain matching easy when linking both tables together. You can change the Content field name directly on the UnField form for each run. The Content field name becomes the root name for any other fields you request in the new table. If you change this name the other field names in the new table will also be changed.

5. Make Duplicate Field?

This option controls whether or not a second copy of the Content field is created in the output table. The Duplicate copy of the Content field can be used for making a translation table providing alternate terms for the Values in the Content field. The most common use for this option is for correcting spelling mistakes.

The default choice is 'No', which means only a single copy, will be made. If you choose 'Yes' then a second copy of the field is created. The name of the Duplicate field will automatically be created as the name you chose for the Content Field + a suffix affixed to the end. The default suffix is "2" but you can change it to whatever you like on the Preferences tab. The duplicate field suffix cannot be Null to ensure that the Content and Duplicate fields have different names. After the new table is created using UnField, you can change the name to whatever you like by opening the table in design mode.

6. Keep Null in Domain?

This option controls whether or not a Null in the original source field will be retained as a Value in the new Content field. The default choice is 'Yes.'

If no Null exists in the source field, this option doesn't do anything. However, if a Null exists in the source field, the option lets you Keep it (the default) as an entry in the new table, or Drop it from inclusion in the new table. If you have Nulls and you drop them from the domain table, keep in mind that you will no longer have a complete collection of every 'unique' entry in the original field within your new domain.

You should also seriously consider using UnClear to evaluate a field's Values before using UnField on it. If Nulls are going to be a concern, use UnClear to replace them with a suitable Value, and then use UnField to create the domain. That will ensure that you have a match between all records if you substitute the domain key for the original Value, and that you can create a join on the field Value to do so--because Nulls cannot be joined between tables. Later, you can clear the Value in the domain table to have the key refer to a Null--just like the original!

7. Add a Numbered PK (Primary Key) Field?

This option controls whether or not a numeric Primary Key field is also created in the new domain table. The four choices are:

'No' -- no PK field is created
'Int' -- the PK is created using an Integer data type
'Long' -- the PK is created using a LongInteger data type
'Auto' -- the PK is created using an AutoNumber field

The default choice is 'Long', meaning that a PK field is created using the LongInteger data type.

If you choose to create a Primary Key field, the default name is the same as you chose for the Content field + a suffix affixed to the end. You can also ignore the suggested name and type whatever you like directly on the form. The default PK suffix is "_ID" but you can change it to whatever you like on the Preferences tab. The PK field suffix cannot be Null, which ensures that the Content and PK fields have different names.

Regardless of the data type you choose, every item in your domain table will be assigned a unique number in the Primary Key field. The source field is sorted alphabetically in the new table, and numbered accordingly. Change any of the Primary Key properties later in table design mode.

DO IT!

Click the DO IT! button to create the new domain table to your specifications.

Other than the 'Working' message, there is nothing displayed to tell you that UnField is processing your field. The speed of assembling the list of unique Values is dependent upon the number of records to evaluate, but the process is rather fast, even on slow machines.

A report appears when UnField is done creating your table. You are then given an option to view the new table, which is simply opened like any other table. Change the design parameters of the table (field names, data types, indices) by going to design mode when the table contents are shown.

Usage Notes:

* By not making either a Duplicate or a Primary Key field, you simply end up with a table of the unique Values in the original source field.

* An error condition occurs if you try to create a new AutoNumber PK when you are using UnField to create a table from an existing AutoNumber field: you are only allowed one AutoNumber field per table in Access. You *can* create a new Primary Key as an Integer or LongInteger data type and still keep your original AutoNumber field--with its AutoNumber properties intact. Or, you can convert your existing AutoNumber field to a LongInteger before running it through UnField if you need to make a new AutoNumber Primary Key for your data.

* "Why doesn't UnField go one step further and create a relationship between the Source and the new domain table?" If you need to build a domain (lookup list) from a field, typically there are misspellings, miscoding and other errors that need to be evaluated and repaired before attempting to replace the original

values with a key from a lookup table. Since so much can go wrong by blindly linking tables, UnField leaves those other steps to you.

* A good way to resolve misspellings (like 'green' and 'greene') and non-uniform coding is to create a Translation table that sits between your original data and the 'list' table you really want to create. UnField can be used to make a table containing two copies of your original field (don't Add the key field when doing this). Correct one of the fields to have the uniform value for any variants you discover, thus resolving any errors. Now use UnField again to make your true domain table from the corrected field, this time adding a Primary Key field. A query can then be constructed to join your original field (with the errors) to the uncorrected column of the translation table, and join the corrected column of the translation table to the same values in the final lookup table you made. A Make Table query can then substitute the original 'values' in the source table with the Primary Key representing the uniform record from the final domain table, after which the translation table can be deleted and list maintenance can now be done directly on the corrected domain keyed to the original table.

* COMPACT your database after any lengthy process of modifying tables, such as using UnField.

UnPivot

UnTools

UnMatch UnCode UnDaTe Preferences

UnTools UnClear UnField UnPivot UnList

UnPivot Help

UnPivot rotates single or multi-column repeating group fields in wide 'flatfile' tables into a new, tall, narrow table

Source Information

Select Source: Table Query

1. [Dropdown]

Number of Records:

Select first Field in the set of Fields to UnPivot:

2. [Dropdown]

This field must represent the START of the UnPivot fields -- all the way to the end of the record. If necessary, make and use a Select query to get the correct field order.

Number of Fields in each UnPivot Group

3. GroupSize: 1 2 3 4 5 1 means single fields

New Table Information

Name the New Table: Use Defaults

4. [Text Box]

For the UnPivot field:

Name the Field to hold the column Labels:

5. [Text Box]

Name the Field to hold the column Values:

6. [Text Box]

For GroupSize > 1 'companion' fields are given the names found in the First UnPivot Group.

7. Create GroupNum field? No Yes

8. Skip Null Records? Yes No

Return to Main Menu Do It!

Exit UnTools Minimize UnTools

© 1999-2003 Steve Tessler

Purpose:

UnPivot is used to create a compact 'vertical' structure from an inefficient 'horizontal' flatfile structure found in many legacy data files. The structural problem is known as a 'repeating group.' Repeating groups are often apparent as a set of fields with a single root name and numbered suffixes (such as Child1, Child2, Child3 to record information about children of an employee). The problem with repeating groups is that the number of items needed is inflexible (only 3 children allowed in the above example) and it incorrectly models the one parent to many children relationship. The data are better stored with the repeating group information in a separate table, and 'stacked' one upon the other rather than set as adjacent fields. UnPivot moves and rotates the repeating group fields into a new table that can be related to the original, and provides a more flexible structure for both storage and query purposes.

Running UnPivot more than once on the same source table (say, through different queries) can normalize more than one 'field,' and help you to create sets of one-to-many relationships that model your data more accurately. Using a Crosstab query (or the Pivot tool in Excel), you can always recreate your original, horizontal structure at any time. Repairing repeating groups brings a table into first normal form, and is often the first step in repairing poorly structured data tables.

You can test UnPivot using the tables "TestUnPivot_1" and "TestUnpivot_3" in the demo database, UnToolsDemo.mdb.

UnPivot terminology: Header fields, Label fields, Value fields, Companion fields, and GroupSize. Please see the end of this UnPivot help for a description of repeating groups and the terminology used by UnPivot. It is placed at the end to keep out of the way once you are familiar with the terms and need only to refer to the basic UnPivot help at a later time.

There are also some trouble-shooting tips following the description of terminology used by UnPivot. These tips cover: new table created but contains zero records; using UnPivot on very wide source input (lots of repeating groups or very long original field names); caution about the size of your mdb file after using UnPivot; and, compacting after UnPivoting.

How To Use UnPivot:

Items below relate to numbered positions on the UnPivot form.

1. Select Source:

Pick the option button for either a Table or Query as the type of source you want. This fills the combo box with the appropriate selections. Select your source from the dropdown list.

Tables: Linked tables can be used as the field source, but the resulting new table will reside in the database (mdb) you are working in, rather than in the database where the source table resides. If you need to create the new table in the source, open the source database and do the work there. A message is displayed if the link appears to be broken. Once a Table is selected, the number of records it holds is determined and displayed below the combo box. For very large or linked tables, it may take a moment for this to complete.

Queries: Only Select and Crosstab queries are allowed and displayed for selection. Queries that are also Parameter queries are rejected when selected. Because a Query must be run to know how many records it returns, "Not known" is displayed for the record count with a Query source. If you want to use a Query source, please test your query to make sure it works before using it as a source. Not all queries will work as input to an UnTool (see the general Help on the UnTools tab for more information about query limitations).

2. Select the first Field in the set of Fields you want to UnPivot:

The list of fields in the chosen source are displayed in the dropdown box. The field you select here must represent the START of the UnPivot fields -- all the way to the end of the record. In other words, if you have two Header fields followed by the set of repeating group fields, you would select the third field--the first field representing the set of labeled repeating group fields. All fields before that one are considered the Header field(s). If necessary, make and use a Select query to get the correct field order before using UnPivot.

