
MatchUp User's Guide

MELISSA DATA

Your Data Superstore

Copyright

Information in this document is subject to change without notice. Companies, names, and data used in examples herein are fictitious unless otherwise noted. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Melissa Data Corporation. This document and the software it describes are furnished under a license agreement, and may be used or copied only in accordance with the terms of the license agreement.

Copyright © 2017 by Melissa Data Corporation. All rights reserved.

Information in this document is subject to change without notice. Melissa Data Corporation assumes no responsibility or liability for any errors, omissions, or inaccuracies that may appear in this document.

Trademarks

MatchUp is a registered trademark of Melissa Data Corp. Windows is a registered trademark of Microsoft Corp.

The following are registrations and trademarks of the United States Postal Service: CASS, CASS Certified, DMM, DPV, DSF², eLOT, First-Class Mail, LACS^{Link}, NCOA^{Link}, PAVE, Planet Code, Post Office, Postal Service, RDI, Standard Mail, U.S. Postal Service, United States Post Office, United States Postal Service, USPS, ZIP, ZIP Code, and ZIP + 4.

NCOA^{Link}, LACS^{Link} and DSF² are provided by a nonexclusive licensee of the USPS. Melissa Data is a nonexclusive licensee of the USPS for DPV processing and a Limited Service Provider Licensee of the USPS for NCOA^{Link}. The prices for NCOA^{Link} and DPV services are not established, controlled, or approved by the United States Postal Service.

All other brands and products are trademarks of their respective holder(s).

Document number: MU0917UG

Last Update: September 25, 2017

MELISSA DATA CORPORATION

22382 Avenida Empresa

Rancho Santa Margarita, CA 92688

Phone: 1-800-MELISSA (1-800-635-4772)

Fax: 949-589-5211

E-mail: info@MelissaData.com

Internet: www.MelissaData.com

For the latest copy of this reference guide, visit <http://www.MelissaData.com/tech/>

Contents

1	Installation and Requirements.....	5
1.1	Introduction	5
1.2	What's New	6
1.3	System Requirements	9
1.4	Installation	10
1.5	CASS Updates	12
1.6	File Requirements	13
2	Concepts.....	23
2.1	Matchcodes	25
2.2	Matchcode Components	26
2.3	Component Properties	30
2.4	Matchcode Combinations	33
2.5	Blank Field Matching	36
2.6	Matchcode Mapping	40
2.7	Optimizing Matchcodes	44
2.8	Other Uses for Swap Matching	50
2.9	Gathering and Scattering	53
2.10	Dupe Group	57
2.11	Canadian Users	59
2.12	United Kingdom Users	60
2.13	International records	62
3	File Menu.....	63
3.1	New Setup	63
3.2	Open Setup	64
3.3	Save Setup	65
3.4	Save Setup As	66
3.5	Exit	67
4	Setup.....	68

4.1 Merge Setup	68
4.2 Merge/Purge Setup	79
4.3 Purge Setup	118
4.4 File Update	136
4.5 CASS	152
5 Processing.....	165
5.1 Merge	165
5.2 Merge/Purge	168
5.3 Purge	171
5.4 File Update	174
5.5 CASS	177
6 Analyze and Report.....	181
6.1 Edit Results	182
6.2 View Reports	198
7 Tools.....	238
7.1 Browse	238
7.2 Modify Structure	275
7.3 ASCII Conversion	278
7.4 Matchcode Editor	288
7.5 User Settings	310
8 Additional Tools.....	336
8.1 View File	336
8.2 Field List	338
8.3 Expression Builder	340
8.4 Check Mapping	343
8.5 Check Truncations	344
8.6 Global Scatter/Gather	346
9 Reference.....	350
9.1 Command Line	350
9.2 Batch Processing	353

9.3 Errors	355
9.4 dBASE	358
9.5 About Melissa Data Corporation	361
9.6 Copyright & License Agreement	364
9.7 Limited Warranty	367
INDEX.....	369

1 Installation and Requirements

1.1 Introduction

MatchUp introduces a new concept in mail list processing. CASSMATE™ allows you to do CASS certification and merge/purge simultaneously - **in one pass**.

You can process CASS only or merge/purge only, but both processes are improved by combining address standardization and deduplication. Users get more CASS hits, deduplication is more accurate, processing is easier because there's only one setup, and it's faster because the records are only passed once.

MatchUp's new features also include direct reading and writing of Excel, Access, dBASE, FoxPro, ASCII, SQL Server*, Oracle*, and DB/2* tables (* optional); over a dozen additional matching capabilities; and more reporting capabilities, including the ability to output and print in Excel and Word.

1.2 What's New

The following is a short list of new features. You will find many more as you read the manual.

New Setups:

- "CASS" - to CASS-certify as well as Delivery Point Validate (DPV) a list.
- "File Update" - to post fields from one table into another.

Main Window:

- Now has a toolbar for commonly-used commands.
- Customize the look (right-click) with 4 different toolbar settings and 3 background settings.
- You can open a setup file by dragging and dropping it anywhere on this window (or you can still open a setup the traditional way)

Matchcodes:

- Extended 8 simultaneous combinations to 16.
- Extended 3 swap match pairs to 8.
- Added new component types and matching strategies.
- "Start at word:" to start extraction after a specified number of words.
- "Stop at word:" to limit amount of words extracted.
- "Advanced" combinations - combination(s) can be selected to be used only for Suppression comparisons but not Regular comparisons, etc. Restrictions can be ANDed or ORed with specified dBASE expression(s).
- "Optimize" button to alter placement of components for faster processing.
- Right-click menu for specifying combination settings.
- CASS validated Zip code matchcode component for CASS users.

Setup: Input Files:

- Advanced Table Type - Base file table on the contents of a field.
- Rank by a specified Custom Expression.
- Advanced Ranking - User-provided list that ranks the priority of a field based on its contents.

Setup: Output Field Mapping:

- Added new output data types.
- Improved Gather/Scatter options.
- Added Drag and Drop to make field mapping easier.

Setup: Output Files:

- Added "Non-Intersected Record File:".
- Added "Format" to output files to produce alternate output file types such as Access, Excel, SQL, etc.
- New "Multi-Buyer Field" and "Multi-Buyer Format".

- Added the ability to CASS the output table.
- Drag and Drop field mapping.

Setup: Advanced:

- Added "Non-Intersected Record File:"

Processor:

- Direct access to MS SQL Server 7/2000 (optional), MS Access, Excel, Oracle 8i/9i/10g (optional), and DB/2 (optional).
- Multi-processor version (optional).

Analyzer:

- Added "Autosize Columns" and "Reset to default settings" option.
- Show/Hide: Added "Dupe groups having hits from (any/all) selected table(s)."
- Added <Source>, <Dupe Group>, and <Status> as available sortable components.
- Added a right-click popup menu for common tasks.

Reporting:

- Enhanced look of reports.
- Added ability to output reports to Excel and Word formats.

1.3 System Requirements

MatchUp requires:

- Microsoft® Windows XP, Vista, 7, 8, or 10.
- Intel Pentium III® or better.
- 64 MB of RAM minimum. 128 MB of RAM is recommended.
- 10 MB (+500 MB with CASSMATE, +500 MB with DPV) of available hard disk space.

Software Limitations:

- MatchUp will process a total of 20 million records in a single process.

1.4 Installation

Please exit all programs before installing MatchUp.

To Install from a CD:

1. Load the installation CD into your CD-ROM drive.
2. Installation program should start automatically and the install program will appear.
3. Follow the on-screen instructions.

If Installation does not Start Automatically:

1. Click **Start** on the task bar.
2. Click **Run...**
3. Type **d:\setup.exe** (where **d:** is your CD-ROM's drive letter).
4. Follow the on-screen instructions.

To Install from an Activation E-Mail:

To receive MatchUp via an electronic download, Melissa Data Corporation must be provided with a valid e-mail address. Electronic orders are followed by a ground shipment of the CD and manual, which are your proof of ownership.

1. Download and save the MatchUp Demo from the "Downloads" page on our website: www.MelissaData.com.
2. Run the downloaded file to install the demo version.
3. Once it is installed, run MatchUp. On the demo window, click **OK**.
4. Go to **Help | Activate MatchUp**.
5. Enter the activation code that was provided in the e-mail sent to you (use cut and paste to save some tedious typing).

1.5 CASS Updates

CASS Updates are mailed bi-monthly.

To Install from a CD:

1. Load the installation CD into your CD-ROM drive.
2. Installation program should start automatically and the install program will appear.
3. Follow the on-screen instructions.

If Installation does not Start Automatically:

1. Click **Start** on the task bar.
2. Click **Run...**
3. Type **d:\setup.exe** (where **d:** is your CD-ROM's drive letter).
4. Follow the on-screen instructions.

In addition to updated CASS and DPV tables, any program updates will be installed automatically.

1.6 File Requirements

MatchUp processes tables in .dbf format (dBASE III+/IV, FoxPro), Access, Excel, and ASCII fixed field, delimited, and flat formats. Add-on modules are available which allow direct processing of tables from Oracle, SQL Server, and DB2.

If you haven't purchased these add-ons, you can still process these formats (as well as others not listed here) through ODBC. Direct processing is usually recommended for large runs, as it is considerably faster than ODBC.

Visual FoxPro databases can be processed directly by MatchUp. However, do not use MatchUp to modify the structure of a Visual FoxPro database. The additional header information (structure indexes, catalogs, etc.) will be stripped.

MatchUp can process dBASE as FoxPro tables containing memo fields. However, it cannot process memo fields themselves. Additionally, you should never try to modify the structure of a database containing memo fields. Also, don't use MatchUp's utilities to sort, copy, append from or pack records in databases containing memo fields.

Databases and Tables:

Because of the varied DBMS formats that MatchUp supports, we have tried to use a consistent naming convention for what is a 'Database' and what is a 'Table'.

SQL Server, Oracle, DB/2, and Access:

A database is a database and a table is a table. No mysteries.

dBASE and ASCII file types:

The traditional dBASE standard doesn't have the database/table relationship that the 'larger' DBMS's use. We've adopted the ODBC standard: the file's folder is referred to as "the database", and the file name is the table. However, if you enter a full path (folder and file) in an edit where a database or table is needed, MatchUp will split the name appropriately, so you aren't continually forced into our mode of thinking.

Excel:

The spreadsheet (.xls file) is the database and the sheet (the active tab on the bottom left when viewing the sheet in Excel) is the table.

ODBC:

The database is the ODBC connection string (which may contain a database name, user names, passwords, and who-knows-what-else). The table may describe a table or it may describe something else, it depends on what type of source you're connecting to.

Field Types:

Just as we had to develop a naming convention for databases and tables, we also had to develop a naming convention for field types. MatchUp uses the following field types:

Fixed Character	Any text, usually with a specific size
Variable Character	Any text, usually with varying sizes.
Integer	Numeric integer values, -2147483647 to 2147483647.
Float	A floating point number.
Decimal	Numeric values with a fixed decimal place.
Logical	A true or false value.
DateTime	Chronological data consisting of a date (and maybe time).

Some DBMS's don't support some field types. In these situations, one type may be substituted for another. For a chart on how a particular DBMS's data type will be interpreted by MatchUp please see the File Requirements section of MatchUp's Help File.

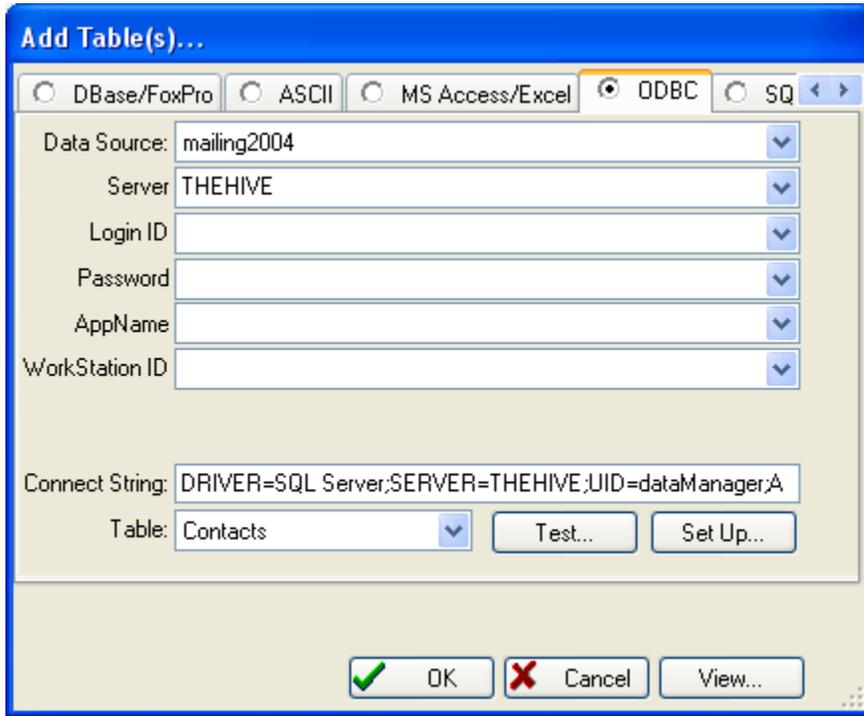
1.6.1 ODBC

Open Database Connectivity (ODBC) is a database-neutral interface. That means an application like MatchUp can access a table through ODBC without any knowledge of the table's method of data storage, location of data, or knowledge of how the data access is being done. For developers, ODBC functions as a 'Rosetta stone' that translates a table's contents into a universal format.

With ODBC, we can process tables that we would otherwise be unable to access. This ability comes at a cost, however. Read operations (building keys) typically take 10% longer to perform. Write operations (updating status fields, and scattering) can take between 20% and 50% longer. With such a decrease in performance, one may wonder why we use ODBC. There are three reasons:

- The alternative, accessing the data directly, is often difficult or impossible, as the internal workings of many databases are proprietary.
- Changes made to a database format by its vendor make maintaining custom database drivers difficult.
- Using a vendor's ODBC driver is a (nearly) guaranteed way of ensuring that a database's integrity is maintained (particularly with multi-user databases).

How to Add or Open an ODBC Table:



At this point, you can select a table in either of three ways:

Method 1:

1. Select the **Data Source** (depending on the data source, additional prompts may appear).
2. Answer any additional prompts appropriately.
3. Select the **Table**. The ODBC driver may ask some additional questions and there may be a small delay while the list of tables is being retrieved.

Method 2:

1. Click **Setup**.
2. Select **File Data Source** or **Machine Data Source**.
3. Select the desired source from the list.
4. Click **OK**. The ODBC driver may ask some additional questions and there may be a small delay while the list of tables is being retrieved.
5. Select the **Table**.

Method 3:

1. Instead of clicking **Add**, simply drag and drop the desired **.dsn** file into the list of sources.
2. The ODBC driver may ask some additional questions and there may be a small delay while the list of tables is being retrieved.
3. Select the **Table**.

Method 2 will often work if Method 1 has difficulties connecting to a data source.

MatchUp's ODBC Requirements:

Not all ODBC drivers are created equal. Some provide more functionality than others. When you try to use an ODBC driver, MatchUp retrieves the driver's capabilities. The following capabilities must exist to read from a table:

- Conformance Level 1 or above
- SQLGetData() support
- SQLFetchScroll() support
- SQLTables() support
- Read access

If the above conditions are not met, MatchUp will display an appropriate message when you try to open a table with such a driver. The following conditions are required if MatchUp is to write to a table:

- SQLSetPos() support
- Absolute Fetching
- Write access

If these last three conditions are not met, MatchUp will still allow you to open a table, but you will not be allowed to use status fields or scatter to the file (basically any option that would involve writing to the table). You can check these capabilities in **Help | Test ODBC Connection**.

Be sure to use the most recent version of a driver (usually available on the vendor's website), as we see occasions when an older version lacked capabilities or was much slower than a newer version. Often the DBMS driver vendor provides a much more complete driver than the stock Microsoft driver that comes with Windows.

Passwords and ODBC:

Accessing ODBC sources that require password access can be a real pain. During several points of a process, MatchUp may need to re-open ODBC data sources. Each time a data source is opened, a password may be required. At the very least, this is bothersome. Even worse, it makes batch (i.e., overnight) operations nearly impossible. There are two solutions:

Solution 1:

If the ODBC driver and database administrator allow it, use a trusted connection. In a nutshell, a trusted connection is a way of associating a Windows user/password login with a database user/password. Only Windows NT, 2000, and XP currently support these connections. Check with your database administrator on the steps to set up a trusted connection.

Solution 2:

You can supply the password to MatchUp in the setup file (.dt):

1. Open the setup file (.dt) in Notepad or some other text editor.
2. Locate the appropriate source file's section start (indicated by [Source#]).
3. You should see a SourceConnection= entry two or three lines below the section start. For example:

```
[Source1]
SourceTable=DEMO
SourceConnection=DRIVER=SQL Server;UID=Marc;PWD=;WSID=MARC
```

4. Note the PWD=; section. Insert the user's password in this area. If the PWD entry is not currently present, add it anywhere in the entry.

```
[Source1]
SourceTable=DEMO
SourceConnection=DRIVER=SQL Server;UID=Marc;PWD=jasmine;WSID=MARC
```

5. Save the file and test it out.

WARNING: If you use this method, you are compromising your password!!! Be sure that this does not violate your company policy!!! By using all of these exclamation points, you can tell we are serious, and will not be held responsible for any negative consequences.