3. GroupSize:

Select the number of fields that identify each individual repeating group. If each field is an individual repeating group field on its own, then you would select a GroupSize of 1. If each repeating group is made up of a labeled repeating group field plus two Companion fields (additional fields that help qualify or describe the value in the repeating group field), you would select a GroupSize of 3. See the examples at the end of this help screen for more information about Companion fields and GroupSize.

A check is made to see if the full set of fields to be UnPivoted is evenly divisible by the selected GroupSize. If not, you get a message telling you about the problem. This might be due to an errant Header field, or one or two groups having a missing Companion field. If you are using a query source, fix the query. If you are working from a table, adjust your fields in a Select query (including 'creating' any missing, empty Companion fields) to meet your GroupSize requirement, and then run UnPivot on that.

UnPivot can handle a repeating GroupSize of 1 through 5. The upper limit is arbitrary, so let me know if you need a higher number (and tell me how your data got that way!). You will name the Label and Value fields for the resulting UnPivoted table, but any Companion fields (GroupSize > 1) are simply given the field names from the first UnPivot group (the first one encountered after the Header fields). You can easily rename the Companion fields by opening the UnPivot output table in design mode.

4. Name the New Table:

** Note the checkbox adjacent to this item. By checking the box you can avoid having to create names at form positions 4, 5, and 6 on your own and have the default names from the Preferences tab applied automatically. By leaving the box unchecked you will have to enter names for this and the next two fields. Unchecking the box if it is already checked will clear the values. You can override these defaults by simply typing whatever names you want in the boxes on the UnPivot form.

UnPivot creates a new table to hold the results of its activities. If you checked the "Use Defaults" checkbox, the name is created by using the prefix defined on the Preferences tab plus the source table/query name; the default prefix is "unp".

If you enter a table name that already exists, you are prompted to Append, Replace, or Cancel back to the UnPivot form to enter a different name. If you choose to Replace the existing table, it will be deleted and a new one made; **THIS IS NOT REVERSIBLE**. If you choose to Append, all fields being appended must match exactly (names and data types) or no records will be added.

Appending can be very useful if you have multiple sources (e.g., text imports) that all need to be UnPivoted into the same table. All fields for the output (e.g., Headers and Companions) must match exactly for appending; and keep the same output table and field names.

5. Name the Field to hold the column Labels:

The name you enter here will hold the original column Labels from the UnPivot field in each repeating group. If you checked the "Use Defaults" checkbox, the name defined on the Preferences tab will be entered for you; the default name is "unpLabels".

The Label field is created as a Text data type set to the size of the largest original UnPivot field Label found. If you want to change the Label field properties to something else, open the new table in design mode after UnPivot creates it and adjust as you see fit

6. Name the Field to hold the column Values:

The name you enter here will hold the Values from the UnPivot field in each repeating group. If you checked the "Use Defaults" checkbox, the name defined on the Preferences tab will be entered for you; the default name is "unpValues".

The Value field created after UnPivoting is set as a Text(255) field. This is because the original set of columns being UnPivoted may vary in how their data types are set, despite supposedly identical kinds of data (this is common when importing data from spreadsheets). Populating the Value field as a Text data type to maximum size is a safe default for UnPivot since data of any type can usually be converted intact. Memo fields will also be processed if they are found within the group of UnPivot fields, but these will be truncated (left-most characters retained) to 255 characters if the string was longer.

If you want to change the Value field data type to something else, open the new table in design mode after UnPivot creates it and adjust as you see fit. This is especially important for number Values as you won't be able to do any math on the Text field in the UnPivoted data. Any errors during your data type conversion on the new table reveal inconsistencies in the original fields' data, and thus can be useful in doing some basic quality assurance of the original data files.

** Note that Companion fields take their names from the First UnPivot group and will need to be renamed in the output table as necessary. They are also set as Text(255) data type/size for the same reason the Value field is created that way. I do not currently have any plans to add the ability to individually name a variable number of Companion fields from the UnPivot form, or to resolve differences in the properties of the Companion fields from the set of repeating groups. Text(255) is a safe bet for holding companion data, and retaining the names from the first UnPivot group is usually sufficient for quickly renaming the Companion fields after the output table is created. You can change any of the Companion fields' properties after making the output table.

7. Create GroupNum field?

This option controls whether or not UnPivot creates a GroupNumber field in the output table. The name of that field is set in the Preferences (not on the UnPivot tab), and the UnTools default is set to the field name "GroupNum".

The default for this option is 'No' (don't make the GroupNumber field). When 'Yes' is selected for the GroupNumber option, UnPivot adds a single field to the output table after the header fields, but before the UnPivoted fields. The value in the GroupNumber field is an Integer identifying the original field order of the Group (column, or set of columns for GroupSize > 1) that was UnPivoted. This option is useful if, for example, the sort order of the original UnPivot fields implies some intrinsic order for the data, and you want to capture that information to use later--especially if the original field names (now Label values) will not sort in a way that restores the original order.

The GroupNumber field is, however, redundant with the Label field--each GroupNumber is associated with only one Label value because the original field names themselves must have been unique. If your ultimate goal is to make a lookup table from the Label field--and the original sort order is irrelevant--don't bother making the GroupNumber field, and instead use UnField on the Label field after UnPivoting.

8. Skip Null Records?

This option controls whether or not UnPivot creates new records in the output table where the original Value in an UnPivot field was Null. The presence of a Null in an UnPivot field usually indicates the absence of data for that particular record for that particular Label. By skipping originally Null values in the repeating groups you can create a more compact output table.

The default for this option is 'Yes' (skip Nulls) and will result in no new record being created when the original Value and the contents of all Companion fields for a record were Null. By choosing 'No' (do not skip Null values) you will create an UnPivoted output table containing all original Values--including Nulls.

* Use the Skip Null option when you can. Consider that UnPivoting a table will result in:

Number of new records = (# of original records) * (# of UnPivot Groups).

For very large tables with lots of UnPivot fields, this can result in an extremely large result table. If you are UnPivoting the largest table in your database, it is not unusual for your mdb file to double in size (think about it). If you are only interested in cells from the source table that are Not Null, using the Skip Null option will result in much smaller tables. For source tables that have a GroupSize > 1 (has Companion fields), the Skip Null option will only drop records in which *ALL* of the fields are Null; that is, if the main UnPivot field is Null but any of the Companion fields have data for that record, the record is retained. This is the most natural behavior, so that UnPivot does not lose any information that was originally present in the columns. You can always create a Delete query after UnPivoting to drop records where the main UnPivot field contains a Null.

* Don't use the Skip Null option when you need to retain all of the original information.

If an entire UnPivot field (or set of fields for GroupSize > 1) contains only Nulls (has no data), using the Skip Null option will skip every record and therefore 'skip' the entire column. No trace of the original column will appear in the output table. If you want to retain records that include the Label values for that field, check 'No' for the Skip Null option. If a Null value has meaning in your data, you would also want to say 'No' to the Skip Null option. Note that skipping Null values does not interfere with recreating your original 'table' using a Crosstab query or the Pivot tool in Excel, where any cell that did not have a corresponding record will default to displaying nothing (Null). Entire original fields that were full of Nulls and dropped, however, will not reappear.

DO IT!

Click the DO IT! button to create the new UnPivot table to your specifications.

Progress during operation of UnPivot can be monitored by watching the status bar, where each UnPivot field/group will run a progress meter. The speed depends upon the size of the data file and, if using a query as the source, the complexity of the underlying query.

* If you run UnPivot against a complex Select/Crosstab query that involves lots of joins, the process may be slow--but it will complete eventually. That's because UnPivot handles each UnPivot field individually and the source query must be run once for each repeating group. To estimate the time UnPivot might need for your data, multiply the time it takes to run the source query *once* by the number of UnPivot columns and you should be close to the time it will take. Go take a walk. It's good to get away from the computer once in a while--especially if it's doing work for you!

A report appears when UnPivot is done creating your table. You are then given an option to view the new table, which is simply opened like any other table. Change the design parameters of the table (field names,

data types, indices) by going to design mode when the table contents are shown. If you had repeating groups with a GroupSize > 1, then this is a good time to go to design mode and change the Companion field names from those taken from the first UnPivot group to something more generic (for example, from "Units1" to just "Units").

***Reward for reading this far: you can even UnPivot a single field, changing its name to a Label associated with each original Value! An Update query can then be created to selectively change the Labels based on the Values or a range of Values. Pretty cool, huh!

=====

Repeating Group Terminology Used For UnPivot:

**** Headers, Labels, Values, and Companions ****

UnPivot considers each set of input fields to consist of 0 or more *Header* fields followed by 1 or more fields in a repeating group. The Header fields are those fields that serve to identify a data row, and are usually either a primary key field or a set of fields that could serve as the primary key for the table. To use UnPivot you will need to arrange your data so that, from left to right, your Header fields precede your repeating group data; the examples below all use that arrangement.

The notation used here is that Header field(s) are enclosed in square brackets [...] and each group in the set of repeating groups is enclosed in curly brackets {...}, like this:

[Header fields], {Group1}, {Group2}, {Group3}, etc.

Below are two examples of typical repeating group layouts, where the first has one Header field and four repeating group fields, and the second has three Header fields and two multi-field repeating groups.

Example 1:

[Employee_ID], {Child1}, {Child2}, {Child3}, {Child4}

Example 2:

[Site, DateTime, Agency], {Chemical_1, Flag_1}, {Chemical_2, Flag_2}

A generic approach to describing the repeating group structure is as a Header plus Label fields (the repeating group) and, if they have them, the Labels' Companion fields. The second example above could be represented as:

Example 2b:

[Header1, Header2, Header3], {Label_1, Companion_1}, {Label_2, Companion_2}

**** GroupSize - Single vs. Multi-Field Repeating Groups ****

The typical repeating group structure is simply a set of individual columns, such as:

[Header],{Label_1},{Label_2},{Label_3},etc.

where each Label column contains Values associated with the item defined by the column name. This is defined within UnPivot as a GroupSize of 1. Each single field in the repeating group is a complete group. UnPivoting the example structure could produce a three-field result table:

[Header], Label, Value.

In the new table each Header-defined record has multiple records--one for each Label and its Value. The new Label field contains the names of the original columns (the Labels "Label_1", "Label_2", etc.) and the Value field contains the data that was stored in each Label column. The Label field contains the different field names from each of the UnPivoted columns which now 'label' the data in the Value field.

A multi-field repeating group might have the following structure:

[Header],{Label_1, Unit_1, Flag_1},{Label_2, Unit_2, Flag_2},{Label_3, Unit_3, Flag_3}, etc.

where each repeating group contains Values for specific Labels as before. In addition, each Labeled Value is accompanied by a consistent set of *Companion* fields that help describe it. In this example, the data Group is actually constructed as a labeled Value, a Unit, and a value Flag. UnPivot needs to recognize a GroupSize of 3 to correctly rotate this structure. Three adjacent fields (a Label and two Companions) make up a complete Group. UnPivoting this structure could produce:

[Header], Label, Value, Unit, Flag.

For each Header, there are multiple Labeled Values just as was constructed for the GroupSize = 1 example above, but each UnPivoted field is also accompanied by its original Unit and Flag data to help describe the *individual* Values. Nothing in the Companion fields is rotated; the values in those fields simply accompany the rotated column Label and Value fields into the new table. UnPivot can rotate structures up to a GroupSize of 5 (a Labeled Value field and 4 Companion fields).

=====

Trouble-shooting Tips

* Problems with UnPivot making a new table with ZERO records are probably due to bad NAMES.

A symptom of bad table or field names when using UnPivot is that UnPivot will appear to run without incident, but no data will be transferred to the 'new' table being created. A more rare symptom is where the user is prompted for the value to use for the funky field--an unexpected 'parameter' query where you can often see the bad name in the dialog prompt. If you get zero records output to the UnPivot table created, try running UnPivot on a simple Select query that renames your fields to see which ones were causing the problem.

* UnPivoting very wide source tables.

For very "wide" source files (dozens of header fields and/or very long field names), UnPivot may have problems because of a string size limit. SQL strings are dynamically created by UnPivot and in certain circumstances the maximum size for the SQL statement is exceeded. If the character limit is exceeded, the SQL phrase will not run.

The solution is as follows, and is useful even if you don't have the length problem:

a. Be sure to have a key field (e.g., ID) used as a unique row identifier in your Source table.

- b. Use a Select query to gather together only the key (ID) field *plus* the UnPivot fields.
- c. Run UnPivot on the query, which has only a single short-named header field (e.g., ID) and is handling only one UnPivot column with it at a time.

This is a generally good strategy anyway, because UnPivot is essentially creating a 'many' table relative to the 'one' table containing the header fields. If you create an ID field reflecting the original row source, after UnPivoting you can consider dropping the UnPivoted fields from the source table and do the correct one-to-many join on the result of the UnPivot using only the row-identifying ID key.

* Beware of the size of both the tables and the database itself (.mdb file) when UnPivoting very large tables.

I have successfully used UnPivot to 'rotate' tables with >60,000 original records consisting of a 12-field header plus 215 UnPivot fields! (Yes, someone actually handed me a 'table' like that!) Don't be impatient, because converting files like those *does* take some time. You can see that UnPivot is working by noting the progress indicators in the progress bar -- one query is run per column being UnPivoted, and you can watch them all get handled. Of course, my experience with smaller tables (<100 records and up to 70 UnPivot fields) is that UnPivot will convert them in seconds (on my Pentium 133), so be kind and patient rather than annoyed with the wait. Consider the time it would take to construct and run 70 append queries by hand or to use an awkward Union query to do the same thing.

* You will probably want to COMPACT your database after running UnPivot.

If the UnPivot operation was successful, consider evaluating your tables and the desired results before dropping the UnPivoted fields from your Source table. Then, compact the database. With careful planning you can end up with a much more flexible table/field structure in a database file only slightly larger than that with which you started -- or even much smaller if you had lots of Nulls that were dropped while UnPivoting. Recover the space the original fields occupied by compacting the database.

UnList

UnList Help

UnList transforms multi-valued data within a single field into an itemized list table, optionally tracking item order

Source Information

Select Source: Table Query

1. [Dropdown]

Number of Records: [Text Box]

Select Field containing multi-valued (list) data:

2. [Dropdown]

Data type: [Dropdown]

3. Define Type of List within the Field

Delimited: Comma Semicolon
 Space Other /

Fixed Width: # Characters/Item [Text Box: 1]

New Table Information

Name the new Table: [Dropdown: 4.]

Name the new individual Item Field: [Text Box: 5.]

7. Make "Item Order" Field? Yes No
Name: [Text Box]

8. Trim separated Items? Yes No

9. Keep Null and Zero-Length String Items?
 Keep Drop

Return to Main Menu Do It!

Exit UnTools Minimize UnTools © 1999-2003 Steve Tessler

Purpose:

UnList is used to separate multi-valued data that are stored as a 'list' in a single field. Two types of lists are recognized: delimited items, such as "red, blue, green"; and, fixed-width items, such as "YYNNYYN". These are 'repeating groups' that are embedded within a single field, and are difficult to examine for individual item values--or their sequence--in the contents. UnList creates a new table containing a copy of each unique multi-valued data string with a record for each separated item, thereby changing the multi-valued data into an orderly, itemized list. An optional Item Order field can store the order position of each item in the original list. This new table can be joined on the multi-valued data string to the original table, thus providing the source table with access to each item and it's position in the original string.

**Note: See end of UnList Help for instructions on how to join the UnList results back to your source table.

You can test UnList using the table 'TestUnList' in the demo database, UnToolsDemo.mdb.

How To Use UnList:

*Preferences: You can set your own defaults for naming and other variables used by UnList on the Preferences tab.

Items below relate to numbered positions on the UnList form.

1. Select Source:

Pick the option button for either a Table or Query as the type of source you want. This fills the combo box with the appropriate selections. Select your source from the dropdown list.

Tables: Linked tables can be used as the field source, but the resulting new table will reside in the database (mdb) you are working in, rather than in the database where the source table resides. If you need to create the new table in the source, open the source database and do the work there. A message is displayed if the link appears to be broken. Once a Table is selected, the number of records it holds is determined and displayed below the combo box. For very large or linked tables, it may take a moment for this to complete.

Queries: Only Select and Crosstab queries are allowed and displayed for selection. Queries that are also Parameter queries are rejected when selected. Because a Query must be run to know how many records it returns, "Not known" is displayed for the record count with a Query source. If you want to use a Query source, please test your query to make sure it works before using it as a source. Not all queries will work as input to an UnTool (see the general Help on the UnTools tab for more information about query limitations).

2. Select Field containing multi-valued (list) data:

A Field in the selected Source is chosen from the dropdown list. When a field is selected its data type is determined and displayed. The data type is not always able to be determined from some Query fields. For Text fields, the size setting is also displayed in parentheses (e.g., "Text(20)" for a 20 character Text field). This field will be copied to the output table with the same data type and size settings.

** The name of the selected field is used as the root of the new table and field names. Default names are assigned automatically after the source field is selected. You can set your own default prefix and suffixes on the Preferences tab. If you change the preferences for UnList naming when you have already selected a source on the form, simply reselect the source field and the new preferences will be applied.

UnList is generally used with Text data type fields, although you may also want to use it to separate digits from numeric fields. Memo, Boolean, Hyperlink, Byte, and GUID fields are rejected and UnList posts a message. Fields with these data types either cannot contain multi-valued data, or cannot be 'grouped'-- which is necessary for creating the unique list of multi-item values.

3. Define Type of List within the Field:

Choose a button to define the type of list that exists in the selected field. UnList recognizes two types of 'lists' in the multi-valued field: those that are DELIMITED by a one-character delimiter, and those are constructed as FIXED-WIDTH data. Here are examples of each:

Delimited (comma), 3 Items: "red, blue, green"

Delimited (using '\$'), 4 Items (one missing): "xy\$z\$\$12"
Fixed-width (1 character), 10 Items: "YNNYYYYNYN"
Delimited (3 kinds), 6 Items: "2000/08/13 15:34:01" --this is a bad UnList candidate

The last example above has 3 kinds of delimiters separating the Items (slash, space and colon). You will have to use an Update query or other means to convert *all* of the delimiters to a *single type* before using UnList to separate the Items. Consider using UnField to make a table with two copies of the 'list' field, and change the duplicate to have only a single delimiter type. Then use UnList on the field with the single delimiter. Using both the UnField and UnList tables, you now have an Itemized translation back to the original format.