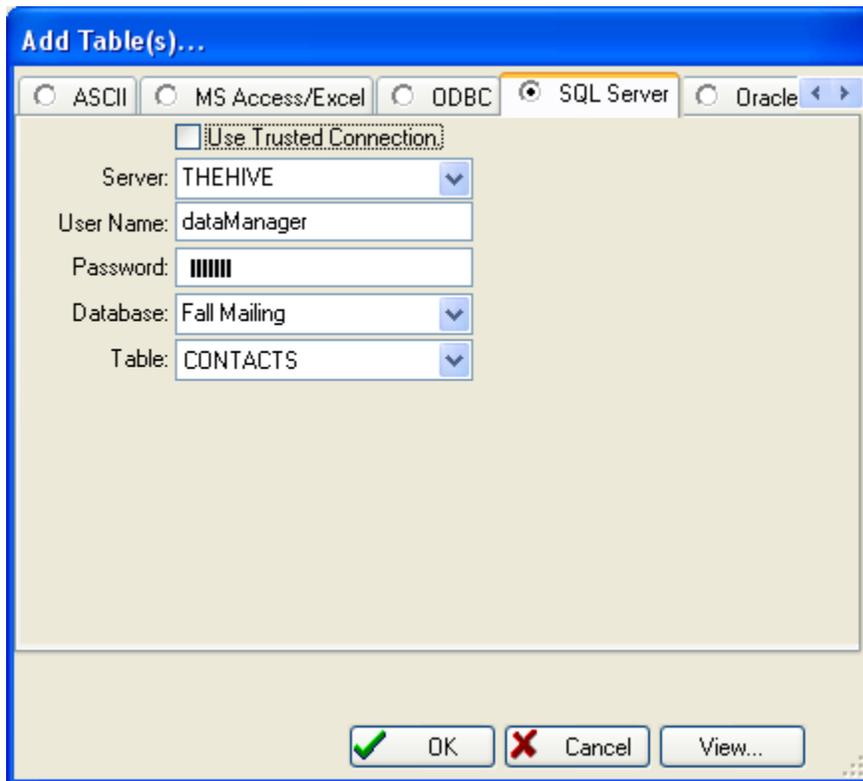
You may be wondering why MatchUp can't remember the password you enter. It is actually the ODBC driver that asks for a password, MatchUp is unaware of this request and is not given the information. The designers of ODBC do this for a good reason: from a security point of view, it is the only way to ensure that an unscrupulous program couldn't use a 'harvested' password for evil instead of good.

1.6.2 SQL Server

SQL Server access is an optional add-on module that must be purchased separately.

How to Add or Open a SQL Server Table:

1. Click the **SQL Server** tab.



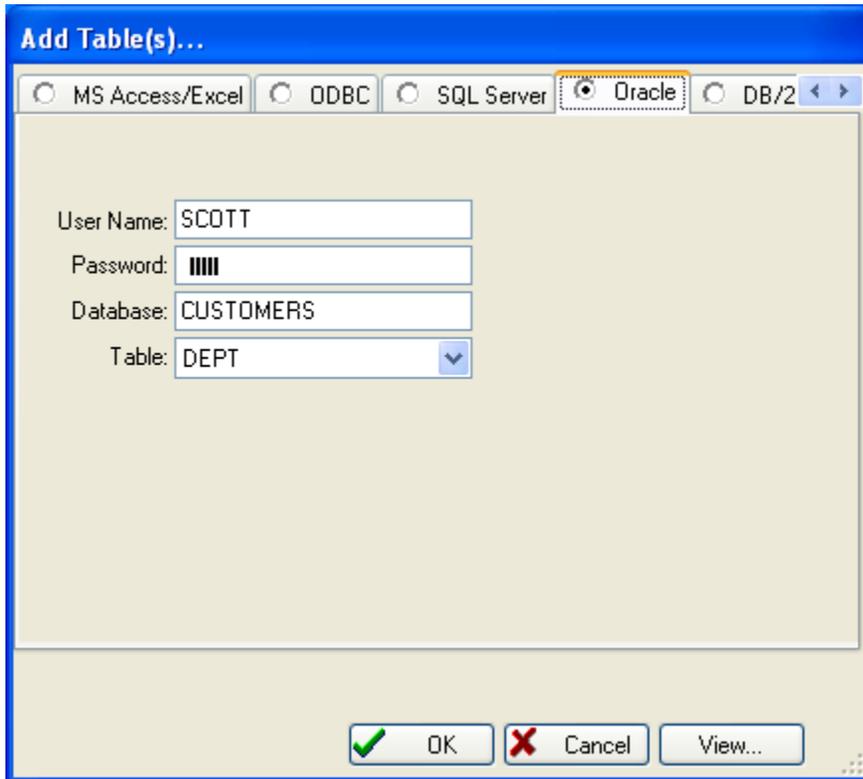
2. Check **Use Trusted Connection** if your SQL Server connection used this technology. If unchecked, the standard login security is used (a **User Name** and **Password** must be entered).
3. Select or type the name of the **Server**.
4. Select or type the name of the **Database**.
5. Select the Table or View that you want to use. There may be a short delay while this list is being retrieved.

1.6.3 Oracle

Oracle access is an optional add-on module that must be purchased separately.

How to Add or Open an Oracle Table:

1. Click the **Oracle** tab.



The screenshot shows a dialog box titled "Add Table(s)..." with a blue header. At the top, there are five radio buttons for different data sources: "MS Access/Excel", "ODBC", "SQL Server", "Oracle", and "DB/2". The "Oracle" radio button is selected and highlighted with a dashed orange border. Below the radio buttons, there are four input fields: "User Name:" with the text "SCOTT", "Password:" with masked characters "||||", "Database:" with the text "CUSTOMERS", and "Table:" with a dropdown menu showing "DEPT". At the bottom of the dialog, there are three buttons: "OK" (with a green checkmark icon), "Cancel" (with a red X icon), and "View...".

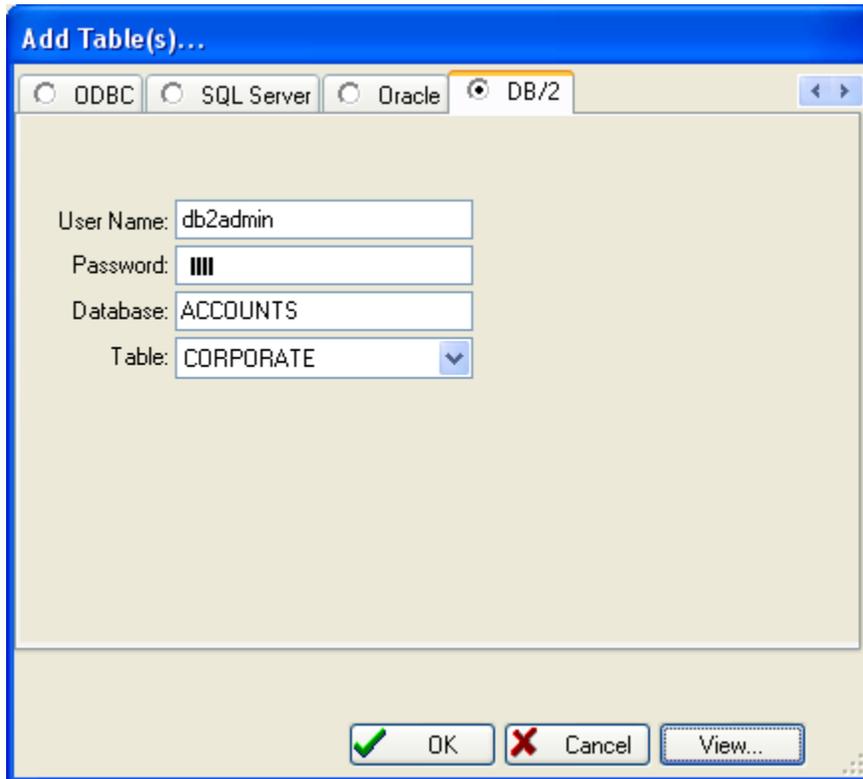
2. Enter the **User Name** and **Password**.
3. Enter the name of the **Database**
4. Select the Table or View that you want to use. There may be a short delay while this list is being retrieved.

1.6.4 DB/2

Oracle access is an optional add-on module that must be purchased separately.

How to Add or Open a DB/2 Table:

1. Click the **DB/2** tab.



The screenshot shows a dialog box titled "Add Table(s)..." with a blue title bar. At the top, there are four radio buttons: "ODBC", "SQL Server", "Oracle", and "DB/2". The "DB/2" radio button is selected. Below the tabs, there are four input fields: "User Name:" containing "db2admin", "Password:" containing "||||", "Database:" containing "ACCOUNTS", and "Table:" containing "CORPORATE" with a dropdown arrow. At the bottom, there are three buttons: "OK" with a green checkmark icon, "Cancel" with a red X icon, and "View..." with a dotted border.

2. Enter the **User Name** and **Password**.
3. Enter the name of the **Database**
4. Select the Table or View that you want to use. There may be a short delay while this list is being retrieved.

2 Concepts

How do you Dedupe?

Nearly every deduplication application uses a *matchcode* key to represent each record. A matchcode key is a string of name and address information that best represents the unique identity of a given record. A typical matchcode key might consist of the Zip/Postal Code, First and/or Last Name, Company Name, and Street Address.

For speed and storage reasons, it is common to use only small segments of these fields. For example, only the first five characters of a last name might be used instead of the entire last name. Pre-processing is often performed on the incoming data to achieve consistency. This processing may be as simple as upper-casing the data or as complex as performing a CASS lookup on an address.

Exactly what goes into the construction of a matchcode key can be found in the Matchcodes section of this manual.

Clustering:

Once a matchcode key is generated for a given record, it can be compared to the keys of other records. Ideally, every record's key would be compared to every other record's key. This, however, is not practical in all but very trivial applications because the number of comparisons grows geometrically with the number of records processed. For example, a record set of 100 records requires 4,950 comparisons (99+98+...). A larger set of 10,000 records requires 49,995,000 comparisons (9,999+9,998+...). Large record sets could take several lifetimes to process - and few customers are willing to wait that long!

So, we developers have made a bold assumption that in order for two matchcode keys to be considered matching, there must be something in the keys that must match exactly. In many cases, this will be all or part of the Zip/Postal Code. So what we do is only compare records that are (in this example) in the same Zip/Postal Code. On the average (in the US using 5-digit Zip Codes), this will cut the average number of comparisons per record by a factor of thousands.

This concept is known as "break grouping", "clustering", "partitioning", or "neighborhood sorting". Chances are good that every deduping program you've ever used has used some type of clustering method.

Here is an example set of matchcode keys using Zip/Postal Code (5 characters), Last Name(4), First Name(2), Street Number(3), Street Name(5):

```
02346BERNMA49 GARD
02346BERNMA49 GARD
02357STARBR18 DAME
02357MILLLI123MAIN
03212STARMA18 DAME
```

When the deduping engine encounters this set of matchcode keys, it compares all the keys in "02346" (2 keys), then "02357" (2 keys), and finally "03212" (1 key). For this small set, 10 comparisons are turned into 2.

2.1 Matchcodes

As discussed in Concepts, a matchcode key is a string of name and address information that best represents the unique identity of a given record using parts of several components (like street number, street name, street suffix, etc).

These components are joined together into a single string. The resultant string is known as a *matchcode key*. To compare two records, one needs to compare the records' keys. If they match, the records are duplicates. Well, it's actually not *that* simple...

Combinations of Components:

Many deduplication applications use the matching technique described above. It's simple, it's fast, and works well most of the time. Some of MatchUp's simpler matchcodes use it themselves. But MatchUp's true power comes from an enhancement to this technique: *concurrent matching*.

MatchUp allows you to specify *combinations* of components (up to sixteen simultaneously). What does this do for you? Assume you were deduping with the following conditions (the number in parenthesis is the size of each component):

Condition #1 Zip/PC (5) + Last Name (5) + Street # (4) + Street Name (4)

Condition #2 Zip/PC (5) + Last Name (5) + PO Box (10)

Here are three example records:

Name:	Joe Smith	Suzi Smith	Billy Bob Smith
Address 1:	326 Main Street	PO Box 11086	PO Box 11086
Address 2:		326 Main St.	
City/State/PC:	Pembroke, MA 02066	Pembroke, MA 02066	Pembroke MA 02066

Here are the matchcode keys that would be derived using the matchcodes listed above:

	Condition #1	Condition #2
Joe's Keys:	02066SMITH326 MAIN	02066SMITH
Suzi's Keys:	02066SMITH326 MAIN	02066SMITH 11086
Billy Bob's Keys:	02066SMITH	02066SMITH 11086

Your garden-variety deduplicator would never match these three records - the matchcodes are close, but not exact. The type of information available from each record is different.

With MatchUp, Joe and Suzi match using condition #1, Suzi and Billy Bob match using condition #2. Billy Bob is considered a match to Joe through *inferred matching* (Billy Bob matches Suzi who matches Joe). Inferred matching is an important part of MatchUp's matching technology.

2.2 Matchcode Components

The following are the matchcode components at your disposal:

Prefix	A name's prefix.
First Name	A name's first name.
Middle Name	A name's middle name.
Last Name	A name's last name.
Suffix	A name's suffix.
Gender	The sex of the name (Male/Female/Neutral).
First/Nickname	A representative nickname for a First Name.
Middle/Nickname	A representative nickname for a Middle Name.
Department/Title	A title and/or department. (Note 1)
Company	A company name. (Note 1)
Company Acronym	A company's acronym. (Note 1, Note 2)
Street Number	An address's street number. (Note 3)
Street Pre-Directional	"South" in "3 South Main St". (Note 3)
Street Name	An address's street name. (Note 3)
Street Suffix	An address's suffix. (Note 3)
Street Post-Directional	"North" in "3 Main St North". (Note 3)
PO Box	PO Boxes also include Farm Routes, Rural Routes, etc. (Note 3)
Street Secondary	Apartments, floors, rooms, etc. (Note 3)
Address	A single unsplit address line. (Note 4)
City	A city. Zip/Postal Code is usually more accurate.
State/Province	A state or province.
Zip9	A full Zip+4 code (9 digits). (Note 5)
Zip5	The Zip code (5 digits). (Note 5)
Zip4	The +4 extension of a Zip+4 code (4 digits). (Note 5)
Validated Zip9	A full Zip+4 code (9 digits) that has been CASS-verified. (Note 9)
Validated Zip5	The CASS-verified Zip code (5 digits). (Note 9)
Validated Zip4	The CASS-verified +4 extension of a Zip+4 code (4 digits). (Note 9)
Postal Code (Canada)	A Canadian Postal Code. (Note 5)
City (UK)	A city in the United Kingdom.
County (UK)	A county in the United Kingdom.
Postcode (UK)	A United Kingdom Postcode.
Country	A country. (Note 6)
Phone/Fax	A phone number. (Note 7)
E-Mail Address	An e-mail address. (Note 8)
Credit Card Number	A credit card number.
General	Any general information. ID, birthday, SSN, etc.
Custom	General information to be processed with a user table. (Note 10)

Notes:

- 1. Company, Company Acronym, Department/Title** Frequently these components don't match exactly because of 'noise words' such as "the", "and", "agency", etc. MatchUp strips these words from these components. The lookup tables that contain these lists are the Company table and the Title/Department table and you can edit them to meet your needs.
- 2. Company Acronym** MatchUp converts any multi-word company name into an acronym (for example, "International Business Machines" is squeezed into "IBM"). Single-word company names are left as they are. This conversion is done after noise words are removed.
- 3. Street Address Components** The seven street address components (Street Number, Street Pre-Directional, Street Name, Street Suffix, Street Post-Directional, PO Box, Street Secondary) are obtained by splitting up to three address lines. Note that PO Box and/or Street Secondary *do not* have to appear on their own line, or in a particular field. MatchUp's proprietary "street smart" splitter does all the work for you.
- 4. Full Address** When using the Full Address component, you are at the mercy of every little deviation in data entry. Because MatchUp's street splitter is so powerful, you will want to use street address components instead of the Full Address in nearly all cases. The only exception may be when processing foreign addresses that don't conform very well to US and Canadian (or optionally UK) addressing formats. This is discussed in more detail in the Advanced Matchcodes section.
- 5. Zip9, Zip5, Zip4, Canadian Postal Code** MatchUp removes dashes and spaces from Zip codes. If you are processing a mix of Canadian Postal Codes and US Zip Codes, use the Zip9 component.
- 6. Country** Countries are verified and corrected with the Country lookup table.
- 7. Phone Number** MatchUp removes non-numeric characters from phone numbers. Leading "1-" and trailing extensions are stripped if present. Numbers lacking an area code are right justified so that the local dialing code and number are aligned with numbers having area codes. If your table often has missing or inaccurate area codes (i.e., after a recent area code split), you will probably want to start at the 4th position of the phone number component. Do not use the rightmost 7 positions, as badly formatted extensions can sometimes cause the phone number to get coded improperly.
- 8. E-Mail Address** MatchUp removes illegal characters from e-mail addresses. Incomplete, changed, and commonly misspelled domain names are corrected using the E-Mail lookup table.
- 9. Validated Zip9, Validated Zip5, Validated Zip4** Whenever the Validated Zip9, Validated Zip5, or Validated Zip4 components are used in a matchcode, MatchUp will use CASS-verified results for Street Number, Street Pre-Directional, Street Name, Street Suffix, Street Post-Directional, PO Box, and Street Secondary. If an address does not CASS verify, MatchUp falls back to using the record's original Zip code, and splits the street components as it would for a non-validated matchcode. Although it is not common, it is possible for a non-validated address to match a validated address; this procedure ensures the best chance of catching such an event.
- 10. Custom** Custom component allows you to use a search and replace table to make substitutions as a matchcode is being built. For example, here in Boston, a local bank has been bought and sold several times, and is currently Bank of America. If you were matching banking executives, "Joe President" at "Fleet Bank" would not match "Joe

President" at "Shawmut Bank", "Bank of Boston", or "BancBoston". A short custom lookup table containing these names (and replacing them with "Bank of America") would take care of this bank's identity crisis. When a custom component is specified in the Matchcode Editor, you must also specify a Custom File, which will be used for this purpose.

2.3 Component Properties

Every component in a matchcode has its own set of properties.

Size:

The number of characters that will be used. Excess characters are trimmed from the right. This trimming is done *after* all other processing. For example, if MatchUp is generating a size 4 Street Name component and encountered "123 Everest Street", it would first split the address to produce a street name of "Everest", and *then* extract the first 4 characters.

Label:

A label that is attached to this component. MatchUp does not use this label, but it can be helpful in remembering what a particular General component was for.

Maximum number of words:

The maximum number of words that will be used. Like the Size property, the removal of excess words is done after all other processing (except Size).

Start:

Where character extraction should take place. Choices are:

- **Left** From the left end of the data (most common choice).
- **Right** From the right end of the data.
- **Position** From a particular character position in the data.
- **Word** From a particular word in the data.

Trim:

Whether leading and/or trailing white space should be stripped from the data. You nearly always want this option turned on for both leading (left) and trailing (right).

Fuzzy:

Fuzzy settings allow for matching of non-exact components. The following settings are available are mutually exclusive (i.e., you can only select one at a time):

- **Phonetex** (pronounced "Fo-NEH-tex") An auditory matching algorithm. It works best in matching words that sound alike but are spelled differently. It is an improvement over the Soundex algorithm described below.
- **Soundex** An auditory matching algorithm originally developed by the Department of Immigration in 1917 and later adopted by the USPS. Although the Phonetex algorithm is vastly superior, the Soundex algorithm is presented for users who need to create a matchcode that emulates one from another application.
- **Containment** Match when one record's component is contained in another record. For example, "no" is contained in "innovation".