Selecting 'Delimited' as the list type automatically resets the Trim option (see below) to 'Yes'. Selecting 'Fixed-width' as the list type resets the Trim option to 'No'. This is typically how such data are handled.

** Delimited List

Common delimiters (comma, semicolon, space) are available by checking the appropriate option button. A Custom delimiter can be used by entering the delimiter character in the box that is available after checking the 'Other' delimiter option. The 'Other' delimiter can only be 1 character in length. The default 'Other' delimiter can be preset for your regular use on the Preferences tab. If you have multiple delimiters in your input data, use UnField as mentioned above, or create an Update query to convert them all to a single type, 1 character in length.

When a delimiter is encountered consecutively in the input data, it is interpreted to mean that the value between them is missing (Null). For example, if an input string consisted of three commas ",,," and UnList was set to recognize a comma as a delimiter, then it would interpret this as four empty strings delimited by three commas: ",,," = ""","","","". Encountering a delimiter as the first character is interpreted as the first Item (before the delimiter) is empty, and encountering a delimiter as the last character is interpreted as the last Item (after the delimiter) is empty. These empty Items can be eliminated from consideration by using the Drop Null option described below.

** Fixed-Width List

Fixed-width lists are simply parsed, left to right, and broken into Item elements. You enter the width used to identify each Item in the input string and that width defines each and every Item within the list. If the Item-width you select is not evenly divisible into the size of the data, extra characters are simply assumed to be missing from the end of the last Item in the list. There is no truncation of the data. Here is a 9-character input string as an example:

"1a2b3c4d5" with a chosen fixed-width of 2
would produce Items "1a","2b","3c","4d","5"

The default 'Fixed Width' value can be preset for your regular use on the Preferences tab. It must be an integer between 1 and 99.

4. Name the new Table:

The New Table name is suggested as the source field name with a prefix attached, and is assigned automatically when the source field is selected. The default prefix is "tlst" but you can change it to whatever you want on the Preferences tab. Null is allowed for the preferred prefix, creating no prefix at all. You can also ignore the suggested new Table name and enter anything you like on the form.

5. Name the new individual Item field:

The Item field is the one that will contain the individual Values separated from the multi-valued source field. The default name for the Item field is the same as your selected source field with a suffix attached, and is assigned automatically when the source field is selected. The default suffix is "Item" but you can change it to whatever you want on the Preferences tab. You can also ignore the suggested Item field name and enter anything you like on the form.

The Item field is created with the same data type and size settings as the multi-valued source field.

6. Make 'Item Order' Field?

This option controls whether or not UnList creates a field in the output table that numbers the individual Items from the multi-valued field. The Item Order is the left-to-right sequence position of Items in the original multi-valued data string, and the Item Order field is created as an Integer. The default choice is 'Yes'--create the Item Order field.

The default name for the Item Order field is the same as your selected source field with a suffix attached, and is assigned automatically when the source field is selected. The default suffix is "ItemOrder" but you can change it to whatever you want on the Preferences tab ("Position" is also a good choice). You can also ignore the suggested Item Order field name and enter anything you like on the form.

When you select the 'No' option, UnList produces a smaller output table. If each unique Item Value does not need to be associated with a position, a Value only needs to appear once in the output to indicate its presence in the original string. For example, the 3-valued input string "red, blue, red" would result in 3 records in the output table using the 'Yes' option, but only 2 records with the 'No' option:

'Yes'	MultiValue	Order	Item
	"red, blue, red"	1	"red"
	"red, blue, red"	2	"blue"
	"red, blue, red"	3	"red"

'No'	MultiValue	Item
	"red, blue, red"	"blue"
	"red, blue, red"	"red"

If you need to count the occurrence of each unique Item value from each input string, you should let UnList create the Item Order field to preserve the number of Items regardless of any value duplication.

7. Trim separated Items?

This option controls whether or not UnList removes leading and trailing spaces from the individual separated Items (a 'trimming'). Spaces within an Item value are left intact. For example, trimming the string " hi there " would produce "hi there". This option may not be appropriate for Items that consist only of spaces, which would result in an empty Item value (e.g., the string " " would become ""). You can always run your own Update query to trim the Items after using UnList, just to be safe.

When you select the type of list (step 3), the Trim option is automatically toggled to 'No' for Fixed-Width fields, and to 'Yes' for Delimited multi-value fields. These are natural settings in my work. You can override the suggested choice by simply clicking on the other option. Let me know if those 'auto-changes' are bothersome to you.

8. Keep Null and Zero-Length String Items?

This option controls whether or not UnList keeps a Null or zero-length string as an Item in the output table. The default choice is 'Keep', which means Nulls and zero-length strings are identified as valid Values and are written as Items to the output table. Selecting the 'No' option will not pass Nulls or zero-length string Items on to the output table.

This option preserves or drops the output of any Null or zero-length string single Item. Where the entire Value in the multi-valued field is only a Null or zero-length string (not actually a list), the record containing that Value is lost when using the Drop choice. If the Trim option is also selected, Items that consisted solely of spaces will now be empty and also be dropped, not appearing in the output table.

Correct numbering of the original Item Order is not affected by dropping empty Items. The original Item Order is assigned to the retained non-empty Items. If the Item Order field is not needed, using the Drop choice of this option along with a No for producing the Item Order list will result in a compact output file containing only non-empty Items extracted from the source field.

DO IT!

Click the DO IT! button to create the new table to your specifications.

Other than the 'Working' message, there is nothing displayed to tell you that UnList is processing your field. It can take a while to create the output table if either the input table is very large (>10K records) or the input field has a lot of unique values and large lists. So, please be patient.

The multi-valued data strings are summarized before parsing, so that each is broken into Items only once in the output table, no matter how many records in the original field had that Value. Therefore, the output table serves to translate the multi-valued data element. Order of the Items is considered in defining a multi-valued data element as unique, and each differently ordered set of identical Items is handled separately ("red, blue" is considered different from "blue, red").

A report appears when UnList is done creating your table. You are then given an option to view the new table, which is opened in read-only mode. Change the design properties of the table (field names, data types, indices) by going to design mode when the table contents are shown. You must close and reopen the table to edit Values.

Using the Item Order numerals, one can use a Crosstab query on the new table to show the separated data in their native order alongside the original multi-value (set Original as row, Item Order as column, Min(Item) as Value).

Usage Notes:

* If you only need to separate numerals from text, use UnCode instead. It has many options for handling the separation.

* **JOINING THE RESULTS BACK TO THE ORIGINAL TABLE** -- UnList makes a new table that only contains the unique set of original multi-valued data. That is, if the original table had 1,000 records containing the value "red, blue, green" then the output table would have only three records representing that value (one for each of the items, "red", "blue", and "green"). This makes for a much more compact output table than if UnList created three records for *each* of the original 1,000 records holding that value.

You must make a query to see the itemized results along with the original source table that contained the multi-valued data string. In the Query designer, select both the original table and the UnList output table. Select the multi-valued field in the original table, and drag'n'drop it on the same field in the UnList table to create a join. Double-click the join line and set the join to show all records from the original, and the matching records in the UnList table. Now you can use fields from either table for output or criteria selection, or to create a make-table query to pass the key field(s) from the original table to a new one along with all or a selected subset of the itemized values.

* If the field contains large lists, then the output table may be very large. Consider the complexity of the multi-valued data before running UnList on a field. Choosing 'No' for making an Item Order field can make a smaller output table. You should consider using UnClear to evaluate the input field, or UnField to create a numbered list of the unique multi-value data elements or of the Items themselves after they've been separated.

* It can get very messy if you try using UnList several times in a row on UnList output to handle multiple delimiters. Trust me; change them all to a single type, then use UnList only once. Your life will be simpler. And remember that a delimiter can only be 1 character in length for use in UnList, but during conversion of multiple delimiters to a single type, you can use any type or length you want (I often use "\$\$" or "#\$#")--just change it to a single character before using UnList.

* You will want to say 'Yes' to the Item Order field option if you need to create new labels for each position. To make labels directly in the output table, use UnList to create the new table and then change the Item Order field from an Integer data type to Text. Next, use an Update query to convert the ordering numerals (1, 2, 3, etc.) to the labels you desire. Finally, run a Crosstab query on the updated table to create columns for each record based on the labels.

* Another way to replace the Item Order with text labels is to create a new table with two fields, one (an Integer) having the list of Item Order numerals you need for your labels, and the other (a Text field) containing the labels you want associated with each Item Order position. Create a query where you link the UnList output table to your new table on the Item Order numeral fields, then use the Text label field along with the Item values in your query output. Get creative! There is usually more than one way to do something.

* Besides separating poorly constructed data elements, a typical use would be to search for matching original records containing a particular Item value. To accomplish this, join the original table to the UnList itemized table on the original field, then use the Item field to set the criteria for the match. Item Order can be used to be even more specific ("blue" in 2nd position).

UnMatch

UnMatch Help UnMatch compares two different fields from tables or queries, showing matched and unmatched Values and their record counts

Source Information

Select Source: Table Query

1. [Dropdown]

Number of Records:

Select Source Matching Field:

2. [Dropdown]

Data type:

To provide all possible sorting and grouping views of the matched/unmatched values, UnMatch creates and opens a table to explore the results. Immediately after examining the table, you can use the enabled red 'X' button to delete it. To keep the table, Don't click the button.

Match Target Information

Select Target: Table Query

3. [Dropdown]

Number of Records:

Select Target Matching Field:

4. [Dropdown]

Data type:

UnMatch Result Table

5. Merge the Fields? Yes No

Name the New Table (or accept default):

6. [Text: _UnMatch_temp_] [Arrow]

Return to Main Menu Do It!

Exit UnTools Minimize UnTools © 1999-2003 Steve Tessler

Purpose:

UnMatch has two functions. 1) UnMatch is used to compare the Values in two fields and show you which Values are in both fields (matched), or in one field but not the other (unmatched). The counts of the Values in each source field are also provided. 2) UnMatch can create a "master list" of the Values from both fields.

You can test UnMatch using the tables 'SourceTable' and 'TargetTable' in the demo database, UnToolsDemo.mdb.

How To Use UnMatch:

To use UnMatch you select two fields from either tables or queries. The field data types must be the same. An option allows you to choose between two different output formats.

*Preferences: You can set the default output table name used by UnMatch on the Preferences tab.

*UnMatch can lose the distinction between Nulls, zero-length strings, or space-only values in any text fields you select for comparison. Use UnClear to check the fields if you think blending them would be a problem.

Items below relate to numbered positions on the UnMatch form.

1. Select Source:

Pick the option button for either a Table or Query as the type of source you want. This fills the combo box with the appropriate selections. Select your source from the dropdown list.

Tables: Linked tables can be used as the field source, but the resulting new table will reside in the database (mdb) you are working in, rather than in the database where the source table resides. If you need to create the new table in the source, open the source database and do the work there. A message is displayed if the link appears to be broken. Once a Table is selected, the number of records it holds is determined and displayed below the combo box. For very large or linked tables, it may take a moment for this to complete.

Queries: Only Select and Crosstab queries are allowed and displayed for selection. Queries that are also Parameter queries are rejected when a field is selected. Because a Query must be run to know how many records it returns, "Not known" is displayed for the record count with a Query source. If you want to use a Query source, please test your query to make sure it works before using it as a source. Not all queries will work as input to an UnTool (see the general Help on the UnTools tab for more information about query limitations).

2. Select Source Matching Field:

A Field in the selected Source is chosen from the dropdown list. When a field is selected its data type is determined and displayed. The data type is not always able to be determined from some Query fields. For Text fields, the size setting is also displayed in parentheses (e.g., "Text(20)" for a 20 character Text field). You cannot use UnMatch on fields with a Memo or Hyperlink data type; a message will be displayed if you try.

3. Select Target:

This works exactly the same as selecting the Source (step 1). The same table or query may be used as the source of both fields being compared.

4. Select Target Matching Field:

A Field in the selected Target is chosen from the dropdown list. This works exactly the same as selecting the Source matching field (step 2). You are not permitted to compare a field to itself. [Trust me, it matches perfectly.]

If you select Source and Target matching fields having different data type settings, a message will be displayed asking you to choose again. The data types must match, although Text fields of different sizes may be compared (see DO IT! below for more information).

5. Merge the Fields?

This option controls whether or not UnMatch creates separate fields in the output table for the matched/unmatched Values from the Source and Target.

In either case, where the original field Values match, the Source and Target record-counts are in the same record to allow a count comparison between fields for that Value.

Choosing 'Yes' (the default) merges the fields and produces a 3-field table:

```
SourceCount  FieldValue  TargetCount
```

This format allows you to sort or filter on all Values found in either table using a single field. This also serves as a single 'Master List' of the full set of Values from both fields.

Choosing 'No' to merge the original values will produce a 4-field table:

```
SourceCount  'SourceField in Table'  'TargetField in Table'  TargetCount
```

The fields holding the Values from the Source and Target are in separate fields named as a concatenation of the Source and Target table and field names. This format provides more information about the Source and Target field and table names for later use, and lets you separately evaluate either field's contents in a more straight-forward manner.

Note: When choosing 'No' to merge the fields, the Source or Target field+tablename can't be longer than 60 characters. This is because the new field names are created in the output table, and field names are limited to 64 characters total (the ' in ' phrase adds 4). If you selected 'No' for the merge option and either of the two new fields' names will be longer than 64 characters, an error message will tell you and reset the Merge option to Yes.

6. Name the New Table (or accept default):

The New Table name is suggested as the default indicated on the Preferences tab. UnTools default name is “_UnMatch_temp_” but you can change it to whatever you want on the Preferences tab. Null is not allowed for the preferred UnMatch output table name. You can also ignore the suggested New Table name and enter anything you like on the form. If you had entered or cleared the name from the text box you can restore the preferred name by clicking on the arrow button to the right of the box.

DO IT!

Click the DO IT! button to create the new table to your specifications.

* If you try to compare Text fields set for different Sizes, you are asked if you are willing to compare a right-truncated version of the longer field to the intact shorter field. This is accomplished using the Left function in a query on the longer field to gather characters from left-to-right to the same length as the shorter field. Neither field is 'trimmed' for the comparison. If you want to do an accurate field match, change the shorter field's size to the larger one, or vice versa, before you run UnMatch.

Other than the 'Working' message, there is nothing displayed to tell you that UnMatch is processing your fields. It can take a while to create the output table if either input table is very large (>10K records) or the matching fields have a lot of unique values. So, please be patient.

A red 'X' button is activated ****and selected**** each time you create a new UnMatch table. Click it and the named output table is deleted instantly. If you move the focus from the button when it's activated, it immediately deactivates and will no longer function to remove the table. So, if you accidentally keep a table that you wanted to delete, you must do so manually. If your next UnMatch uses the same table name, you can overwrite it (with confirmation).

Caution: If you have Nulls, zero-length strings, and values consisting only of various numbers of spaces in Text fields being compared, UnMatch can lose the distinction between these value types, and they may be blended together into only one or two values. The exact cause is unknown at this time and is being investigated. In the meantime, before using UnMatch use UnClear to evaluate the fields being blended to see if you may be affected (i.e., have those value types in your fields).

Using the Output Table:

* UnMatch creates a table for its output. The table can be manipulated using field sorting or by right-clicking to use "Filter by Selection" and "Filter Excluding Selection" options to narrow your view. You can also create your own, more elaborate, queries against the table. The down-side of using a table for the UnMatch results is that you may need to compact your database more often than you like if you always delete the UnMatch tables. One solution is to link to your tables from a fresh mdb file, and use UnMatch from there to create the UnMatch table; this approach has no affect on your source database.

* The number of original records having a given Value is displayed for both the source and target fields. If both Source and Target have non-Null record counts for a Value, the Value exists in both fields (a match). If only **one** field has records with a particular Value, it will have a non-Null count while the **other** field will have a Null record count (an unmatched Value).

* Nulls in the original fields are identified and counted for each field. The Null count should be the top record of the UnMatch output table as first displayed (before you do any sorting/filtering). Zero-length strings will sort below Nulls, as will strings that appear Null but actually contain one or more spaces.

* You can see Values unique to a specific field (unmatched) by **selecting** records where the count is Null in the **other** field (right-click to use "Filter by Selection").

* You can see Values which exist in both tables (matched) by **excluding** records with Null counts -- first for one field, and then the other (right-click to use "Filter Excluding Selection").

* By using the Merge option you are also creating a single 'Master List' of the unique values found in the two fields.

* COMPACT your database after any process of creating/deleting tables, such as using UnMatch. You will get better performance and a smaller file.

UnCode

The screenshot shows the UnTools application window with the UnCode tool selected. The interface includes a menu bar with options like UnTools, UnClear, UnField, UnPivot, UnList, UnMatch, UnCode, UnDaTe, and Preferences. A help box at the top explains that UnCode separates numbers from other characters in a text field, appending two new fields to the table. The main area is divided into three sections: Source Information, New Field Information, and Options. Source Information includes fields for selecting a source table and field, and a data type dropdown. New Field Information includes fields for naming the new fields for numbers and text. Options include radio buttons for 'Number as Double or Text?', '1st 'Period' is decimal point?', 'Allow Negative Numbers?', and 'Keep Spaces?'. A 'Do It!' button is at the bottom right, and 'Return to Main Menu', 'Exit UnTools', and 'Minimize UnTools' buttons are at the bottom left. A copyright notice '© 1999-2003 Steve Tessler' is at the bottom right.

UnCode separates numbers from other characters in a text field, appending two new fields to the table

Source Information

Select Source Table:

1. [Dropdown]

Number of Records:

Select Field to UnCode:

2. [Dropdown]

Data type:

Options

Number as Double or Text? Dbl Text

1st 'Period' is decimal point? Yes No

Allow Negative Numbers? Yes No

Keep Spaces? All Trim None

Return to [Main Menu](#) Do It!

Exit UnTools Minimize UnTools © 1999-2003 Steve Tessler

Purpose:

UnCode is used to separate numbers from text in mixed alphanumeric data. It appends two new fields to the source table to hold the separated values. UnCode works well with the most common types of flagged-numeric (such as "< 2.5") and--with the right option settings--even some unusual forms. But, UnCode is not intended to provide a complete cleaning solution for poorly constructed data.