- **Frequency** Match the characters in one record's component to the characters in another without any regard to the sequence. For example "abcdef" would match "badcfe".
- **Fast Near** A typographical matching algorithm. It works best in matching words that don't match because of a few typographical errors. Exactly how many errors is specified on a scale from 1 to 4 (1 being the tightest). The Fast Near algorithm is a speedy approximation of the Accurate Near algorithm described below. The tradeoff for speed is accuracy; sometimes Fast Near will find false matches or miss true matches.
- **Accurate Near** A typographical matching algorithm. The Accurate Near algorithm produces better results than the Fast Near algorithm, but at the cost of speed.
- **Frequency Near** Like Frequency matching except that a slider lets you specify how many characters may be different between the components.
- **Vowels Only** Only vowels will be compared. Consonants will be ignored.
- **Consonants Only** Only consonants will be compared. Vowels will be ignored.
- **Alphas Only** Only alphabetic characters will be compared.
- **Numerics Only** Only numeric characters will be compared.

If the mutually exclusive requirement is too restrictive for you to bear, check the Advanced Matchcodes topic for a workaround. See Matching Strategy.

Short/Empty:

There are several options available for comparing incomplete data. These options can be used together (i.e., not mutually exclusive).

- **Initial Only** Will match a full word to an initial (for example, "J" and "John").
- **One Blank Field** Will match a full word to no data (for example, "John" and "").
- **Both Blank Fields** Match this component if both fields are blank. This is a very important concept in creating matchcodes. See Blank Field Matching for more information.

Swap Match:

Swap matching is the ability to compare one component to another component. For example, if you were to swap match a First Name component and a Last Name component, you could match "John Smith" to "Smith John". Swap matching is always defined for a *pair* of components. MatchUp allows you to specify up to 8 swap pairs (named "Pair A" through "Pair H"). It is strongly recommended that both member components of the swap pair have the same properties. See Swap Matching for more information.

Uppercasing:

Components are always converted into uppercase. No exceptions. This upper casing is internal to MatchUp - we don't change the casing of your data.

Foreign Character Translation:

Foreign characters are translated into English equivalents. For example, "Ç" is converted to "C". All translations are based on the assumption that your data was entered with the 1252 (Windows Latin 1) code page. If this is not true, you can specify an alternate code page in Translation Table. Again, this is internal and does not actually change your data.

2.4 Matchcode Combinations

As discussed in Matchcodes, MatchUp allows you to specify up to sixteen combinations of components that define a match. In order for a match to be found, *any one* of the combinations must be found to match. Programmer-type users will think of this as a *boolean OR* condition. Remembering this fact will keep you out of a lot of trouble with matchcodes.

Specifying Matchcode Combinations:

You specify matchcode combinations in a columnar format. For example, say we wanted to create a matchcode that matched on the following conditions (this is the same example as found in the Matchcodes topic):

Condition #1 Zip/PC (5) + Last Name (5) + Street # (4) + Street Name (4)

Condition #2 Zip/PC (5) + Last Name (5) + PO Box (10)

Remember, satisfying condition #1 **or** condition #2 will constitute a match. We would create the following matchcode:

Component	Size	1	2
Zip/PC	5	X	X
Last Name	5	X	X
Street #	4	X	
Street Name	4	X	
PO Box	10		X

For clarity, columns 3 through 16 have been omitted for this example. The trick to understanding this table is to look at the vertical columns of X's. For example, looking at column 1, we see X's in Zip/PC, Last Name, Street #, and Street Name which indicates our desired condition #1 exactly. Looking at column 2, we see X's in Zip/PC, Last Name, and PO Box matching our desired condition #2.

Let's look at a more advanced example:

Component	Size	1	2	3	4
Zip/PC	5	X	X	X	X
Last Name	5	X	X		
Company	10			X	X
Street #	5	X		X	
Street Name	5	X		X	
PO Box	10		X		X

This matchcode defines four conditions:

Condition #1 Zip/PC (5) + Last Name (5) + Street # (5) + Street Name (5)

Condition #2 Zip/PC (5) + Last Name (5) + PO Box (10)

Condition #3 Zip/PC (5) + Company (10) + Street # (5) + Street Name (5)

Condition #4 Zip/PC (5) + Company (10) + PO Box (10)

Combinations #1 and #2 are very much like the previous example. Combinations #3 and #4 allow for a different situation, where a Company Name matches instead of Last Name.

First Component Requirements:

MatchUp places some special restrictions on the first component in a given matchcode. These restrictions are necessary so that MatchUp can properly cluster your record set (see Concepts). The following must be true for a matchcode's first component:

- It must be used in every condition that you are using.
- It cannot use certain types of Fuzzy Matching: Containment, Frequency, Fast Near, Frequency Near or Accurate Near (other types are okay though).
- It cannot use Initial Only matching.
- It cannot use One Blank Field matching.
- It cannot use Swap Matching.

When using the Matchcode Editor to create matchcodes, the editor will ensure that these conditions are met. If you come across a situation that demands that you have no such component, you can use the workarounds discussed in the Advanced Matchcodes section.

2.5 Blank Field Matching

This needs a special discussion, as its importance is often overlooked. As discussed in the Component Properties section if this property is on, then the absence of data in both records would indicate a match. If this property is off, then two records with that missing piece of data, but matching in every other way, will *not* match.

Upon reading this, one may wonder why on earth anyone would not desire this behavior. Actually, you won't for certain situations. Take the following matchcode (keep a close eye on the **Blank** column):

Component	Size	Blank	1	2
Zip/PC	5	Yes	X	X
Last Name	5	Yes	X	X
Street #	4	Yes	X	
Street Name	4	Yes	X	
PO Box	10	Yes		X

Using what we've learned about Matchcode Combinations, we can derive the matchcode combinations:

Condition #1 Zip/PC (5) + Last Name (5) + Street # (4) + Street Name (4)

Condition #2 Zip/PC (5) + Last Name (5) + PO Box (10)

Using the following records as an example:

Name:	Joe Smith	Suzi Smith
Address:	326 Main Street	405 Main St
City/State/PC:	Pembroke, MA 02359	Pembroke, MA 02359

The following matchcode keys would be generated:

	Zip/PC	Last Name	Street #	Street Name	PO Box
1	02359	SMITH	326	MAIN	
2	02359	SMITH	405	MAIN	

Looking at these matchcode keys, it is clear that this pair do not satisfy condition #1. But because blank field matching is selected, they *do* satisfy condition #2. The Zip/PC, Last Name, and PO Box *are exactly the same*. Therefore the two records *do* match.

Obviously, this is wrong. Lets just make one change to the matchcode:

Component	Size	Blank	1	2
Zip/PC	5	Yes	X	X
Last Name	5	Yes	X	X
Street #	4	Yes	X	
Street Name	4	Yes	X	
PO Box	10	No		X

The same comparison is done for combination #2, but the match is disallowed this time because we have indicated that missing (blank) information is not allowed to figure in the matching condition. Looking at another example (using the same matchcode):

Name:	Joe Smith	Suzi Smith
Address:	PO Box 123	PO Box 456
City/State/PC:	Pembroke, MA 02359	Pembroke, MA 02359

We get the following matchcode keys:

	Zip/PC	Last Name	Street #	Street Name	PO Box
1	02359	SMITH			123
2	02359	SMITH			456

We've got the same problem as before, but this time its combination #1 which is the culprit. So an even better matchcode would be:

Component	Size	Blank	1	2
Zip/PC	5	Yes	X	X
Last Name	5	Yes	X	X
Street #	4	No	X	
Street Name	4	No	X	
PO Box	10	No		X

Now what we have here is a matchcode that works pretty well. There's one more tweak that you might want to do, however - turn on Both Blank Fields for Street #. Occasionally, you may come across records such as:

Name:	Joe Notarangelo	Suzi Notarangelo
Address:	Oceanfront Estates	Oceanfront Est.
City/State/PC:	Pembroke, MA 02359	Pembroke, MA 02359

Not to make any wild generalizations, but this seems to be a trend in your up-scale

neighborhoods. Neither street address has a Street # component, though it is very likely these records should match.

So our new improved matchcode will do the trick:

Component	Size	Blank	1	2
Zip/PC	5	Yes	X	X
Last Name	5	Yes	X	X
Street #	4	Yes	X	
Street Name	4	No	X	
PO Box	10	No		X

Because Street Name still has Both Blank Fields off, we are protected from the accidental combination #1 match. The above matchcode (with or without Last Name) is the basis for nearly all of MatchUp's canned matchcodes.

2.6 Matchcode Mapping

The components in a MatchUp matchcode represent specific types of data, and they aren't directly linked to the fields in your databases. Mapping creates the link between your data and the matchcode.

This may seem like an extra step, but in fact, it's a great time save. Because the matchcodes are abstract, they can be used with many different files and field structures. This saves you the time of recreating them for each new job.

Say you used this matchcode (Address, Last Name):

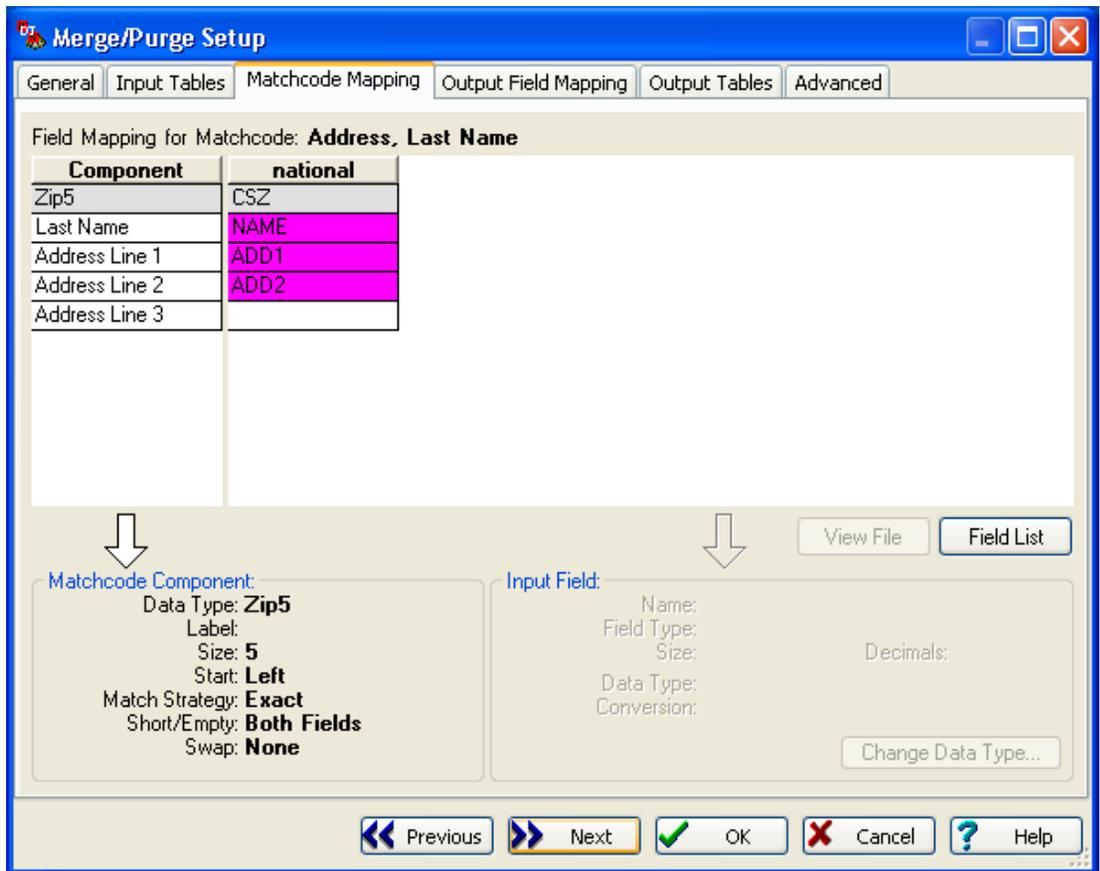
Component	Size	Fuzzy	1	2
Zip5	5	No	X	X
Last Name	5	No	X	X
Street #	5	No	X	
Street Name	10	No	X	
PO Box	10	No		X

And you are working with a table which contains the following fields:

- NAME** Contains full names ("Mr. John Smith").
- ADD1** Contains first (primary) address line ("50 Cole Parkway").
- ADD2** Contains second (secondary) address line ("Suite 34").
- CSZ** Contains City/State/Zip ("Scituate, MA 02066").

You need to create the link between your table's fields (NAME, ADD1, ADD2 and CSZ) and the matchcode components (Zip5, Last Name, Street #, Street Name, PO Box).

On the Matchcode Mapping tab, you would do the following (actually MatchUp will probably default intelligently and do it for you):



This mapping tells MatchUp that the 5-digit zip code information is in a field named CSZ. The Last Name can be found in a field called NAME.

The purple background indicates that some sort of conversion is to take place. If you highlight a mapped field, the "Conversion" text (bottom right) tells you what conversion is needed. In our example, a "CSZ to Zip 5" conversion for the CSZ to Zip 5 mapping, "Full to LN" conversion for NAME to Last Name, and an "Address Split" for STR and STR2.

Matchcode mappings follow three rules:

1. For every Matchcode Component, you should specify a mapping. The only exception is specified in Rule 2.
2. Actual Address components names (such as Street Number, Street Pre-Directional, Street Name, Street Suffix, Street Post-Directional, PO Box, and Street Secondary)

are not listed for mapping purposes. Instead the names Address Line 1, Address Line 2, and Address Line 3 are used. In the example above, we've used three address components in the matchcode (Street #, Street Name, PO Box), however we've only used two (of possible three) address line.

3. If you use address components in your matchcode Address Lines 1-3 will require at least one line to be mapped, but not all. If you only have one address field in your database, you will only need to map that field to Address Line 1.

Note on Rule #1:

Sometimes people have a bit of a problem with Rule #1. Say for example, your database doesn't have a name field. What does one map to the Name Component in that case? You have no business using this matchcode. *Don't use matchcodes that you haven't got the data to support.* Another matchcode should be created or selected that better suits your data. Of course, you don't have to have *exactly* what the matchcode calls for (if MatchUp can convert your data to what it needs, you're okay).

2.7 Optimizing Matchcodes

Some matchcodes process much faster than others in spite of the fact that they detect the same matches. This section will help you to create the most efficient matchcode. 99 times out of 100, clicking the **Optimize** button in the Matchcode Editor will optimize your matchcode sufficiently. But we've included this discussion so that you can better understand why certain things were done while optimizing, as well as what you can do to make the optimizer work better.

Optimizing can make a huge difference in processing speed. We've actually seen 58 hour runs reduced to 4 hours simply by using matchcode optimizations.

It is important that you get your matchcode working the way you want before attempting any of these optimizations. If your matchcode is not functioning properly, these optimizations will not help, and could quite possibly make the situation worse.

Component Sequence:

As discussed in the Matchcode Combinations section, the first component of a matchcode has certain restrictions:

- It must be used in every combination.
- It cannot use certain types of Fuzzy Matching: Containment, Frequency, Fast Near, Frequency Near or Accurate Near (other types are okay, though).
- It cannot use Initial Only matching.
- It cannot use One Blank Field matching.
- It cannot use Swap matching.

If the matchcode's second component also follows these conditions, MatchUp will incorporate it into its clustering scheme (clustering is discussed in Concepts). Furthermore, third, fourth, etc. components will be used if they too satisfy these conditions. Incorporating a component into a cluster greatly reduces the number of comparisons MatchUp has to perform which, in turn, speeds up your processing.

Your existing matchcode(s) may be only a few easy steps away from this optimization. For example:

Component	Size	Fuzzy	Blank	1	2
Zip/PC	5	No	Yes	X	X
Street #	5	No	Yes	X	
Street Name	5	No	No	X	
PO Box	10	No	No		X
Last Name	5	No	Yes	X	X

As shown here, MatchUp will only cluster by Zip/PC. But note that the last component (Last Name) satisfies all the conditions listed earlier.

If you move it to the second position, it too will be used by MatchUp in a cluster:

Component	Size	Fuzzy	Blank	1	2
Zip/PC	5	No	Yes	X	X
Last Name	5	No	Yes	X	X
Street #	5	No	Yes	X	
Street Name	5	No	No	X	
PO Box	7	No	No		X

That's all there is to it; instant speed. The Matchcode Optimizer will always perform this optimization.

Fuzzy Algorithms:

Fuzzy algorithms fall into two categories: *early matching* and *late matching*.

Early matching algorithms are algorithms where a string is transformed into a (usually shorter) representation and comparisons are performed on this result. In MatchUp, these transformations are performed during key generation, which means that early matching algorithms pay a speed penalty once per record: as the record's key is built.

Late matching algorithms are actual comparison algorithms. Usually one string is shifted in one direction or another, and often a matrix of some sort is used to derive a result. In MatchUp, these operations are performed during key comparison. As a result, late matching algorithms pay a speed penalty every time a record is compared to another record. This may happen *several hundred* times per record.

Obviously, late matching is much slower than early matching. If you find that a particular matchcode is very slow, changing to a faster fuzzy matching algorithm may be the way to go. Often, a faster algorithm will give nearly the same results; but, as always, you should test this before processing live data.

The fuzzy algorithms, ranked from slowest to fastest:

Algorithm	Late or Early	Speed (10=fastest)
Accurate Near	Late	1
Fast Near	Late	3
Containment	Late	4
Frequency Near	Late	4
Frequency	Late	6
Phonetex	Early	7
Soundex	Early	8
Vowels Only	Early	9
Numerics Only	Early	9
Consonants Only	Early	9
Alphas Only	Early	9
Exact	N/A	10

The speed values are very rough estimates.

Another benefit of using a faster fuzzy algorithm is that you may be able to exploit the Component Sequence optimization listed earlier. All of the early matching algorithms satisfy the restrictions listed, so they are fair game.

The Matchcode Optimizer will not perform this optimization as it can have a drastic impact on your matching results.

Unnecessary Components:

Components that aren't used in any combinations (i.e., have no X's in columns 1 through 16) are a sign of bad matchcode design. If you find yourself with a matchcode having components like this, you should be asking yourself why. There is one exception, an advanced technique, explained in the Advanced Matchcodes section. Because of this one exception, the Matchcode Optimizer will not automatically remove unnecessary components.