You can test UnCode using the table 'TestUnCode' in the demo database, UnToolsDemo.mdb.

How To Use UnCode:

Items below relate to numbered positions on the UnCode form.

1. Select Source Table:

Because UnCode appends new fields to a table, it can only work with tables and not query sources. The selection combo box contains a list of tables in the database. Select your source from the dropdown list.

You cannot use UnCode on a linked table. Open the source database and do the work there, or import the table rather than linking it.

2. Select Field to UnCode:

A Field in the selected Source is chosen from the dropdown list. This field contains the mixed alphanumeric string that is to be separated into number and text parts. Only Text data type fields are accepted as input to UnCode. If you want to UnCode data stored as other data types, convert them or copy them to a Text field before using UnCode.

3. Name the New Field to hold Number Part:

The Number Part field is the one that will contain the numeric portion of the separated Value. There is no default name, so you must enter one. If you enter the name of a field that already exists, you are given the option to overwrite it. Do so with caution, because it is not reversible.

4. Name the New Field to hold Text Part:

The Text Part field is the one that will contain the non-numeric portion of the separated Value. There is no default name, so you must enter one. If you enter the name of a field that already exists, you are given the option to overwrite it. Do so with caution, because it is not reversible.

Options

There are four UnCode options to specify how the mixed value separation operates; in combination you have 24 variants. You can use these options with more than one run of UnCode on the same data to get a very specific outcome.

A brief explanation of each option choice is also quickly available from a separate button on the UnCode form. The default option settings tend to give the most reliable separation of mixed values for common alphanumeric coding. Here are descriptions of the UnCode options:

* Number as Dbl or Text?

Dbl (default): Stores numerals in a double-precision number field.

Text: Use the 'Text' option only if you want to preserve *all* leading or trailing zeros (e.g., postal zip code '08628', value precision '2.000') or to keep multiple periods or dashes with the numeral string.

This option controls the output data type for the field which will store the Number Part of the UnCoded field. The default is to create a new field for the Number part as a Double-precision number, but you can also choose 'Text' as the data type.

The Text option allows you to generate 'number' strings having leading zeros (like some US postal zip codes), and it will also retain *all* of the trailing zeros on a high-precision number (e.g., '2.000'). Saving the Number Part of the UnCoded field as a Double-precision number results in the loss of any leading or trailing zeros.

If there is more than one period or dash in the input string (see below), all of them will stay with the number-part if you choose both the Text data type option *and* the options to keep the period as a decimal point and allow negative numbers.

* 1st 'Period' is decimal point?

Yes (default): Allows first period to be used as decimal point before, between or after numeric digits. Use data type Text option to keep all periods embedded within numeric string, otherwise Double data type can yield unwanted results ('1.2.3' --> '1.2' !!).

No: No periods go with numerals (2.5 --> 25), all periods go with text part (e.g., 'Green St.').

This option controls how periods are handled. The default is to allow decimal numbers by preserving the first period in your string as a decimal point (for example, "<.2" becomes "<" and "0.2"). When you choose this option any additional decimals beyond the first will truncate the number (e.g., '1.2.3' becomes '1.2'), unless you are creating a Text rather than a double-precision field. If you opt to put periods with the text part (e.g., "15 lbs." becomes "15" and "lbs."), all numerals are compressed together ('1.2.3' becomes '123').

* Allow Negative Numbers?

Yes (default): Allows one 'dash' to be used as a negative indicator *before* the first numeric digit. Use data type Text option to keep all dashes embedded within numeric string, otherwise Double data type can yield unwanted results ('3-5' --> '3' !!).

No: No dashes go with numerals (-3-5 --> 35), all dashes go with text part (e.g., 'A-B-C').

This option controls how dashes are handled. The default is to allow negative numbers by preserving the first dash that precedes your numerals (for example, '> -1.2' becomes "-1.2" and ">"). Any additional dashes or those that do not start the numeric string will truncate the number part (for example, '-1-2-3' becomes '-1'), unless you are creating a Text rather than double-precision field. By not allowing negative numbers, all dashes go with the Text part.

* Keep Spaces?

All: No spaces are removed, and all travel with the text part into the new field.

Trim (default): The text part retains all spaces, but leading and trailing spaces are removed before writing the value to the new field.

None: All spaces are discarded when the string is parsed; all numeric and text characters clump together.

This option controls how spaces are handled. In all cases, spaces are considered text characters and will initially go with the Text part of the separation. The default is to create the text string using all of the spaces, then Trim off any leading or trailing spaces before writing to the data record. You can also choose to keep All spaces (all are preserved with the Text part), or to keep None of the spaces (all are removed).

DO IT!

Click the DO IT! button to create the new fields in your table and populate them with the separate numeric and non-numeric portions of your original data strings.

Progress during operation of UnCode can be monitored by watching the status bar, where records being processed are counted off. A report message box appears when it is done, and you have the option to view the results of the separation using a simple browsing form. The browsing form allows you to sort and filter the results to see if the separation worked as you wanted. The form always opens with a default size and equal column widths, but is resizable for the viewing session.

Usage Notes:

* The UnCode options let you create data groups that can then be manipulated in other ways, so be sure to consider your needs or experiment with the options to get the most progress toward your ultimate goal during the initial number/character separation. You may want to run UnCode on the same strings with different options--or on the results of previous runs--to get a more precise conversion. You may also need to precede or follow the use of UnCode with other steps to clean up the data.

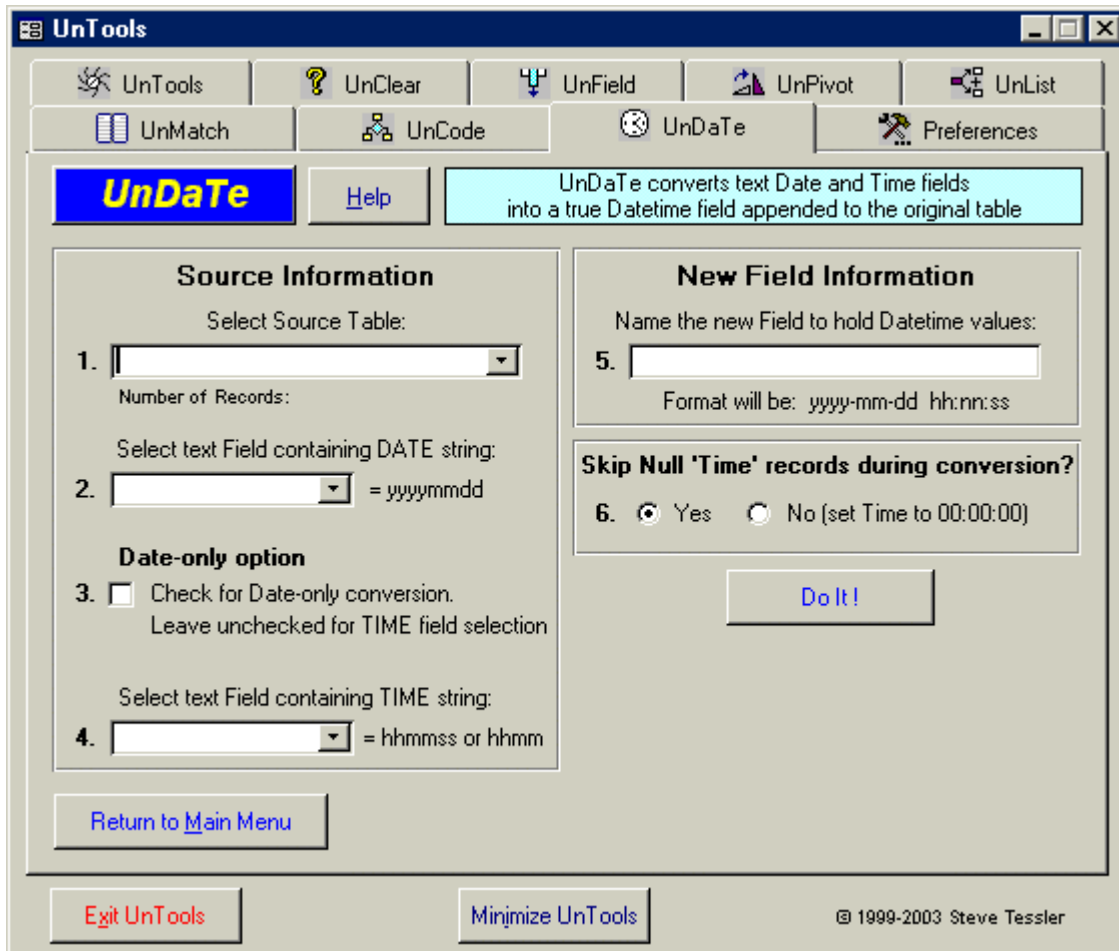
* Compound phrases with numerals in both the number and text parts of the mixed field can be very difficult to separate the way you want, even with all the options available in UnCode. For example, the string "3 m2" which means "three square meters" will become "32" and "m" when UnCoded with the default options. Be aware of this. You should compare the numeral and text strings that result from running UnCode against the original field, and apply whatever Trim or search-and-replace operations suit your data needs. Also consider the following alternate approaches:

* If your data are delimited in some way, you may want to use UnList to separate them first and then use UnCode on the result. For example, the string "<0.3 ft." (translated as "less than 3 tenths of a foot") is best separated using UnList with a 'space' delimiter; this results in two Items: "<0.3" and "ft.". Now you can use UnCode on the output table from UnList to remove the value flag from the numeric and end up with the three Items that truly define the original mixed string: "<", "0.3", "ft.".

* Some adjustments are better to perform *before* UnCoding the field. For example, you can use search-and-replace to change "m2" to "sqm" (for example, in the original string "3 m2"), then run UnCode to separate the numbers from the code. Once that is done, use search-and-replace again to change 'sqm' back to 'm2' in the now-separated Text field.

* COMPACT your database after any lengthy process of modifying tables, such as using UnCode. You may get better performance and a smaller file.

UnDaTe



Purpose:

UnDaTe is used to combine separate numeric strings for date and time stored in Text fields, such as "20010825" and "1430", into a true datetime data type value. UnDaTe creates a new datetime field appended to the source table to hold the correctly converted values. It has options for date-only conversion (a default time is assigned), missing time values, and also handles midnight times stored as "2400" hours.

You can test UnDaTe using the table 'TestUnDaTe' in the demo database, UnToolsDemo.mdb.

Do You Really Need To Use UnDaTe?

Dates and Times shouldn't be stored as text strings, but they often are. This module was created to handle separately stored text Date and Time data and create a new true Datetime field. The format handled is

purely numeric (no usual punctuation associated with date-time displays) and rather specialized; it represents a data format that I encounter on a regular basis and includes an occasional 'midnight' coded as 24:00 (Access insists that midnight is 00:00 at the start of a day, not the end of one). If your Date/Time strings store the correct and usual punctuation, however, using UnDaTe may be more trouble than it's worth for your data. You can always combine values in a query first -- as a test -- to see if you truly need a new converted field to hold the values. Try the following in a query (where CDate is a built-in conversion function, tDate is the name of your text date field, and tTime is the name of your text time field):

```
TrueDateTime: CDate([tDate] & " " & [tTime])
```

If the output in the query is correct for your text Date and Time strings, you don't really need to convert anything and UnDaTe is unnecessary. Just use the concatenation expression above to work with the Datetime version of your data.

On the other hand, if your Date and Time strings are messy and potentially invalid, UnDaTe may be useful. See the information below about rearranging your Date and Time strings to the format needed as input to UnDaTe.

How To Use UnDaTe:

Items below relate to numbered positions on the UnDaTe form.

1. Select Source Table:

Because UnDaTe appends new fields to a table, it can only work with tables and not query sources. The selection combo box contains a list of tables in the database. Select your source from the dropdown list.

You cannot use UnDaTe on a linked table. Open the source database and do the work there, or import the table rather than linking it.

2. Select text Field containing DATE string:

A Field in the selected Source is chosen from the dropdown list. This field contains the DATE text string. Only Text data type fields are accepted as input to UnDaTe. If you want to use UnDaTe on data stored as other data types, convert them or copy them to a Text field before using UnDaTe.

The DATE string must be 8 characters long. This refers to the actual size of the character string to be converted, and not the Size setting for the text field itself. However, attempts to select a DATE text field with a Size setting shorter than the minimum expected will be rejected.

The input format expected for Date strings is 'yyyymmdd'. You will need to remove any punctuation from the Date string ("/" or "-") and add leading zeros where necessary for month and day (for example, to get the 8 June 1999 value "19990608"). It's just not practical to support every possible variant, so only one Date format ('yyyymmdd') is supported.

If your input Date numeric strings are not in UnDaTe format you will have to rearrange your text strings to use the tool. An example of using a query to rearrange strings to the proper UnDaTe format is given at the end of the UnDaTe help.

3. Date-only Option

This option controls whether or not a particular run of UnDaTe is for a Date field only (no Time data used to append to the date). The default is unchecked, meaning a Time field is required.

You can convert Date strings alone by checking the box, in which case a default time of '00:00:00' will be created for each successfully converted Date. The Time field selector and Null Time option portions of the UnDaTe form are disabled if this option is checked (they serve no purpose without Time data to check). Unchecking the option if it was checked re-enables those form components.

4. Select text Field containing TIME string:

* If the Date-only option was selected, the Time field selector is disabled and a default time of 00:00:00 will be assigned to each correctly converted Date string in the output field.

A Field in the selected Source is chosen from the dropdown list. This field contains the TIME text string. Only Text data type fields are accepted as input to UnDaTe. If you want to use UnDaTe on data stored as other data types, convert them or copy them to a Text field before using UnDaTe.

The TIME string must be 4 or 6 characters long. This refers to the actual size of the character strings to be converted, and not the Size setting for the text field itself. However, attempts to select a TIME Text field with a Size setting shorter than the minimum expected will be rejected.

The input format expected for Times are 'hhmm' or 'hhmmss'. Time is only handled in 24-hour format (for example, 0145 is 1:45 AM, 1345 is 1:45 PM). It's just not practical to support every possible variant, so the two most common numeric-only Time formats are supported. You will need to remove any punctuation from the Time strings and add leading zeros where necessary. A.M./P.M. letters in any form in time strings are not allowed, but if these exist in your current format they can assist you in identifying and modifying your 12-hour formats to 24-hour formats.

* Midnight is understood by Access as the start of a day when Time is given in the text string as 000000 or 0000. When midnight is stored as 240000 or 2400 in your files, a conversion is done by UnDaTe to a time of 235959 (one second before midnight) and then 1 second is added to the Datetime created to get us to the 000000 'midnight' time recognized by Access. This means that if your input data stores midnight as 240000, the converted result will have a different *day* than you started with. Although the actual time represented *is* identical, you need to be aware of the meaning of the date change when checking your data conversion results. Access is trusted to create the correct date by adding the 1 second to 23:59:59 of a valid Datetime, e.g., leap year dates.

If your input Time numeric strings are not in UnDaTe format you will have to rearrange your text strings to use the tool. An example of using a query to rearrange strings to the proper UnDaTe format is given at the end of the UnDaTe help.

5. Name the new Field to hold Datetime values:

The new Datetime field is the one that will contain the converted Date and Time strings. There is no default name, so you must enter one. If you enter the name of a field that already exists, you are given the

option to overwrite it. Overwriting is not reversible, so use caution. If you overwrite an existing field, the original one is deleted and the new one is created with a Datetime data type.

The display format for the new Datetime field is set as 'yyyy-mm-dd hh:nn:ss'. You can easily change the format of the new Datetime field in table design mode. This default format has the advantage of sorting in correct chronological order when converted (or exported) to text.

6. Skip Null 'Time' records during conversion?

* If the Date-only option was selected, the Skip Null Time option is disabled and a default time of 00:00:00 will be assigned to each correctly converted Date string in the output field.

This option controls how Null Time data strings are handled with respect to creating a converted Date. It only applies when a valid Date is detected; if the Date string is invalid a Null is immediately written to the output Datetime field without any Time checking.

When the 'Yes' option is selected (the default) and a valid Date string is accompanied by a Null Time value, no conversion of the Date is performed and the resulting value in the new Datetime field is Null.

When the 'No' option is selected and a valid Date string is accompanied by a Null Time value, a default Time of 00:00:00 (start of the day) is appended to the Date for the new Datetime value.

DO IT!

Click the DO IT! button to create and populate the new Datetime field in your table.

Progress during operation of UnDaTe can be monitored by watching the status bar, where records being processed are counted off. A report message box appears when it is done, and you have the option to view the results of the Datetime conversion using a simple browsing form. The browsing form allows you to sort and filter the results. The form always opens with a default size and equal column widths, but is resizable for the viewing session.

Usage Notes:

* If Date/Time conversion failures are primarily due to bad Time values, use the Date-only option at step 3. You can use other methods for cleaning up your bad Time values and then append those times to the converted dates. In the meantime, the converted dates will let you perform date arithmetic that was not possible with the original text date strings.

* You can use UnCode to help with your preprocessing chores--stripping non-numeric characters and spaces, and saving numeric date or time strings as Text.