For example, in the matchcode:

Component	Size	Fuzzy	Blank	1	2
Zip/PC	5	No	Yes	X	X
Last Name	5	No	Yes	X	X
First Name	5	No	Yes		
Street #	5	No	Yes	X	
Street Name	5	No	No	X	
PO Box	10	No	No		X

First name is not being used in any combination. Maybe it was used in an earlier iteration of this matchcode, but not any more.

Unnecessary Combinations:

Take a look at the following matchcode:

Component	Size	Fuzzy	Blank	1	2	3	4
Zip/PC	5	No	Yes	X	X	X	X
Last Name	5	No	Yes	X	X	X	X
First Name	5	No	Yes	X	X		
Street #	5	No	Yes	X		X	
Street Name	5	No	No	X		X	
PO Box	10	No	No		X		X

When a user calls our tech support with such a matchcode, we know we're in for a long call. This is usually a warning sign that the user doesn't quite understand the concept of Matchcode Combinations. What's wrong?

Here are the four conditions for matching:

Condition #1 Zip/PC (5) + Last Name (5) + First Name (5) + Street # (5) + Street Name (5)

Condition #2 Zip/PC (5) + Last Name (5) + First Name (5) + PO Box (10)

Condition #3 Zip/PC (5) + Last Name (5) + Street # (5) + Street Name (5)

Condition #4 Zip/PC (5) + Last Name (5) + PO Box (10)

There is absolutely no match that will be detected by condition #1 that won't be detected by condition #3. Similarly, matches found by condition #2 will always be found by condition #4. In other words, condition 3 is a subset of condition 1, and condition 4 is a subset of condition 2. Subsets are generally bad.

So either conditions 1 and 2 aren't needed or conditions 3 and 4 were a mistake. If you do eliminate conditions 1 and 2, you probably will want to remove the First Name component, as it won't be needed (discussed above).

There is one exception to this rule, discussed in the Advanced Matchcodes section. The Matchcode Optimizer will perform this operation (unless Advanced Matchcode Settings are used).

2.8 Other Uses for Swap Matching

Swap matching is used to catch matches when two fields are flipped around. The most common occasion is catching the "John Smith" and "Smith John" records. But there are other uses:

- Comparing household records, where there's two or three first or full names. Although the list provider will tell you differently, you know you can't always rely on the convention used (i.e., they'll say "It's always husband, wife, then children", but the first record will read wife, child, husband):

Data Type	Size	Start	Fuzzy	Short/Em...	Swap	1	2	3	4	5	6	7	8
Zip5	5	Left	Exact	Both Fields	None	X	X	X	X	X	X		
Last Name	5	Left	Exact	None	None	X	X	X	X	X	X		
First Name	5	Left	Exact	None	A/B	X	X						
First Name	5	Left	Exact	None	A/C			X	X				
First Name	5	Left	Exact	None	B/C					X	X		
PO Box	10	Left	Exact	None	None	X		X		X			
Street Number	5	Left	Exact	Both Fields	None		X		X		X		
Street Name	4	Left	Exact	None	None		X		X		X		

In the above example you would want to select **Either component can match** for Swap Pairs A, B, and C. See Swap Match Pairs for more information.

- Comparing up to three address lines. Although the address splitter works well in the US and Canada, some European countries can cause problems. A typical Euro-Matchcode will not use street split components and look at three address lines instead: The swap matching ensures that every address line is compared with every other address line.

Data Type	Size	Start	Fuzzy	Short/Em...	Swap	1	2	3	4
Zip9	10	Left	Exact	Both Fields	None	X	X	X	
Last Name	5	Left	Exact	None	None	X	X	X	
First Name	5	Left	Exact	None	None	X	X	X	
Address	10	Left	Exact	None	A/B	X			
Address	10	Left	Exact	None	A/C		X		
Address	10	Left	Exact	None	B/C			X	

Like the previous example you would want to select **Either component can match** for Swap Pairs A, B, and C. See Swap Match Pairs for more information.

- Don't discard the street split component matchcodes just because you're working on a foreign database, however. Sometimes the street splitter will yield usable results. Often, a combination will do the trick:

Data Type	Size	Start	Fuzzy	Short/Em...	Swap	1	2	3	4	5	6
Zip9	10	Left	Exact	Both Fields	None	X	X	X	X	X	
Last Name	5	Left	Exact	None	None	X	X	X	X	X	
First Name	5	Left	Exact	None	None	X	X	X	X	X	
PO Box	10	Left	Exact	None	None	X					
Street Number	5	Left	Exact	None	None		X				
Street Name	4	Left	Exact	None	None		X				
Address	10	Left	Exact	None	A/B			X			
Address	10	Left	Exact	None	A/C				X		
General	10	Left	Exact	None	B/C					X	

2.9 Gathering and Scattering

2.9.1 Gathering

When duplicates are found, MatchUp's basic merge/purge operation can be described in these steps:

1. Group the block of duplicate records into one "Dupe Group".
2. Using the ranking method you specified on the General tab, determine one of the records to be the "Output Record".
3. Copy the output record to the output table.
4. Copy the remaining records in the dupe group to the dupe table (if a dupe table was specified).

Note step 3:

Only the contents of output record are copied to the output table. Which means that the quality of your output database is completely controlled by the specified ranking method.

In some cases, this is not enough. For example, you may have one database with up-to-date addresses, and a second database with new phone numbers, but bad addresses. Obviously, you will prioritize the first list over the second list. But what do you do about those phone numbers?

This is where gathering comes in. Gathering allows you to consolidate data from records other than the output record. To specify gathering for an output field, you need to do three things:

1. On the Output Field Mapping tab, highlight the output field that will receive the consolidated information. Check the "Gather" box in the "Output Field" box (bottom left). This activates gathering for this particular output field.
2. Select the desired gathering method from the drop down list to the right of the "Gather" box. The details about the various gathering methods are described below.
3. Highlight the *input field* that will contribute to the consolidated information. Check the "Gather" box in the "Input Field" box (bottom right). This activates gathering contribution from this particular input field. If you want several input fields (i.e., from other tables) to also contribute, you must *highlight each one* and check the "Gather" box each time.

2.9.2 Gathering Methods

There are many different ways of consolidating information:

First Data:

MatchUp scans through the Dupe Group and takes the first field it comes across that contains data. This may not necessarily be the output record. If you are using the First Data method on several output fields, the consolidated data will not necessarily come from the same record - *each output field is considered individually*. This is a very common method of consolidation and is available for all output field types (character, date, logical, and numeric).

Join:

MatchUp joins together all of the data contributed into a single string. Each piece of data is not separated by a space; leading and trailing spaces are stripped from the incoming data. Note that the size of the output field should be adjusted to accommodate this joining of data. This method of consolidation is sometimes used when generating a multi-buyer field. Join is only available for character fields.

Join with Spaces:

Same as Join above, but each piece of data is separated by a space.

Longest:

MatchUp scans the group of duplicates and takes the longest data string that it comes across. Leading and trailing spaces are not considered as part of the size. The Longest method is rarely used, but it can be used as a simple "take the field with the most data" selection. Longest is only available for character fields.

Shortest:

The opposite of Longest above.

Maximum Value:

MatchUp scans the group of duplicates and takes the field with the largest value. For transactional databases, this is a good way to determine a maximum purchase value. Maximum is available for character and numeric fields.

Minimum Value:

Same as Maximum, above, but takes the field with the smallest value.

Add Values:

The value in each contributing field is added together to obtain a total result. For transactional databases, this is an excellent way to determine total sale values and the like. Add is only available for numeric fields.

Earliest Date:

The field with the earliest date is selected for output.

Latest Date:

Just like Earliest, above, but takes the field with the latest date. For transactional databases, this is a good way to obtain "last called" or "last updated" information. Latest is only available for date fields.

AND Values:

Logically "ands" the fields together. This is a very rarely used method, only available for logical fields.

OR Values:

Logically "ors" the fields together. This is a very rarely used method, only available for logical fields.

Stack Group (A-H):

There are 8 stack groups (letters A through H). Each letter represents a series of output fields that MatchUp will incrementally fill. For example, you may have output fields FIRST1, FIRST2, FIRST3, and FIRST4 each indicated as Stack Group A, and LAST1, LAST2, LAST3, and LAST4 each indicated as Stack Group B. MatchUp will take the first name information from the output record and deposit it into FIRST1, the last name information into LAST1. The next record's first and last name information will be deposited into FIRST2 and LAST2, the next record into FIRST3 and LAST3, and so on. MatchUp will do this for as many fields in a series as you can specify. When it runs out of fields in a series, the data is discarded. You don't necessarily need to have the same number of output fields in each series (though that is the most common configuration). Be sure to map each source field to each output field (i.e., FIRST to FIRST1, FIRST2, FIRST3 and FIRST4). Also, remember to check "Gather" for each mapped field (input and output).

Fields in a Stack Series don't necessarily have to be adjacent to one another. Stack Fields are populated in the order they appear in "Output Field Mapping", even if they are numerically out of order. That is, if you've specified FIRST3, FIRST2, and FIRST1 in that order, FIRST3 will be populated before FIRST2. You should drag and drop the output fields into a more suitable sequence.

2.9.3 Scattering

Scattering is the process of taking the information in the output record and re-distributing it back into your source file(s). Like gathering, you have complete control over which source files and fields should receive the new information. Gathering and scattering are two independent processes. You don't need to gather to scatter and vice versa, though in most cases, you will find the two used in concert. Selecting an input field for scattering involves one step:

1. Highlight the input field that will receive data from the output record. Check the "Scatter" box in the "Input Field" box (bottom right). This activates scattering redistribution to this particular input field. If you want several input fields (i.e., from other files) to receive data, you must highlight each one and check the "Scatter" box.

In some cases, you may want to gather from one field, but scatter into another. For these cases, you can specify that field in "Alternate Scatter Field". Data will be gathered from the mapped field, but scattered to the specified Alternate Field.

2.10 Dupe Group

Every original record encountered by MatchUp gets a unique number. When a duplicate record is found, it is assigned the same number as the original. This number is known as the *dupe group*.

MatchUp reports this. Its uses include:

- A way for you to connect a 'parent' record with its 'child' records. - also creating a group identifier
- A way to sort your data which will ensure that duplicate records are grouped together.
- An easy way to see if similar records fell into the same group when browsing duplicate tables.

2.11 Canadian Users

MatchUp recognizes Canadian provinces and postal codes. In fact, it will abbreviate province names to their two letter abbreviation automatically. Here's a couple notes for the benefit of our neighbors to the north...

MatchUp does handle the "QC" province abbreviation for Quebec, and "PQ" entries are automatically changed to "QC".

In Canada, "5-20 Main Street" means "20 Main Street, Apt 5", but in the US, it means "5 Main Street, Apt 20". When deduping, MatchUp uses the contents of the zip/postal code as a basis to determine a record's country of origin, and splits this type of address accordingly.

When creating matchcodes for use with Canadian Postal Codes, you should use the **Postal Code** component. However, if your file is a mix of US and Canadian records, use **Zip9** as the component type. Don't worry. **Zip9** *will not* adversely affect processing of Canadian records. The goal here is to prevent the deduper from trying to extract a Zip +4 from a Canadian Postal Code.

2.12 United Kingdom Users

MatchUp can recognize United Kingdom Cities, Counties, and Postcodes. When creating matchcodes for use with United Kingdom addresses, you should use the **Postcode (UK)** component. Depending on your requirements, you also may want to use the **City (UK)** and **County (UK)** components. The Postcode component is structured in the following format: AADDIII, where AA is the Postcode Area (left justified), DD is the Postcode district (right justified), and III is the Inward Code (left justified). Extra spaces and dashes are removed as this structuring is done, so the size of this component is always 8.

Like any other matchcode component, you can compare a portion of the Postcode by reducing it's size and/or starting at a specific position. For example, starting at position 5 for a size of 3 will compare just the Inward code.

MatchUp's street splitter will not split United Kingdom street addresses as well as Canadian and US addresses. Usually, a matchcode containing a mix of split address components and full address components is a good way to get the benefit of the street splitter (which often does perform well), along with a full-address match backup. We have included the **United Kingdom Address** matchcode to be used as a starting point for you to build on.

Our United Kingdom users are welcome to contribute any comments, matchcodes and data that would improve MatchUp's processing. Our exposure to United Kingdom addresses isn't like it is to US and Canadian addresses, so any help from more experienced users is appreciated.

2.13 International records

MatchUp was designed to work with US and Canadian addresses, and it does a pretty good job with addresses from other English speaking countries.

The big problem with international records is with the Street Splitter. Try doing a test run with one of the "canned" matchcodes. If the street splits are not working, use the full address when creating a matchcode instead of using the components (such as street number, street name, etc.).

Often users have had great success when combining the full address and street splitter. For example, here's an international version of one of our supplied matchcodes:

Component	Size	Start	Fuzzy	Short/Empty	1	2	3
General	10	Left	No	Both Empty	X	X	X
Last Name	5	Left	No	Both Empty	X	X	X
First Name	3	Left	No	Both Empty	X	X	X
PO Box	7	Left	No	No		X	
Street #	4	Left	No	Both Empty		X	
Street Name	4	Left	No	No		X	
Full Address	20	Left	No	No			X

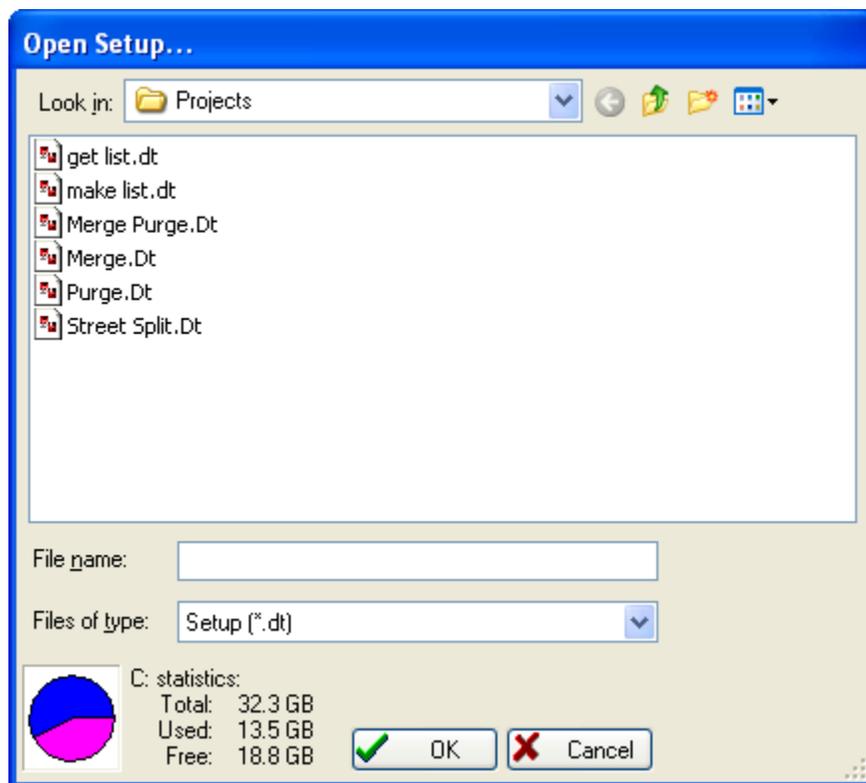
3 File Menu

3.1 New Setup



New Setup lets you select which process you want to use on your tables.

3.2 Open Setup



Once you open a setup, you can edit, process, print, or analyze the results.

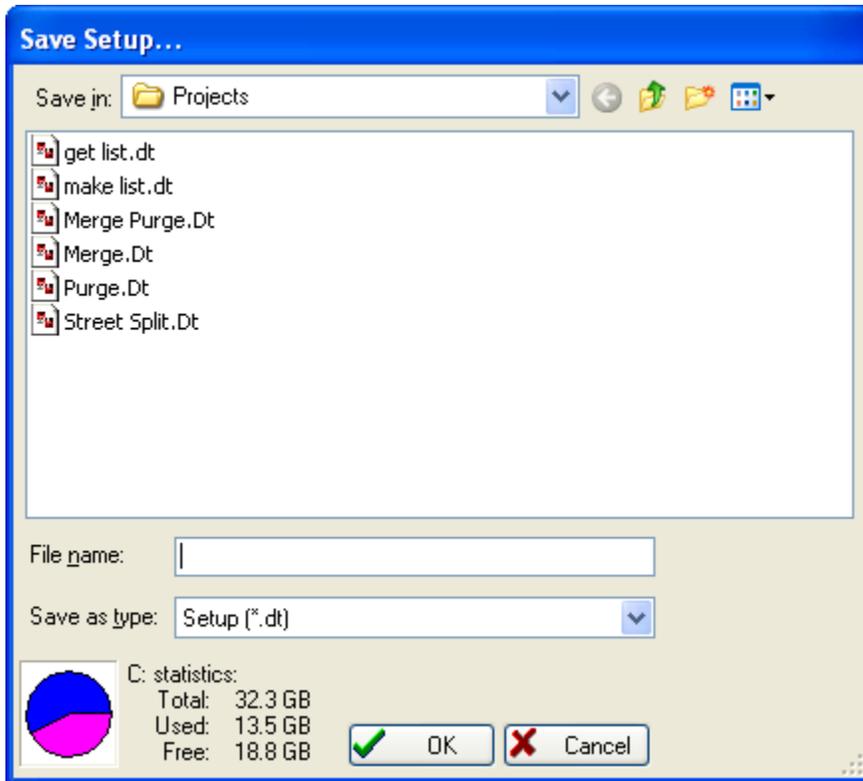
When you create a setup, MatchUp saves the setup details in a file with a **.dt** extension. Table locations, field names, matchcode details, etc. are all stored in the setup file.

3.3 Save Setup

Use **File | Save Setup** to save a setup that you have created or modified. If you have never saved this setup before, you will be prompted for a file name. You can give a setup a new name by using **Save Setup As**. The settings you save are available for future reuse through **Open Setup**.

3.4 Save Setup As

Use **File | Save Setup As** to save a setup with a new file name:



This allows you to slightly modify an existing setup without losing the old setup and without having to completely re-enter a new setup.

3.5 Exit

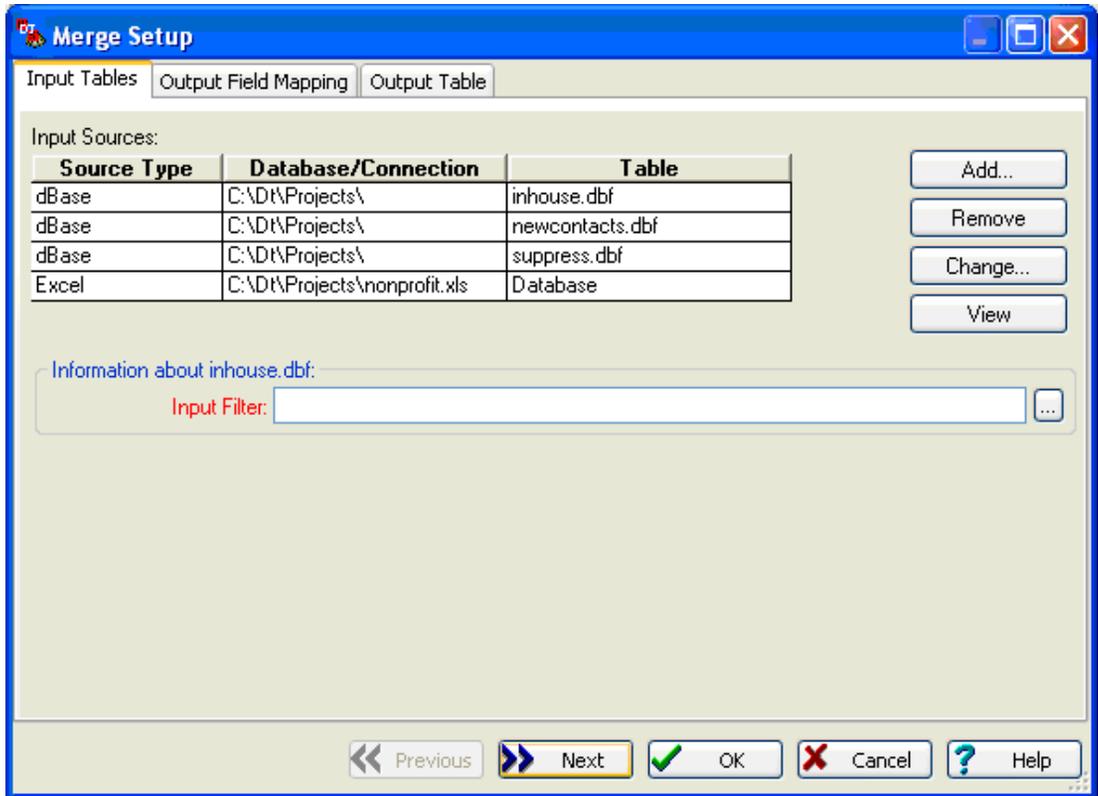
Use **File | Exit** to exit MatchUp. If you've made changes to the setup, you will be prompted to save the changes.

4 Setup

4.1 Merge Setup

Select **Merge** if you want to take several tables and put them into a single table. A merge will not detect nor eliminate duplicate records.

4.1.1 Input Tables



This screen displays the tables you have selected and each table's format type (dBASE, Excel Access, ASCII Delimited, etc.).

You can change the placement of an input table by clicking and dragging it into the desired position. Alternately, you can press CTRL+UP ARROW or CTRL+DOWN ARROW to move the table. Note that the order of input tables has no impact on processing, but is purely for your benefit.

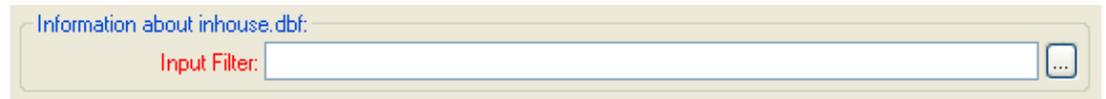
Add... Allows you to add an additional table to the setup.

Remove Removes a table from the merge setup.

Change... Allows you to replace a table in the merge setup with another table.

View Although you cannot edit the data, you can view your table here.

Input Tables: Information about

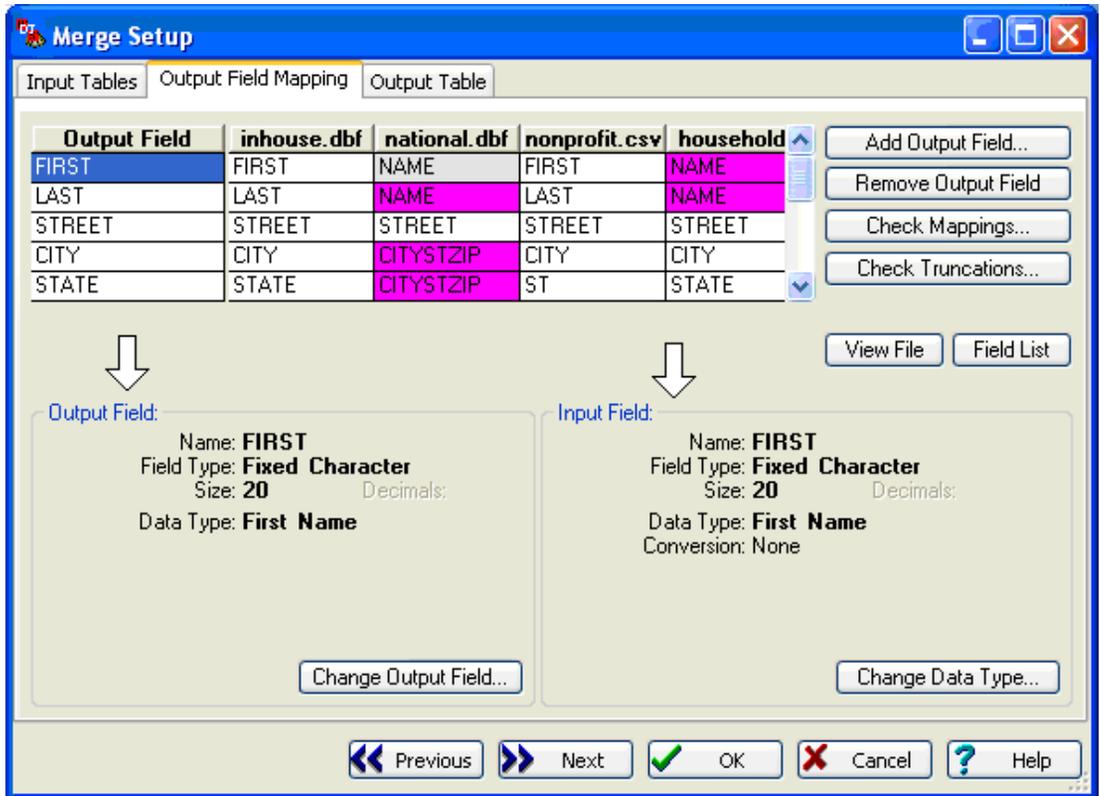


Input Filter (optional) Enter the input filtering criteria for each source table. MatchUp will allow you to use a different filter for each input table. The filter must be entered in dBASE syntax. Instead of typing in a dBASE expression, or if you're unfamiliar with dBASE syntax, you can use the Expression Builder.

Table is Read-Only This warning message will appear if MatchUp will not be able to write any data back to the table.

ASCII File (Cannot be modified) This warning message will appear if the Input Table is ASCII - MatchUp cannot write results back to ASCII tables.

4.1.2 Output Field Mapping



The initial Output table's structure is determined in Default Output Table. You can modify the structure here, but it is easier to start with one that is close to what you want for your output table.

This screen tells MatchUp how it should go about mapping input data into the desired output structure. When you create a setup, MatchUp will try to determine the field mappings for you. Usually it does a pretty good job. If it doesn't, simply double-click on the incorrectly mapped field and select the correct field. Alternately, you can right-click on the field and select the field from a drop-down menu. If your field naming conventions are different from ours, see Field Naming.

In addition to mapping fields, MatchUp determines if a conversion is needed to get the correct data type from the input field to the desired output field. For example, if an input field contains full names ("Mr. John Smith"), but the desired output fields are First Name and Last Name (as two separate fields), MatchUp will perform a name split. On this screen, this action is indicated in the "Conversion" prompt on the bottom right. If the Conversion is incorrect, you can click **Change Data Type** to select a more appropriate data type for the input field. Alternately, you can right-click on the field.

Note that information sometimes has to be extracted from complex fields. For example, under national.dbf above, First and Last will be extracted from Name.

Add Output Field Adds a new field(s) to the output table's structure.

Remove Output Field Selects a field(s) to remove from the output structure.

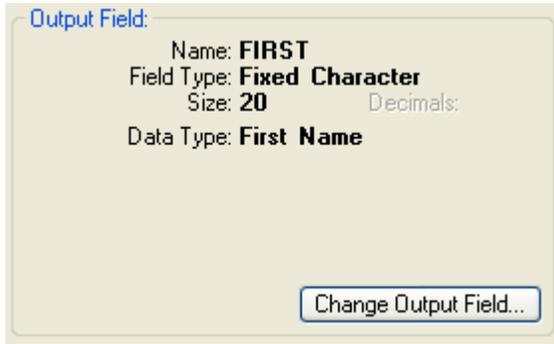
Check Mappings Ensures that you haven't forgotten any source table's fields.

Check Truncations MatchUp warns you about the possibility of truncation here.

View File Although you cannot edit the data, you can view your table here. This is a great way to make sure you have the right table!

Field List Shows a list of each field in each table. You can use your mouse to drag and drop fields on this list into the desired mappings on the Output Field Mapping Screen.

Output Field Mapping: Output Field



Describes the currently highlighted output field:

Name Name of the field.

Field Type Valid types are: Fixed Character, Variable Character, Integer, Float, Decimal, Logical, and DateTime.

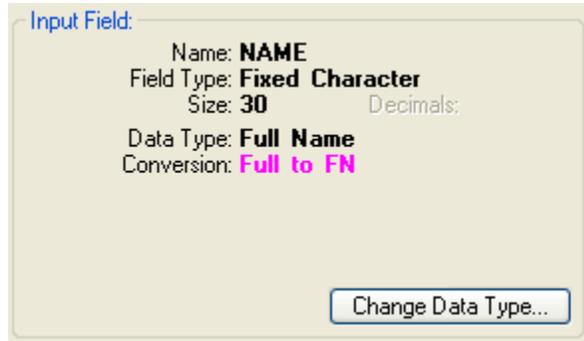
Size Size of the field.

Decimals If the field type is numeric there may be a decimal.

Data Type The type of information contained in this output field. Make sure this is correct or the fields will not merge properly.

Change Output Field Allows you to change all the specifications of a field: Name, Field Type, Size, Decimals, and Data Type.

Output Field Mapping: Input Field



Describes the currently highlighted input field:

Name Name of the field.

Field Type Valid types are: Fixed Character, Variable Character, Integer, Float, Decimal, Logical, and DateTime.

Size Size of the field.

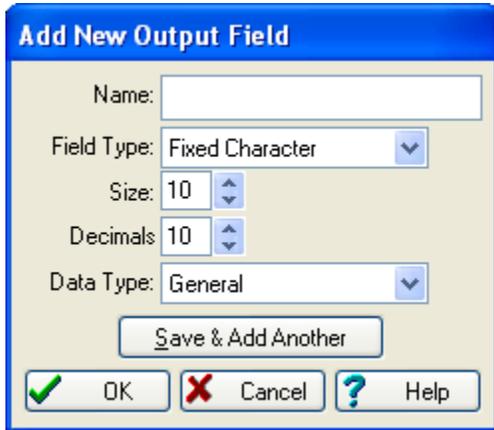
Decimals If the field type is numeric there may be a decimal.

Data Type The type of information contained in this output field. Make sure this is correct or the fields will not merge properly. MatchUp tries to determine a field's type based on the field name. For example, MatchUp will default to "Address" if the field's name starts with "ADD", or "STR". If MatchUp doesn't pick correctly, click **Change Data Type** and select the description which best suits this field. You can customize these defaults in Field Naming.

Conversion Indicates what type of conversion will be done to get from the highlighted field's data type to the output field's data type. For example, an input field containing a full name will have to be split if the output table has first name and last name fields.

Change Data Type Allows you to change the Data Type of an input field. Single right-click for a pop-up menu or double right-click for a dialog box.

Output Field Mapping: Add Output Field



Name Name of the field.

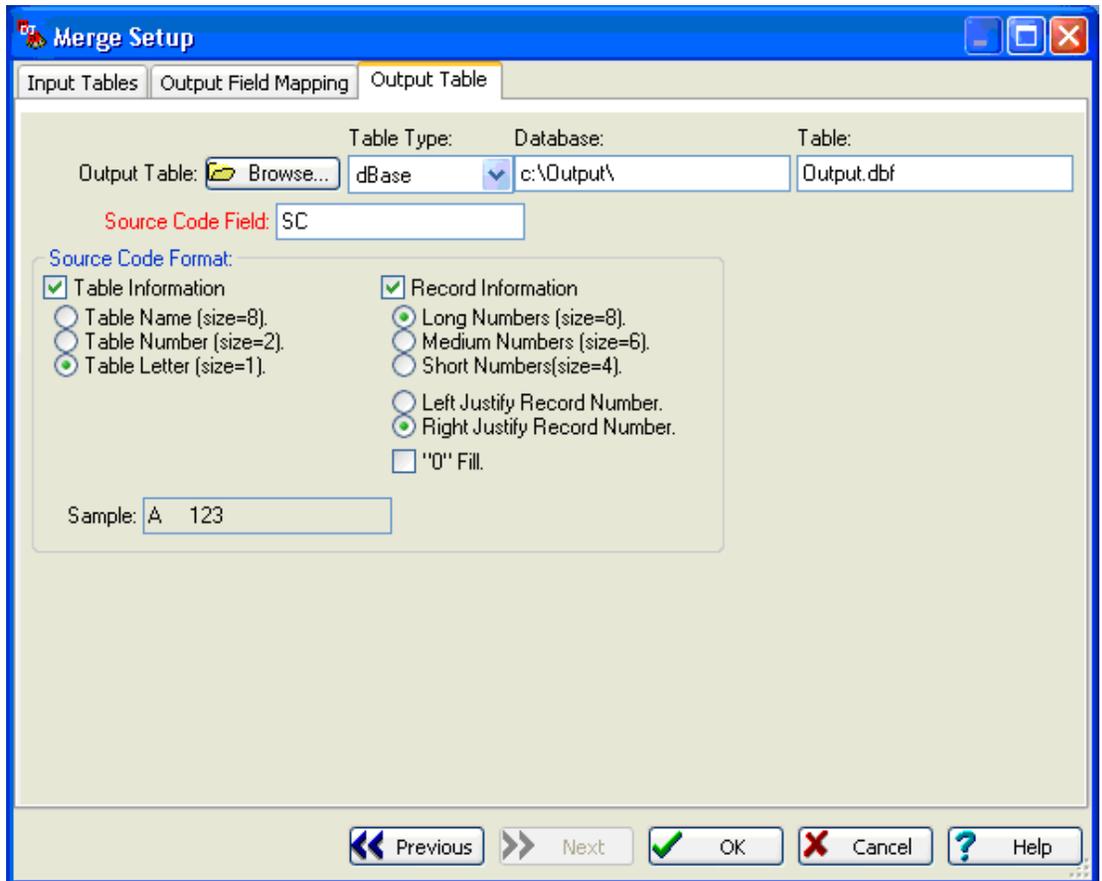
Field Type Valid types are: Fixed Character, Variable Character, Integer, Float, Decimal, Logical, and DateTime.

Size Size of the field. Some Field Types have fixed sizes.

Decimals If the field type is numeric there might be a decimal.

Data Type The type of information contained in this output field. Make sure this is correct or the fields will not merge properly. MatchUp tries to determine a field's type based on the field name. For example, MatchUp will default to "Address" if the field's name starts with "ADD", or "STR".

4.1.3 Output Table



Output Table Specify the table that will receive the merged records. You can use the **Browse** button for the easiest way to select an output table.

Source Code Field: (Advanced) (Optional) Enter the field to be used for source tracking. Source tracking gives you a way to track what table (and optionally record number) the data in the output table came from. If you like, you can name a new field, and it will be added to the table. Table Information and Record Information allow you to format the information in the Source Code field.

Table Information Select how you would like to indicate which table a record came from:

- **Table Name** Uses up to 8 characters of the input table's name.
- **Table Number** Uses up to 2 digits to store the input table number.
- **Table Letter** Uses 1 letter to indicate the input table.

Record Information Select how you would like to indicate which record a record came from:

- **Long numbers** Uses up to 8 digits to store the input record number.
- **Medium numbers** Uses up to 6 digits to store the input record number.
- **Short numbers** Uses up to 4 digits to store the input record number.
- **"0" Fill** Fills the unused record information spaces with "0"s.
- **Left Justify** Left justifies the record number.
- **Right Justify** Right justifies the record number. If you select Right Justify, you also have the option to **"0" Fill**.

Sample Shows you what a typical Source Code Field will look like.

4.2 Merge/Purge Setup

Select **Merge/Purge** if you want to take several tables, remove matching records, and put them into a single table.

4.2.1 General

The screenshot shows the 'Merge/Purge Setup' dialog box with the 'General' tab selected. The 'Matchcode' is set to 'Address, Company OR Last Name'. Below this, a table lists the fields and their properties. The 'Matching Rules' section contains four rules. The 'Ranking' section has four radio button options, with the second one selected. The 'Order' section has two radio button options, with the first one selected. At the bottom, there are navigation buttons: 'Previous', 'Next', 'OK', 'Cancel', and 'Help'.

Matchcode: Address, Company OR Last Name Matchcode Editor...

Address, Company OR Last Name:

Data Type	Size	Start	Fuzzy	Short/Empty	Swap	1	2	3	4	5	6	7	8	9	10	11
Zip5	5	Left	Exact	Both Fields	None	X	X	X	X							
Last Name	10	Left	Exact	Both Fields	None	X	X									
Company	10	Left	Exact	Both Fields	None			X	X							
Street Number	5	Left	Exact	Both Fields	None	X		X								

Matching Rules:

- 1: Z5+LN+S#+SN
- 2: Z5+LN+POB
- 3: Z5+Co+S#+SN
- 4: Z5+Co+POB

Ranking:

- MatchUp will select the best record.
- I will assign priorities or specify a priority field.
- Prioritization will be random for each record.
- A uniform distribution of records will be selected.
- Doesn't matter.

Order:

- Ascending ("A" has a higher priority than "B").
A over B, 01 over 02, 3/10/68 over 4/10/68
- Descending ("A" has a lower priority than "B").
B over A, 02 over 01, 4/10/68 over 3/10/68

When matching records have the same rank...

- the first record processed will be selected.
- the last record processed will be selected.
- a random record will be selected.