* If your input Date and Time numeric strings are not in UnDaTe format you will have to rearrange your text strings to use the tool. It's just not practical to support every possible variant, so one Date format ("yyyymmdd") and two Time variants ("hhmm" or "hhmmss") are supported. If you are not sure how to rearrange your strings, look up the Left, Right and Mid functions in Access's help system for assistance and examples. If you have good numeric date strings as a starting point, then this is a fairly simple process. For example, to change the date string "221197" (22 Nov 1997) to UnDaTe format "19971122", we need to

reshuffle the date elements and add the '19' to the year part. Here is the rearrangement formula for that example, and can be implemented using an Update query:

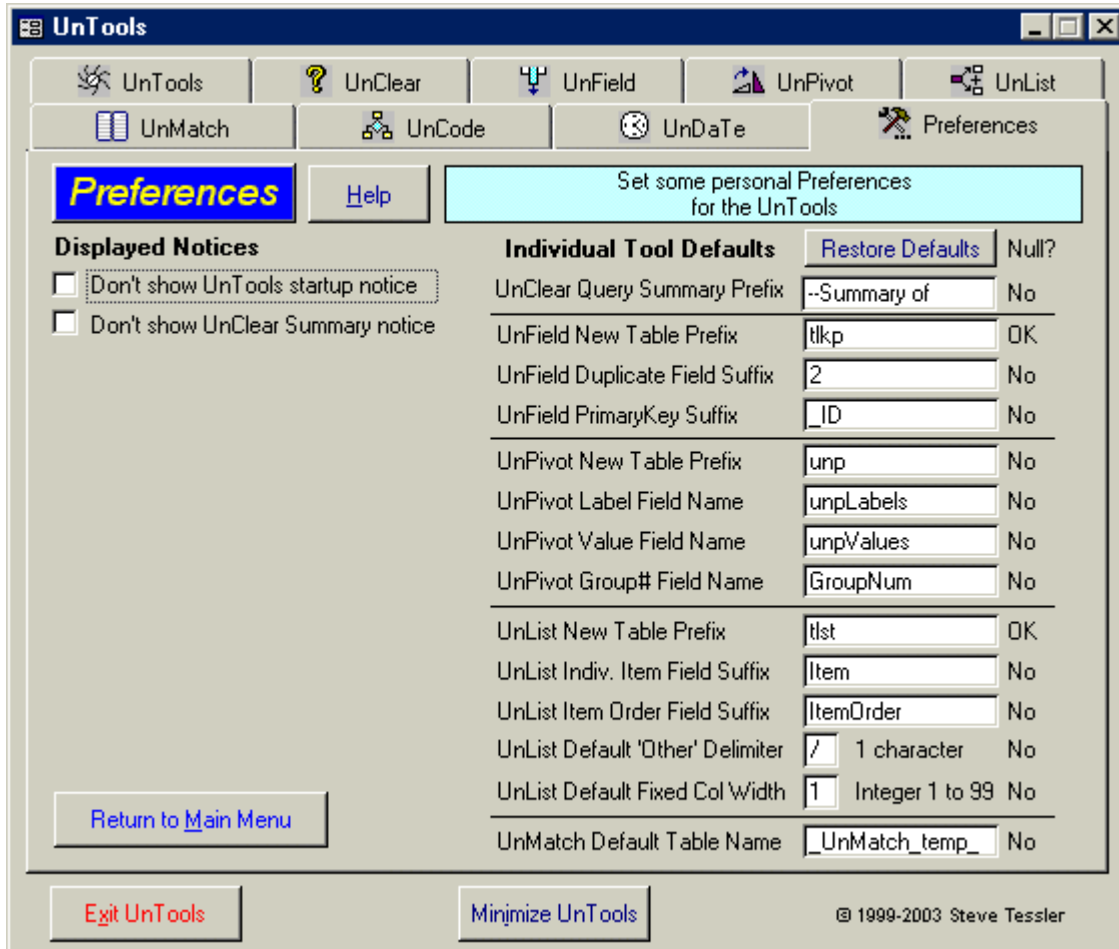
```
[strNewDate] = '19' & Right([strOldDate],2) & Mid([strOldDate],3,2) & Left([strOldDate],2)
```

Rearranging other starting formats simply requires a different conversion expression. Experiment until you find the one you need to get your data in UnDaTe format.

* Some validation is done against your input strings. Dates can't have month digits > 12, or days > 31, and Times can't have hours > 24, or minutes/seconds > 59. The "IsDate" function is run on strings that pass those tests, which should catch invalid dates (e.g., February 30 or June 31). For Date/Time combinations that fail these checks, a Null is inserted into the new field holding converted Datetimes. Those Nulls can then be used to identify bad original date or time inputs. Since valid dates exist in the future as well as the present and past, you would probably want to check the context of a valid date in the year 2345. It is helpful to do some basic quality assurance evaluation before and after using UnDaTe. Check with your local time-travelers to see if they can verify those past/future date values for you. :-)

* COMPACT your database after any lengthy process of modifying tables, such as using UnDaTe.

UnTools Preferences



Displayed Notices

Certain notices are displayed by default when you first run the UnTools. You can turn off the redisplay of the notices on the forms themselves, or through the Preferences. You can also cause the notices to be re-displayed by unchecking the notice on the Preferences form if they had been previously turned off. The notices are associated with:

- the opening splash screen
- the notice about the UnClear Summary Query display (if query view option is selected)

Individual Tool Defaults

- changing preferred values affects defaults used on the forms
- clicking on the Restore Defaults button returns all individual tool preferences to the original default value

UnClear

UnClear constructs a default query name when using the Query option for summarizing a field. The query name consists of a preferred Prefix followed by the source field name. Null is not allowed for the preferred prefix. Although a default name is used to create and open the query, you can Save the query using any name you like once it is opened.

The UnTools default for the UnClear query output name Prefix is "--Summary of ", so that a summary based on a field named YourField would create a query named "--Summary of YourField".

UnField

UnField automatically provides a table prefix and two field suffixes when creating a domain table. The defaults for these values can be set here, but you can also type names you prefer directly on the UnField form for each run. If you change the preferences for UnField naming when you have already selected a source on the form, simply reselect the source field and the new preferences will be applied.

Defaults include:

- Prefix you want for the new table name:
Assigned default = prefix + original source field name
Null [no prefix] is acceptable
UnTools built-in default is "tlkp"

- New Field Name
Automatic default = the original source field name
No preference is associated with this value, which can be changed on the UnField form
The New field name is used as the root-name of the next 2 fields

- Suffix you want for the optional duplicate field:
Assigned default = New field name + suffix
Null is not allowed--to distinguish the duplicate from the main field copy
UnTools built-in default is "2"

- Suffix you want added to the optional Primary Key field
Assigned default = New field name + suffix
Null is not allowed--to distinguish the PK from the main field copy
UnTools built-in default is "_ID"

UnPivot

Note: A checkbox on the UnPivot form labeled "Use Defaults" controls whether the defaults below are applied to your source/field selection. By default on that form, the names are not provided. This allows you to enter a single set of names that are used for multiple appends from different input tables to the same output table. Checking the "Use Defaults" box on the form automatically generates the default names described below. The only exception is the GroupNum field name, which is applied from your default value on the Preferences form. The "Use Defaults" has no affect on GroupNum.

UnPivot automatically provides a table prefix and three field names when creating the output table. The defaults for these values can be set here, but you can also type names you prefer directly on the UnPivot form for each run. If you change the preferences for UnPivot naming when you have already selected a source on the form, simply reselect the source field and the new preferences will be applied. The only naming caveat is that the names for the new table and its fields must be unique. In particular, new fields must not have the same name as any name in the UnPivot header fields or companion fields in the source table or query.

Defaults include:

- Prefix for the new table name - the name of the table containing the UnPivot output
Assigned default = prefix + source table or query name
Null is not allowed--to distinguish the new table from the original source
UnTools built-in default is "unp"
- Field name for the Label field holding the original UnPivoted field names
Null is not allowed
UnTools built-in default is "unpLabels"
- Field name for the Value field holding the original UnPivoted field values
Null is not allowed
UnTools built-in default is "unpValues"
- UnPivot has an option for creating a GroupNumber field in the output table. The name of that field is provided here in the Preferences, and not on the main UnPivot form. The preferred default name entered in the Preferences will be used when that option is selected.

UnTools built-in default is "GroupNum"

UnList

UnList automatically provides a table prefix and two field suffixes when creating the Item output table. The defaults for these values can be set here, but you can also type names you prefer directly on the UnList form for each run. If you change the preferences for UnList naming when you have already selected a source on the form, simply reselect the source field and the new preferences will be applied. The only naming caveat is that the original source, Item, and ItemOrder fields must all have different names. You can also set the default delimiter that will be available from the Other option on the form, and the default fixed width.

Defaults include:

- Original Field Name - the field selected for UnList processing
Automatic default = the original source field name
No change is allowed; the original field is always copied to the output table.
The Original field name is used as the root-name for the table and next 2 fields
- Prefix you want for the new table name
Assigned default = preferred prefix + Original source field name
Null [no prefix] is acceptable
UnTools built-in default prefix is "tlst"
- Suffix you want for the Item field (separate parts of the multi-value data)
Assigned default = Original source field name + preferred suffix

Null is not allowed--to distinguish the Item field from the Original field
UnTools built-in default suffix is "Item"

- Suffix you want added to the optional ItemOrder field (holds Item position number)
Assigned default = Original source field name + preferred suffix
Null is not allowed--to distinguish the ItemOrder field from the Original field
UnTools built-in default suffix is "ItemOrder"

- Default Other delimiter.
Set the default delimiter that you want to appear in the Other box on the main form.
This can only be 1 character in length.
Null is not allowed; at least one character is required.
UnTools built-in default 'Other' delimiter is a slash "/".

- Default fixed width value.
Set the default fixed width that you want to appear in the box on the main form.
This can only be an integer between 1 and 99.
Null is not allowed.
UnTools built-in default fixed width is 1.

UnMatch

UnMatch provides a default table name for its output. The preferred default name entered in the Preferences will appear automatically for you. You can also type whatever name you want for the output table directly on the form, creating new tables with each run.

The UnTools default for the UnMatch result table name is "_UnMatch_temp_".