Navigation: Previous Next OK Cancel Help

General: Matchcode

Matchcode: Matchcode Editor...

Address, Company OR Last Name:

Data Type	Size	Start	Fuzzy	Short/Em...	S...	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Zip5	5	Left	Exact	Both Fields	None	X	X	X	X												
Last Name	10	Left	Exact	Both Fields	None	X	X														
Company	10	Left	Exact	Both Fields	None			X	X												
Street Number	5	Left	Exact	Both Fields	None	X		X													
Street Name	10	Left	Exact	None	None	X		X													
PO Box	10	Left	Exact	None	None		X		X												

Matching Rules:
 1: Z5+LN+S#+SN
 2: Z5+LN+POB
 3: Z5+Co+S#+SN
 4: Z5+Co+POB

Select the matchcode that best suits your needs. A selection is supplied, and you will probably find a suitable one from this collection. If not, you can click Matchcode Editor to create (or edit) a matchcode.

The table above lists the properties of the currently selected matchcode. It is for viewing purposes only.

Matching Rules: MatchUp uses up to 16 matchcode "combinations" which are evaluated simultaneously. In the table, the X's in the columns marked 1 through 16 define component combinations that should be considered a match. The Matching Rules section simply displays them in an alternate form.

General: Ranking

<p>Ranking:</p> <ul style="list-style-type: none"> <input type="radio"/> MatchUp will select the best record. <input checked="" type="radio"/> I will assign priorities or specify a priority field. <input type="radio"/> Prioritization will be random for each record. <input type="radio"/> A uniform distribution of records will be selected. <input type="radio"/> Doesn't matter. 	<p>Order:</p> <ul style="list-style-type: none"> <input checked="" type="radio"/> Ascending ("A" has a higher priority than "B"). A over B, 01 over 02, 3/10/68 over 4/10/68 <input type="radio"/> Descending ("A" has a lower priority than "B"). B over A, 02 over 01, 4/10/68 over 3/10/68 <p>When matching records have the same rank...</p> <ul style="list-style-type: none"> <input checked="" type="radio"/> the first record processed will be selected. <input type="radio"/> the last record processed will be selected. <input type="radio"/> a random record will be selected.
---	---

Ranking Allows you to choose which records to favor when dupes are found:

- **MatchUp will select the best record** Take the record with the most complete information.
- **I will assign priorities or specify a priority field** Allows you to choose which table or records to favor when duplicate records are found. If you select this option, you will be able to assign a rank or rank field for each input table on the Input Tables tab.
- **Prioritization will be random for each record** MatchUp will randomly select a record to use as the best record when duplicate records are found.
- **A uniform distribution of records will be selected** Instructs MatchUp to try (as best it can) to assign output records uniformly from all input files.
- **Doesn't Matter** Take the first record you come across.

What's the difference between Random and Uniform?

Random ranking tends to take an appropriate proportion of output records from each source. Uniform Distribution tries to take an equal quantity of output records from each source. For example, if there were 100 output records:

	Total Records	Random	Uniform
File1	9,000	90	50
File2	1,000	10	50

Ascending MatchUp will select records with a lower rank over records with a higher rank (i.e., "A" over "B", "1" over "2", and "03/10/68" over "04/10/68").

Descending MatchUp will select records with a higher rank over records with a lower rank (i.e., "C" over "B", "3" over "2", and "05/10/68" over "04/10/68").

When matching records have the same rank If MatchUp can't decide which record to use because they both have the same rank (ie, a tie), a backup plan can be selected:

- **The first record processed will be selected**
- **The last record processed will be selected**
- **A random record will be selected**

4.2.2 Input Tables

Merge/Purge Setup

General | **Input Tables** | Matchcode Mapping | Output Field Mapping | Output Tables | Advanced

Input Sources:

Table Type	Source Type	Database/Connection	Table	Alias
Regular	dBase	C:\DT\Projects\	inhouse.dbf	inhouse
Regular	dBase	C:\DT\Projects\	national.dbf	national
Suppression	dBase	C:\DT\Projects\	suppress.dbf	suppress
Regular	Excel	C:\DT\Projects\household.xls	household\$	household\$
Regular	ASCII Delimited	C:\DT\Projects\	nonprofit.csv	nonprofit

Buttons: Add... Remove Change... View

Information about inhouse:

Table Type:

- Regular.
- Suppression.
- Intersection.
- Self Purge.
- No Purge.

Advanced...

Alias: inhouse

Priority:

- Assign Priority:
- Contents of Field: AGE
- Custom Expression: AGE

Status Marking:

- Mark rejected records for deletion.
- Status Field: <None>
- Count Field: <None>

Navigation: Previous Next OK Cancel Help

Input Tables: Input Sources

Input Sources:

File Type	Source Type	Database/Connection	Table	Alias
Regular	dBase	C:\dt3\Projects\	inhouse.dbf	inhouse
Regular	dBase	C:\dt3\Projects\	national.dbf	national
Suppression	dBase	C:\dt3\Projects\	suppress.dbf	suppress
Regular	Excel	C:\dt3\Projects\household.xls	household\$	household
Regular	ASCII Delimited	C:\dt3\Projects\	nonprofit.csv	nonprofit

Buttons: Add... Remove Change... View

This screen displays the tables you have added.

You can change the placement of an input table by clicking and dragging it into the desired position. Alternately, you can press CTRL+UP ARROW or CTRL+DOWN ARROW to change

the table's position. Note that the order of input tables has no impact on processing, but is purely for your benefit.

Add... Allows you to add an additional table to the setup.

Remove Removes a table from the merge/purge setup.

Change... Allows you to replace a table in the merge/purge setup with another table.

View Although you cannot edit the data, you can view your table here.

Input Tables: Information about

Information about Suppress:

Table Type:

- Regular.
- Suppression.
- Intersection.
- Self Purge.
- No Purge.

Advanced...

Alias: Suppress

Priority:

- Assign Priority:
- Contents of Field: LASTCALL Advanced...
- Custom Expression:

Status Marking:

Mark rejected records for deletion.

Status Field: <None>

Count Field: <None>

Table Type Select how the highlighted table will be handled:

- **Regular** This table will be deduped against itself and any other Regular tables.
- **Suppression** Records in Regular, No Purge, and Self Purge tables which also appear in a Suppression list will be rejected. Ideal for "do not mail" lists.
- **Intersection** Records in Regular, No Purge, and Self Purge tables which do not appear on an Intersect list will be rejected. (opposite of Suppression).
- **Self Purge** This table will be matched against itself, Suppression, and Intersection tables, but not against Regular, No Purge, or other Self Purge tables. Self Purge tables are used when you would like to match several tables against a Suppression list, and purge internal dupes, but not match the tables against each other.
- **No Purge** This table will not be deduped against any other table. It will, however, be matched against Suppression and Intersection tables. No Purge tables are used when you would like to match several tables against a Suppression list, but not dedupe the tables internally or against each other.
- **Advanced** Click if the contents of a field dictate the Table Type.

Alias Enter a name you would like to refer to this table. This name will appear on all reports.

Priority Select a ranking method for the highlighted table:

- **Assign Priority** Enter a rank for this table. This option only appears if you selected I will assign priorities or specify a priority field under the General tab.
- **Contents of Field** Enter the name of the rank field. This option only appears if you selected I will assign priorities or specify a priority field under the General tab. This field must already exist in your table.
- **Advanced Rank** Check this option when the values in the specified Field are not easily compared.
- **Custom Expression** Enter a valid dBASE expression for evaluating rank. Use it when you need a tighter criteria than a single field or table will allow. For example, if you want to mail to the oldest female in a household, enter the expression: SEX + DtoS(BIRTHDATE). Only appears if you selected I will assign priorities or specify a priority field under the General tab.

Status Marking You can select methods of marking a record's disposition with these options:

- **Mark rejected records for deletion** If checked, rejected records will be marked for deletion. This can only be done with dBASE tables.
- **Status Field** (optional) Select a field name for capturing a record's disposition. During processing, this field will be populated with a code indicating whether this record is a dupe, has dupes, was suppressed, etc. See MatchUp Status Codes.
- **Count Field** (optional) Select a field name to receive a record's dupe count. During processing, this count field will be populated with a number indicating how many dupes a record has. See Counting Method.

Information about: Advanced Table Type

Sometimes tables are created by a Merge process and contain a code field specifying their original source. Before MatchUp, to process different source codes as different table types, users would have to 'undo' the merged file into separate files, then Merge/Purge them. Now you can specify the table type as Advanced and specify the field containing the code indicating how the record should be processed.

Advanced Table Type Settings for inHouse.dbf:

Base Table Type selection on contents of field: STATUS

Number of characters to use from field: 3 Ignore case
(when specifying multiple codes, separate each code with a comma)

Table Type 'Regular' when code is: SUB

Table Type 'Suppression' when code is: DMA

Table Type 'Intersection' when code is:

Table Type 'Self Purge' when code is:

Table Type 'No Purge' when code is:

Any codes not in any of the above lists should be processed:

Not Processed (filtered).

Table Type 'Regular'.

Table Type 'Suppression'.

Table Type 'Intersection'.

Table Type 'Self Purge'.

Table Type 'No Purge'.

Previous Next OK Cancel Help

Base Table Type selection on contents of field The field containing a code indicating how a record should be processed.

Number of characters to use from field How many characters in the above field should be used in considering how a record should be processed.

Ignore case Whether or not case should be considered when looking up a code in the lists below.

Table Type 'Regular' when code is A list of code(s) that indicate that a record should be processed as a regular record.

Table Type 'Suppression' when code is A list of code(s) that indicate that a record should be processed as a suppression record.

Table Type 'Intersection' when code is A list of code(s) that indicate that a record should be processed as an intersection record.

Table Type 'Self Purge' when code is A list of code(s) that indicate that a record should be processed as a self purge record.

Table Type 'No Purge' when code is A list of code(s) that indicate that a record should be processed as a no purge record.

Any codes not in any of the above lists should be processed Other codes will be processed in the following manner:

- **Not Processed (filtered)**
- **Table Type 'Regular'**
- **Table Type 'Suppression'**
- **Table Type 'Intersection'**
- **Table Type 'Self Purge'**
- **Table Type 'No Purge'**

Information about: Advanced Rank

When the field chosen to provide the ranking has values which can not easily be compared, Advanced Ranking can provide an order to those values.

For example, a rank field containing Months would wind up with unusual prioritization if the values were compared by character strings: Apr, Dec, Feb, Jan, Jul, Jun, etc.

Advanced Rank Settings for national.dbf:

Map the value in the Rank Field to an alternate set of priorities.

Number of characters to use from field: Ignore case

Use an external file listing ranks and priorities:

This list should be a two-field dBase file. The first field will be used as the Lookup Rank, the second field, the Desired Priority.

Prioritize ranks in this order:

The above list should list rank codes, from lowest to highest priority. Separate each code with a comma.

Codes that don't appear in the above list should be:

Not Processed (filtered).

Assigned the lowest priority.

Assigned the highest priority.

When the list of prioritized ranks is quite long, it can be get tedious to manually enter a list of ranks in the manner specified above. A more efficient method is to create a two-field lookup table (dBASE III format) and specify it using the first option, **Use an external file listing ranks and priorities**. In our example, the lookup table would look like the following:

Month	Priority
Jan	1
Feb	2
Mar	3
Apr	4
May	5
Jun	6
Jul	7
Aug	8
Sep	9
Oct	10
Nov	11
Dec	12

Number of characters to use from field How many characters in the specified rank field should be used in considering a record's priority.

Ignore case Whether or not case should be considered when looking up a rank in the list below.

Advanced rank assignment can be done in one of two ways:

- **Use an External File** If you have a pre-existing dBASE table with the desired ranks, or have many potential codes - in which case creating a table is a good idea, use this method. One advantage of this method is you can use the external table in other setups.
- **Rank in This Order** Enter a string of codes in ranking order from lowest to highest priority. When you don't have a lot of different codes in the selected field, or don't want to bother creating a new dBASE file, use this method.

Codes that don't appear on the above list should be Specify what to do when a record has a value in the Rank Field which is not present in the above list:

- **Not Processed (filtered)**
- **Assigned the lowest priority**
- **Assigned the highest priority**

4.2.3 Matchcode Mapping

Merge/Purge Setup

General | Input Tables | Matchcode Mapping | Output Field Mapping | Output Tables | Advanced

Field Mapping for Matchcode: **Address, Company OR Last Name**

Component	inhouse	national	suppress	household	nonprofit
Zip5	ZIP	CITYSTZIP	CSZ	ZIP	
Last Name	LAST	NAME		NAME	LAST
Company					
Address Line 1	STREET	STREET	ADDR1	STREET	STREET
Address Line 2			ADDR2		
Address Line 3					

↓

Matchcode Component:
 Data Type: **Zip5**
 Label:
 Size: **5**
 Start: **Left**
 Match Strategy: **Exact**
 Short/Empty: **Both Fields**
 Swap: **None**

↓

Input Field:
 Name: **ZIP**
 Field Type: **Fixed Character**
 Size: **10** Decimals:
 Data Type: **Zip9**
 Conversion: **Zip9 to Zip5**

View File Field List

Change Data Type...

Previous Next OK Cancel Help

Matchcode Mapping: Field Mapping for Matchcode

Field Mapping for Matchcode: **Address, Company OR Last Name**

Component	inhouse	national	suppress	household
Zip5	ZIP	CITYSTZIP	CSZ	ZIP
Last Name	LAST	NAME		NAME
Company				
Address Line 1	STREET	STREET	ADDR1	STREET
Address Line 2			ADDR2	
Address Line 3				

This screen tells MatchUp which field contains the data for each component of the matchcode.

These components are matched with fields from your input table. MatchUp tries to determine a field's type based on the field name. For example, MatchUp will default to "Address" if the field's name starts with "ADD", or "STR". You can customize these defaults in Field Naming. If MatchUp doesn't pick correctly, double-click on the incorrect field name and select the correct field.

The purple field names are fields that need to be parsed to get the needed matchcode component. So, even though this matchcode is using last name, MatchUp can use a full name field and find the information it needs.

And, although this matchcode uses street number, street name and PO Box, MatchUp does not ask for these components individually (as few databases provide data in that way). Instead, MatchUp asks for up to three address lines which it will parse internally.

Matchcode Mapping: Matchcode Component



Describes the currently highlighted matchcode component:

Data Type The currently selected matchcode component.

Size The number of characters this matchcode uses from the data.

Label A label that is attached to this component. MatchUp does not itself use this label, but it can be helpful in remembering what a particular General component was for. The label is specified in the Matchcode Editor.

Start Specifies where the matchcode extracts the data: from the Left, Right, or a specific position.

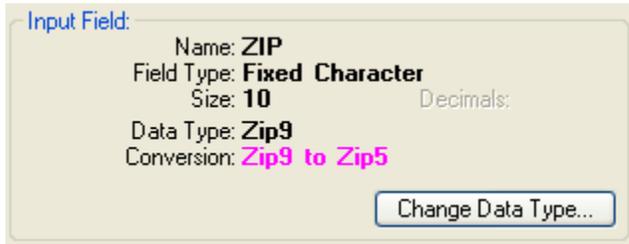
Match Strategy Lists the component's fuzzy matching capability (if any).

Short/Empty Should blank or short fields be considered as matching or not?

Swap If this component is part of a Swap Pair, the letter that was assigned to that pair ("A" through "H") will be listed.

See also: Matching Strategies

Matchcode Mapping: Input Field



The screenshot shows a dialog box titled "Input Field:" with the following configuration details:

- Name: ZIP
- Field Type: Fixed Character
- Size: 10
- Decimals: (empty)
- Data Type: Zip9
- Conversion: Zip9 to Zip5

At the bottom right of the dialog is a button labeled "Change Data Type..."

Describes the currently highlighted input field:

Name Name of the field.

Field Type Valid types are: Fixed Character, Variable Character, Integer, Float, Decimal, Logical, and DateTime.

Size Size of the field.

Decimals If the field type is numeric there may be a decimal.

Data Type The type of information contained in this source field. Make sure this is correct or the fields will not merge properly. MatchUp tries to determine a field's type based on the field name. For example, MatchUp will default to "Address" if the field's name starts with "ADD", or "STR". If MatchUp doesn't pick correctly, click **Change Data Type** and select the description which best suits this field. You can customize these defaults in Field Naming.

Conversion? Indicates what type of conversion will be done to get from the highlighted field's data type to the output field's data type. For example, an input field containing city, state and zip will have to be split to compare Zip codes.

Change Data Type Allows you to change the Data Type of an input field. Single right-click for a pop-up menu or double right-click for a dialog box.

4.2.4 Output Field Mapping

The screenshot shows the 'Merge/Purge Setup' dialog box with the 'Output Field Mapping' tab selected. The main table lists various fields and their mappings across different data sources. Below the table, the 'Output Field' and 'Input Field' configuration panels are visible, both showing the 'FIRST' field with a 'Fixed Character' type and 'First Name' data type. The 'Output Field' panel has the 'Gather' checkbox checked, while the 'Input Field' panel has 'Scatter to alternate field' checked.

Output Field	inhouse	national	suppress	Database	nonprofit
FIRST	FIRST	NAME		FIRST	FIRST
LAST	LAST	NAME		LAST	LAST
STREET	STREET	STREET	ADDR1	STREET	STREET
CITY	CITY	CITYSTZIP	CSZ	CITY	CITY
STATE	STATE	CITYSTZIP	CSZ	STATE	ST
ZIP	ZIP	CITYSTZIP	CSZ	ZIP	
LASTCALL	LASTCALL	LASTCALL		LASTCALL	
AGE	AGE	AGE		AGE	

Output Field:
 Name: **FIRST**
 Field Type: **Fixed Character**
 Size: **20** Decimals:
 Data Type: **First Name**
 Gather Method:

Input Field:
 Name: **FIRST**
 Field Type: **Fixed Character**
 Size: **20** Decimals:
 Data Type: **First Name**
 Conversion: None
 Gather
 Scatter to this field
 Scatter to alternate field:

Navigation buttons: Previous, Next, OK, Cancel, Help.

Output Field Mapping: Output Field Mapping

This screenshot is identical to the one above, showing the 'Merge/Purge Setup' dialog box with the 'Output Field Mapping' tab. It displays the table of field mappings and the configuration options for the 'FIRST' field.

Output Field	inhouse	national	suppress	Database	nonprofit
FIRST	FIRST	NAME		FIRST	FIRST
LAST	LAST	NAME		LAST	LAST
STREET	STREET	STREET	ADDR1	STREET	STREET
CITY	CITY	CITYSTZIP	CSZ	CITY	CITY
STATE	STATE	CITYSTZIP	CSZ	STATE	ST
ZIP	ZIP	CITYSTZIP	CSZ	ZIP	
LASTCALL	LASTCALL	LASTCALL		LASTCALL	
AGE	AGE	AGE		AGE	

The initial Output file's structure is determined in Default Output Table. You can modify the structure here, but you will probably want to start with one that is close to what you want for your output file.

Note that information sometimes has to be extracted from complex fields. For example,

under national.dbf above, First and Last will be extracted from Name.

Add Output Field Adds a new field(s) to the output file's structure

Remove Output Field Removes a field(s) from the output structure.

Check Mappings Use Check Mapping to ensure that you haven't forgotten any source file's fields.

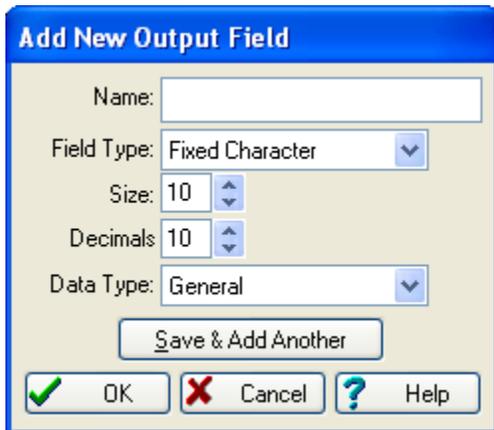
Check Truncations MatchUp warns you about the possibility of truncation here.

Global Scatter/Gather An easier way to Scatter/Gather...

View File Although you cannot edit the data, you can view your file here. This is a great way to make sure you have the right file!

Field List Shows a list of each field in each table. You can use your mouse to drag and drop fields on this list into the desired mappings on the Output Field Mapping Screen.

Output Field Mapping: Add Output Field



Name Name of the field.

Field Type Valid types are: Fixed Character, Variable Character, Integer, Float, Decimal, Logical, and DateTime.

Size Size of the field. Some Field Types have fixed sizes.

Decimals If the field type is numeric there might be a decimal.

Data Type The type of information contained in this output field. Make sure this is correct or the fields will not merge properly. MatchUp tries to determine a field's type based on the field name. For example, MatchUp will default to "Address" if the field's name starts with "ADD", or "STR".

Output Field Mapping: Output Field

Output Field:

Name: **FIRST**
Field Type: **Fixed Character**
Size: **20** Decimals:
Data Type: **First Name**

Gather Method:

Describes the currently highlighted output field:

Name Name of the field.

Field Type Valid types are: Fixed Character, Variable Character, Integer, Float, Decimal, Logical, and DateTime.

Size Size of the field.

Decimals If the field type is numeric there may be a decimal.

Data Type The type of information contained in this output field. Make sure this is correct or the fields will not merge properly.

Gather Check Gather if you want data gathered for the highlighted output field.

Method This option is not available unless Gather is checked for the highlighted output field. See Gathering Methods.

Change Output Field Allows you to change all the specifications of a field: Name, Field Type, Size, Decimals, and Data Type.

Output Field Mapping: Input Field

Input Field:

Name: **FIRST**
Field Type: **Fixed Character**
Size: **20** Decimals:
Data Type: **First Name**
Conversion: None

Gather
 Scatter to this field
 Scatter to alternate field:

Change Data Type...

Describes the currently highlighted input field:

Name Name of the field.

Field Type Valid types are: Fixed Character, Variable Character, Integer, Float, Decimal, Logical, and DateTime.

Size Size of the field.

Decimals If the field type is numeric there may be a decimal.

Data Type The type of information contained in this output field. Make sure this is correct or the fields will not merge properly. MatchUp tries to determine a field's type based on the field name. For example, MatchUp will default to "Address" if the field's name starts with "ADD", or "STR". If MatchUp doesn't pick correctly, click **Change Output Field** and select the description which best suits this field. You can customize these defaults in Field Naming.

Conversion? Indicates what type of conversion will be done to get from the highlighted field's data type to the output field's data type. For example, an input field containing city, state and zip will have to be split if the output table has separate city, state and zip fields.

Gather Check Gather if you want data from the highlighted input field to contribute in the gathering process for the highlighted output field. This option is not available if Gather is not checked for the highlighted output field.

In most cases, when you are gathering to an output field, you will want to get the data from each input file. But there are cases when this is not desirable. For instance, if you were gathering phone numbers, and one contributing database had outdated area codes, you would probably not want to gather phone numbers from that database.

Scatter to this field Check this box if you would like the deposit contents of the highlighted

output field to be deposited into the current field.

Scatter to alternate field Check this box if you would like to deposit the contents of the highlighted output field to an another input field.

Scattering is used to update source files (for example, updating phone numbers in a database). Note that you don't have to gather in order to scatter. In fact, it is often not desirable (like in a change of address update). Scattering is not available for read-only tables or Suppression tables.

Note that scattering is one of the only functions in MatchUp that will overwrite your data. Use this option with caution and always maintain a backup!

Change Data Type Allows you to change the Data Type of an input field. Single right-click for a pop-up menu or double right-click for a dialog box.

4.2.5 Output Tables

The screenshot shows the 'Merge/Purge Setup' dialog box with the 'Output Tables' tab selected. The dialog has a blue title bar and a standard Windows window border. The main area is divided into several sections for configuring output tables and fields.

	Table Type:	Database:	Table:
Output Table:	dBase	C:\D\Projects\	Output.dbf
Duplicate Table:	dBase	C:\D\Projects\	Dupe.dbf
Suppression Table:	<None>		
Non-Intersection Table:	<None>		

Below the table, there are several input fields and buttons:

- Status Field: STATUS
- Count Field: COUNT
- Output Matchcode Field: OUTPUTMC
- Dupe Group Field: DUPEGROUP
- Multi-Buyer Count Field: MBCOUNT
- Multi-Buyer Field: MULTIBUYER | 10
- Source Code Field: SC
- Multi-Buyer Format... button
- Source Code Format... button

At the bottom of the dialog, there is a 'CASS...' button with a postage stamp icon. The bottom-most row contains navigation buttons: Previous, Next, OK, Cancel, and Help.

Result Tables Allows you to specify where you'd like to put Output, Duplicate, Suppressed and Non-Intersected records. You can use the **Browse** button for the easiest way to select a table.

- **Output Table** (optional) Specify the table that will receive output records. Output records consist of unique records and one record from each duplicate group.
- **Duplicate Table** (optional) Specify the table that will receive duplicate records.
- **Suppression Table** (optional) Specify the table that will receive suppressed records. Suppressed records consist of Regular, Self Purge, and No Purge records that did not match against a Suppression list.
- **Non-Intersection Table** (optional) Specify the table that will receive non-intersected records. Non-intersected records consist of Regular, Self Purge, and No Purge records that did not match against an Intersection list.

Status Field (optional) Select a field name for capturing a record's disposition. During processing, this field will be populated with a code indicating whether this record is a dupe, has dupes, was suppressed, etc. See MatchUp Status Codes.

Output Matchcode Field (Advanced) (optional) Enter a field name for the Output Matchcode. During processing, MatchUp will put the record's matchcode in this field.

There are a few practical uses for the Output Matchcode Field:

- Reusing this matchcode speeds processing in future runs.
- You can use this matchcode as a lookup field in your own database programs.
- It can be used as one of those "secret codes" people sometimes print on "Anonymous" Business Reply Cards.

Multi-Buyer Count Field (Advanced) (optional) Specify a field to receive a record's multi-buyer count. During processing, this field will be populated with a number indicating how many inter-file dupes a record has. However, see Counting Method for alternate multi-buyer counting methods.

Source Code Field (Advanced) (optional) Specify a field that contains source information that should be used from this file when generating the Source Code Reports (See Reports).

Source Code Format Allows you to select the format of this code.

Count Field (optional) Select a field name to receive a record's dupe count. During processing, this count field will be populated with a number indicating how many dupes a record has. See Counting Method for the various counting methods.

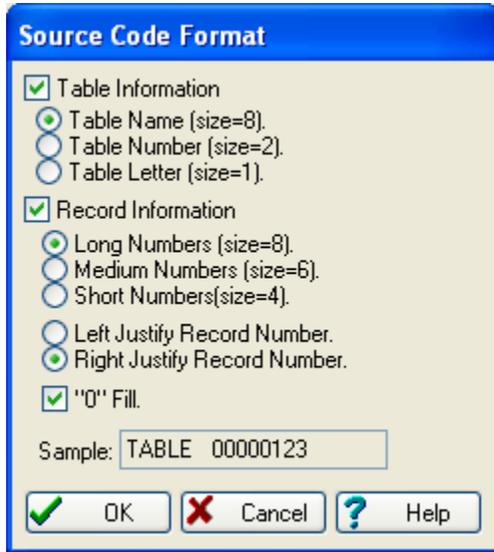
Dupe Group Field (Advanced) (optional) Enter a field name which will receive the dupe group number for this record. Each original record that MatchUp comes across is assigned a number. When a dupe is found, it is assigned the same number. See Dupe Groups.

Multi-Buyer Field (Advanced) (optional) Specify a field which will receive multi-buyer information for this record. See Multi-Buyer format.

Multi-Buyer Format Allows you to select the format of this code.

CASS Allows you to CASS and/or DPV the output table.

Output Tables: Source Code Format



Source tracking gives you a way to track what table (and optionally record number) a record in the output table came from. If you like, you can name a new field, and it will be added to the table. Table Information and Record Information allow you to format the information in the Source Code field.

Table Information Select how you would like to show which table a record is from:

- **Table Name** Uses up to 8 characters of the input table's name.
- **Table Number** Uses up to 2 numbers to store the input table number
- **Table Letter** Uses 1 letter to indicate which input table number

Record Information Select how you would like to indicate which record a record came from:

- **Long Numbers** Uses up to 8 numbers to store the input record number.
- **Medium Numbers** Uses up to 6 numbers to store the input record number.
- **Short Numbers** Uses up to 4 numbers to store the input record number.
- **Left Justify** Left justifies the record number.
- **Right Justify** Right justifies the record number. If you select Right Justify, you also have the option to "0" Fill.
- **"0" Fill** Fills the unused record information spaces with "0"s.

Sample Shows you what the Source Code Field will look like.

Output Tables: Multi-Buyer Format

Multi-Buyer Format

Source Code:
A key will be retrieved from the **Input Source Code Field** that was specified on the **Advanced Tab**. If no field was specified for a source, a File Letter will be used instead.

Table Letter.

Table Number:
 1 Digit.
 2 Digits.
 3 Digits.

Delimit sources with:
 No delimiter.
 Space.
 Comma.

Omit repeated sources.

Sample:

OK Cancel Help

Multi-Buyer tracking gives you a way to track what input table(s) the data in the output table came from.

Source Code A key will be retrieved from the Input Source Code Field (specified on the Advanced Tab), not the Source Code Field that you selected on the Output Tables screen. If no field was specified for a particular source table, a table letter will be used instead.

Table Letter Uses 1 letter to indicate the input table.

Table Number Uses up to 1, 2, or 3 digits to store the input table number. You must specify the number of digits you wish to use.

Delimit sources with Select how you would like to show which table a record is from: no delimited, space, or comma.

Omit repeated sources Only report a source once for each table.

Sample Shows you what the Multi-Buyer data will look like.

Using the Multi-Buyer Field:

Now that you have one, what do you do with it? Well for starters, it's great for counts. Say you're wondering how many output records had dupes in all three files. A simple **Tools | Browse | File Control | Count** "For" condition would be:

```
'A' $ MultiBuyer .AND. 'B' $ MultiBuyer .AND. 'C' $ MultiBuyer
```

You'll find the "\$" ("is contained in") operator very useful for these queries.

How about how many output records came from the second file and have dupes in the third file?

```
Left(MultiBuyer,1)='B' .AND. 'C' $ MultiBuyer
```

You can count on the first letter to represent the output record, and any following letters to represent dupe records.

Output Tables: CASS

Our CASS module checks, corrects and verifies addresses, by comparing them to the USPS national database. The verification process standardizes your address to USPS specifications by correcting misspellings, fixing nonstandard abbreviations, and adding missing data.

CASS: Input/Output Fields

CASS Setup

Input/Output Fields | Additional Output Fields | Options

CASS the Output Table

Input:			Output:	
Company:	CO		Company:	<None>
Address Line 1:	STR		Address Line 1:	C_ADDRESS
Address Line 2:	<None>		Address Line 2:	<None>
Suite:	<None>		Suite:	C_APT
City or City/St/Zip:	CITY	Specify same Field for all 4 entries to join into single output.	City:	C_CITY
State:	ST		State:	C_STATE
Zip:	ZIP		Zip:	C_ZIP5
Plus 4:	<None>		Plus 4:	C_ZIP4
Urbanization:	<None>		Urbanization:	URBAN
Carrier Route:	<None>		Carrier Route:	CR
Delivery Point Code:	<None>		Delivery Point Code:	DPC
DP Check Digit:	<None>		DP Check Digit:	DPCD

Copy Input >> Output

OK Cancel Help

CASS the Output Table Turns CASS on or off for the Output Table.

Input MatchUp determines which field corresponds with which input line. You must provide at least one address line, a city and a state for CASS processing. The more input information you provide, the better the results.

- **Company** Very large companies sometimes get their own Plus 4, so providing a company can increase accuracy.

- **Address Line 1 & 2** The delivery address lines.
- **Suite** The suite name and number.
- **City or City/St/Zip** The city, or if it is not pre-split, the entire city/state/zip.
- **State** The 2 letter state abbreviation.
- **Zip** The 5-digit zip code.
- **Plus 4** The zip plus 4 extension.
- **Urbanization** A description of an area, sector, or development within a geographic area. It is commonly used in urban Puerto Rico, as it describes the location of a given street.
- **Carrier Route** Delivery carrier route.
- **Delivery Point Code** A two digit string which describe the 10th and 11th positions of a 12-digit POSTNet barcode.
- **Delivery Point Check Digit** A single digit which describes the 12th position of a 12-digit POSTNet barcode.

Output Specify the fields that will receive the updated address information. You can elect to use the input fields or new fields:

- **Company**
- **Address Line 1 & 2**
- **Suite**
- **City** If you desire the output city, state and zip code to all be in the same field, specify the same output field for **City, State, Zip** and **Plus 4**.
- **State**
- **Zip**
- **Plus 4** If the **Zip** and **Plus 4** are the same field, the field will contain both codes.
- **Urbanization**
- **Carrier Route**
- **Delivery Point Code**
- **Delivery Point Check Digit** If this field is the same as the **Delivery Point Code**, it will be appended to that value.

Carrier Route:

The carrier route is a letter followed by three numbers (for example "R009" and "C039"). The alphabetic character describes the type of delivery:

Code	Description
B	PO Box
C	City Delivery
G	General Delivery
H	Highway Contract
R	Rural Route

CASS: Additional Output Fields

MatchUp will populate any of these additional fields from the CASS database:

The screenshot shows the 'CASS Setup' dialog box with the 'Additional Output Fields' tab selected. The dialog is divided into three sections: Demographics, Geographics, and Split Components. Each section contains several dropdown menus for selecting output fields. At the bottom right, there are three buttons: 'OK', 'Cancel', and 'Help'.

Section	Field Name	Selected Value
Demographics:	City Abbreviation:	CITY_ABBR
	County:	COUNTY
	County FIPS:	FIPS
	Time Zone:	TIMEZONE
	Time Zone Code:	TZC
	MSA Number:	MSA
	PMSA Number:	PMSA
Geographics:	Congressional Dist:	DISTRICT
	Latitude:	LATITUDE
Geographics:	Longitude:	LONGITUDE
	Split Components:	
	Number:	NUMBER
	Pre-Directional:	PREDIR
	Name:	NAME
	Suffix:	SUFFIX
	Post-Directional:	POSTDIR
	Secondary:	SECONDARY
	PO Box:	POBOX
Address Status:		
	LACS:	LACS
	EWS:	EWS
	Address Type:	ADDRTYPE
	DPV Footnotes:	FOOTNOTES

Demographics Geopolitical information:

- **City Abbreviation** City names longer than 13 letters will have the Postal Services' official abbreviation. If the city is 13 letters or less, the full city name is returned. For example, "Fort Lauderdale" for "Ft Lauderdale" and "Fredericksburg" for "Fredericksbrg".
- **County** The name of the County the verified address resides in.
- **County FIPS** Federal Information Processing Standard. A 5 digit code defined by the U.S. Census Bureau. The first two digits represent the state code, the last three representing the residing county.
- **Time Zone** A description of the zip code's time zone. example: "Eastern Time"
- **Time Zone Code** A 1 or 2 digit code representing the verified addresses' time zone (see below).

- **MSA Number** Metropolitan Statistical Area Number. A 4 digit code for a metropolitan area made up of one or more counties and meeting statistical criteria defined by the U.S. Office of Management and Budget (OMB). example: "1922"
- **PMSA Number** Primary Metropolitan Statistical Area. A 4 digit code for metropolitan regions with a population greater than one million.
- **Congressional Dist** A 2 digit number representing the congressional district associated with the verified address.

Geographics Note that latitude and longitude are accurate only to the 5-digit zip level, and not the plus 4.

- **Latitude** The 7-digit latitude coordinate (accurate to 4 decimal places) of the center of a Zip code.
- **Longitude** The 9-digit longitude of the center of a Zip code.

Split Components See Matchcode Components for more information on the components of a street address. The Split Components fields are populated regardless of whether or not an address has been successfully CASSed. If a record has been CASSed, these components will be populated with the parsed components as they exist in the USPS database. If not, MatchUp's internal street splitter will parse the components instead.

Address Status

- **LACS** Locatable Address Conversion Service. The total number of records which were flagged as having undergone a change to a city-style address (to allow emergency services to locate these addresses more efficiently). The address change is not reflected here.
- **EWS** Early Warning System Data. Count of new addresses scheduled to be included in the next release of the bimonthly USPS national database. New construction projects for example, are flagged by the EWS file.
- **Address Type** A single character code indicating the type of verified address.
- **DPV Footnotes** A set of one or more 2 character codes containing delivery point information.

Time Zone Code:

Code	Zone
(blank)	Military
4	Atlantic Time
5	Eastern Time
6	Central Time
7	Mountain Time
8	Pacific Time
9	Alaska time
10	Hawaii Time
11	Samoa Time
13	Marshall Island Time
14	Guam Time
15	Palau Time

Address Type:

Code	Description
(blank)	unverified
F	Company Address
G	General Delivery
H	High-Rise or business complex
P	PO Box
R	Rural Route
S	Residential Address

DPV Footnotes:

Code	Description
AA	Input address matched to the Zip+4 file
A1	Input address not matched to the Zip+4 file
BB	Input address matched to DPV (all components)
CC	Input address primary number (street number) matched to DPV but secondary number not matched
N1	Input address primary number matched to DPV but secondary number missing
M1	Input address primary number missing
M3	Input address primary number invalid
P1	Input address PO, RR or HC box number missing
P3	Input address PO, RR or HC box number invalid
RR	Input address matched to CMRA
R1	Input address matched to CMRA but secondary number not present
F1	Address was coded to a military address
G1	Address was coded to a General Delivery address
U1	Address was coded to a unique Zip Code

CASS: Options

The screenshot shows the 'CASS Setup' dialog box with the 'Options' tab selected. The 'Output Error Code' is set to 'ERROR' and the 'Output Status Code' is set to 'STATUS'. Under 'If there's a coding error...', the 'Clear the Input Plus 4 field' and 'Clear the Input Carrier Route field' options are unchecked. Under 'If joining City/State/Zip...', the 'Delimit City & State with a comma.' option is selected. The 'CASS database date' is 11/15/2006 and the 'CASS database expires' on 02/28/2007. The 'Perform Delivery Point Validation (DPV)' and 'Use CASSMate enhanced processing' options are checked. The 'Form 3553 Information' section contains the following fields: List Name: Output.dbf, Processor Name: Peoplesmith Software, Mailing Company: Peoplesmith Software, Address: 50 Cole Parkway, Suite 34, City: Scituate, State: MA, and Zip: 02066-1337. The dialog box has 'OK', 'Cancel', and 'Help' buttons at the bottom right.

Output Error Code Select the field you want to use for the error code.

Output Status Code Select the field you want to use for the status code.

If there's a coding error What should MatchUp do if it comes across a coding error (you can choose one or both):

- **Clear the Input Plus 4 field**
- **Clear the Input Carrier Route field**

City/State Delimiter If you elect to put city, state, and zip data into a single field, how would you like them delimited?

- **Delimit City & State with a space**
- **Delimit City & State with a comma**

Zip/Plus 4 Delimiter Same deal, but separating the Zip and Plus 4:

- **Delimit Zip & Plus 4 with a dash**
- **Delimit Zip & Plus 4 with a space**
- **Delimit Zip & Plus 4 with no delimiters**

Perform Delivery Point Validation (DPV) Check if you want to verify each address's delivery point during CASS verification.

Use CASSmate enhanced processing Whether or not to use the enhanced power of CASSmate to get more CASS matches. Enhanced CASSmate processing is only used when normal CASSing attempts have yielded no results.

Form 3553 information This is the general information necessary to fill out the USPS Form 3553 for the CASS certification. Be sure to fill in this information or your Form 3553 will not print out complete.

Error Codes:

Code	Description
(blank)	No Error
M	Multiple Matches More than one record matches the address and there is not enough information available in the input address to break the tie between multiple records. Passing more complete information, such as city names or urbanization names, can help reduce the number of multiple match errors.
N	No Street Data for Zip The Zip Code exists but no streets begin with the same letter in that Zip Code.
R	Range Error The address was found but the street number in the input address was not between the low and high range in the CASS database.
T	Component Error Either the directionals or the suffix field did not match the CASS database, and there was more than one choice for correcting the address. For example, if the given address was "100 Main St" and the only addresses found were "100 E Main St" and "100 Main Ave" the error code "T" would be returned because we do not know whether to add the directional "E" or to change the suffix to "Ave".
U	Unknown Street An exact street name match could not be found and phonetically matching the street name resulted in either no matches or matches to more than one street name.
X	Non-Deliverable Address The physical location exists but there are no homes on this street. One reason might be that there are railroad tracks or a river running alongside this street, as they would prevent construction of homes in this location.
Z	Invalid Zip Code The Zip Code does not exist and could not be determined by the city and state.
C	Canadian Postal Code The Zip Code matches the format characteristics of a Canadian Postal Code.
W	EWS File Record The Zip Code was found in the Early Warning System Data. These are new addresses scheduled to be included in the next USPS national database.

EWS:

Having a EWS file prevents new addresses from being miscoded with information from the (soon-to-be-outdated) USPS database. For example, an address of 44 Legacy Drive might be changed to 44 Legacy Street if Legacy Drive does not exist in the EWS File. However, if Legacy Drive is in the EWS File, the address is left alone (no Plus 4) and coded with an EWS Error. The next CASS database update should contain the new information and will code the address correctly.

The EWS file is updated weekly every Thursday. To get the most current file, you can download it from <ftp://www.MelissaData.com/updates/ews.txt>. Place the **ews.txt** file in the same folder as your other CASS Databases (mdAddr.dat, mdAddr.lic, etc.) overwriting the previous week's file.

Status Codes:

Code	Description
D	Demo Mode If processing with the demo CASS database, records not in the state of Nevada will be coded with "D".
E	Expired Database The CASS database has expired.
S	Standardized but Not Coded Standardization means that some conversion was done on the address (for example, changing "Post Office Box" to "PO Box").
V	Street Number Validated to DPV Level This record has been DPV validated. You can check the DPV Footnotes for more information about the level of validation.
X	Address Not Coded Check the Error code and/or DPV Footnotes for more information.
7	Multiple Matches There were multiple matches for the address that were all in the same Zip Code and Carrier Route. The returned Zip Code and Carrier Route will be correct but you will not get any Plus 4 information.
9	Fully Coded The address was fully CASS coded.

4.2.6 Advanced

The screenshot shows the 'Merge/Purge Setup' dialog box with the 'Advanced' tab selected. The 'Table Type' table lists three tables: 'dupe.dbf', 'inhouse.dbf' (selected), and 'national.dbf'. Below the table, the 'Information about inhouse.dbf:' section contains several configuration options:

Table Type	Source Type	Database/Connection	Table
Regular	dBase	C:\Dt\Projects\	dupe.dbf
Regular	dBase	C:\Dt\Projects\	inhouse.dbf
Regular	dBase	C:\Dt\Projects\	national.dbf

Information about inhouse.dbf:

Input Matchcode: <None> Output Dupe Group: <None>
Output Matchcode: <None> Input Source Code: <None>
Output Multi-Buyer Count: <None> (Use leftmost [] characters).

Input Filter: []

	Table Type:	Database:	Table:
Output Table:	<None>		
Duplicate Table:	<None>		
Suppression Table:	<None>		
Non-Intersection Table:	<None>		

Navigation buttons: Previous, Next, OK, Cancel, Help.

View File Although you cannot edit the data, you can view your file here.

Input Matchcode (Advanced) (optional) If you have used MatchUp to process this file before and saved the matchcode (See "Output Matchcode" below), you may use this matchcode instead of having the program build it again. Why? For lists that you process all the time, it's much faster to build a matchcode once and reuse it.

Input Matchcode Restrictions:

- The matchcode you used to build this matchcode must be the exact same one you are using now - MatchUp cannot verify this for you.
- This matchcode field does not automatically update if you change the contents of a record. You must adjust the matchcode accordingly. (With one exception - if you fill a record's Input Matchcode Field with "*****" (five asterisks), MatchUp will rebuild the Matchcode.

Output Matchcode (Advanced) (optional) Enter a field name for the Output Matchcode. During processing, MatchUp will put the record's matchcode in this field.

There are a few practical uses for the Output Matchcode Field:

- Reusing this matchcode speeds processing in future runs.
- You can use this matchcode as a lookup field in your own database programs.
- It can be used as one of those "secret codes" people sometimes print on "Anonymous" Business Reply Cards.

Output Multi-Buyer Count (Advanced) (optional) Enter a field name which will receive a record's multi-buyer count. During processing, this field will be populated with a number indicating how many inter-file dupes a record has. However, see Counting Method for alternate multi-buyer counting methods.

Output Dupe Group (Advanced) (optional) Enter a field name which will receive the dupe group number for this record. Each original record that MatchUp comes across is assigned a number. When a dupe is found, it is assigned the same number. (See Dupe Groups)

Input Source Code (Advanced) (optional) Specify a field that contains source information that should be used from this table when generating the Source Reports (See Reports).

Use leftmost ... characters (Advanced) Specify how many leading characters should be extracted from the Input Source Field and used in generating the Source Reports (See Reports).

Input Filter (Advanced) (optional) Enter a filter condition to limit records processed from the currently highlighted input table. This filter condition must be specified in dBASE Syntax or you can access the Expression Builder.

Result Tables Allows you to specify where you'd like to put Output, Duplicate, Suppressed and Non-Intersected records. You can use the **Browse** button for the easiest way to select a table. Records placed in these tables will be specific to the currently highlighted input table. You can specify result table(s) for any input table(s).

- **Output Table (Advanced)** (optional) Specify the table that will receive output records from the highlighted input table. Output records consist of unique records and one record from each duplicate group.
- **Duplicate Table (Advanced)** (optional) Specify the table that will receive duplicate records from the highlighted input table.
- **Suppression Table (Advanced)** (optional) Specify the table that will receive suppressed records from the highlighted input table. Suppressed records consist of Regular, Self Purge, and No Purge records that did not match against a Suppression list.
- **Non-Intersection Table (Advanced)** (optional) Specify the table that will receive non-intersected records from the highlighted input table. Non-intersected records consist of Regular, Self Purge, and No Purge records that did not match against an Intersection list.

4.3 Purge Setup

Select **Purge** if you want to take one (or several) tables and remove matching records.

4.3.1 General

The screenshot shows the 'Purge Setup' dialog box with the 'General' tab selected. The 'Matchcode' is set to 'Address, Company OR Last Name'. Below this, a table lists the fields included in the matchcode. The 'Matching Rules' section lists four rules. The 'Ranking' section has four radio button options, with the second one selected. The 'Order' section has two radio button options, with the first one selected. The 'When matching records have the same rank...' section has three radio button options, with the first one selected. At the bottom, there are buttons for 'Previous', 'Next', 'OK', 'Cancel', and 'Help'.

Matchcode: Address, Company OR Last Name Matchcode Editor...

Address, Company OR Last Name:

Data Type	Size	Start	Fuzzy	Short/Empty	Swap	1	2	3	4	5	6	7	8	9	10	11	1
Zip5	5	Left	Exact	Both Fields	None	X	X	X	X								
Last Name	10	Left	Exact	Both Fields	None	X	X										
Company	10	Left	Exact	Both Fields	None			X	X								
Street Number	5	Left	Exact	Both Fields	None	X		X									

Matching Rules:

- 1: Z5+LN+S#+SN
- 2: Z5+LN+POB
- 3: Z5+Co+S#+SN
- 4: Z5+Co+POB

Ranking:

- MatchUp will select the best record.
- I will assign priorities or specify a priority field.
- Prioritization will be random for each record.
- A uniform distribution of records will be selected.
- Doesn't matter.

Order:

- Ascending ("A" has a higher priority than "B").
A over B, 01 over 02, 3/10/68 over 4/10/68
- Descending ("A" has a lower priority than "B").
B over A, 02 over 01, 4/10/68 over 3/10/68

When matching records have the same rank...

- the first record processed will be selected.
- the last record processed will be selected.
- a random record will be selected.

Navigation: Previous, Next, OK, Cancel, Help

General: Matchcode

Matchcode: Matchcode Editor...

Address, Company OR Last Name:

Data Type	Size	Start	Fuzzy	Short/Em...	S...	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Zip5	5	Left	Exact	Both Fields	None	X	X	X	X												
Last Name	10	Left	Exact	Both Fields	None	X	X														
Company	10	Left	Exact	Both Fields	None			X	X												
Street Number	5	Left	Exact	Both Fields	None	X		X													
Street Name	10	Left	Exact	None	None	X		X													
PO Box	10	Left	Exact	None	None		X		X												

Matching Rules:
 1: Z5+LN+S#+SN
 2: Z5+LN+POB
 3: Z5+Co+S#+SN
 4: Z5+Co+POB

Select the matchcode that best suits your needs. A selection is supplied, and you will probably find a suitable one from this collection. If not, you can click Matchcode Editor to create (or edit) a matchcode.

The table above lists the properties of the currently selected matchcode. It is for viewing purposes only.

Matching Rules: MatchUp uses up to 16 matchcode "combinations" which are evaluated simultaneously. In the table, the X's in the columns marked 1 through 16 define component combinations that should be considered a match. The Matching Rules section simply displays them in an alternate form.

General: Ranking

Ranking:	Order:
<input type="radio"/> MatchUp will select the best record.	<input checked="" type="radio"/> Ascending ("A" has a higher priority than "B"). A over B, 01 over 02, 3/10/68 over 4/10/68
<input checked="" type="radio"/> I will assign priorities or specify a priority field.	<input type="radio"/> Descending ("A" has a lower priority than "B"). B over A, 02 over 01, 4/10/68 over 3/10/68
<input type="radio"/> Prioritization will be random for each record.	
<input type="radio"/> A uniform distribution of records will be selected.	
<input type="radio"/> Doesn't matter.	
	When matching records have the same rank...
	<input checked="" type="radio"/> the first record processed will be selected.
	<input type="radio"/> the last record processed will be selected.
	<input type="radio"/> a random record will be selected.

Ranking Allows you to choose which records to favor when dupes are found:

- **MatchUp will select the best record** Take the record with the most complete information.
- **I will assign priorities or specify a priority field** Allows you to choose which table or records to favor when duplicate records are found. If you select this option, you will be able to assign a rank or rank field for each input table on the Input Tables tab.
- **Prioritization will be random for each record** MatchUp will randomly select a record to use as the best record when duplicate records are found.
- **A uniform distribution of records will be selected** Instructs MatchUp to try (as best it can) to assign output records uniformly from all input files.
- **Doesn't Matter** Take the first record you come across.

What's the difference between Random and Uniform?

Random ranking tends to take an appropriate proportion of output records from each source. Uniform Distribution tries to take an equal quantity of output records from each source. For example, if there were 100 output records:

	Total Records	Random	Uniform
File1	9,000	90	50
File2	1,000	10	50

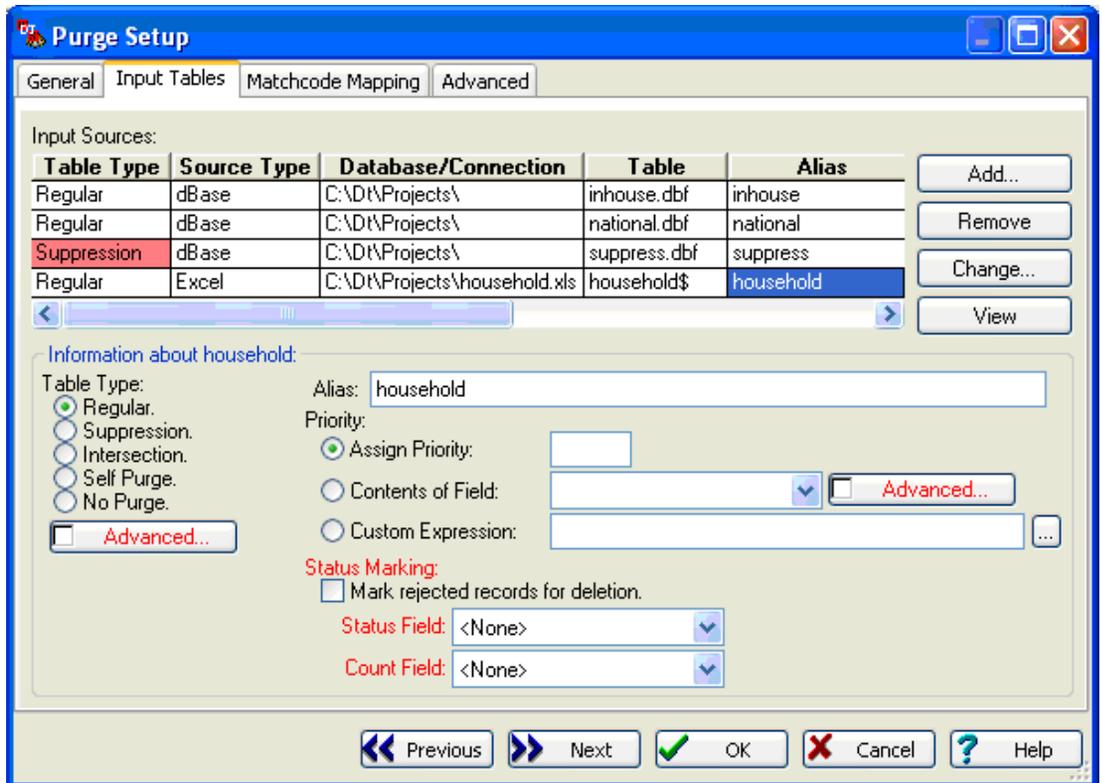
Ascending MatchUp will select records with a lower rank over records with a higher rank (i.e., "A" over "B", "1" over "2", and "03/10/68" over "04/10/68").

Descending MatchUp will select records with a higher rank over records with a lower rank (i.e., "C" over "B", "3" over "2", and "05/10/68" over "04/10/68").

When matching records have the same rank If MatchUp can't decide which record to use because they both have the same rank (ie, a tie), a backup plan can be selected:

- **The first record processed will be selected**
- **The last record processed will be selected**
- **A random record will be selected**

4.3.2 Input Tables



Input Tables: Input Sources

Input Sources:

File Type	Source Type	Database/Connection	Table	Alias
Regular	dBase	C:\dt3\Projects\	inhouse.dbf	inhouse
Regular	dBase	C:\dt3\Projects\	national.dbf	national
Suppression	dBase	C:\dt3\Projects\	suppress.dbf	suppress
Regular	Excel	C:\dt3\Projects\household.xls	household\$	household
Regular	ASCII Delimited	C:\dt3\Projects\	nonprofit.csv	nonprofit

Buttons: Add..., Remove, Change..., View

This screen displays the tables you have added.

You can change the placement of an input table by clicking and dragging it into the desired position. Alternately, you can press CTRL+UP ARROW or CTRL+DOWN ARROW to change the table's position. Note that the order of input tables has no impact on processing, but is purely for your benefit.

Add... Allows you to add an additional table to the setup.

Remove Removes a table from the merge/purge setup.

Change... Allows you to replace a table in the merge/purge setup with another table.

View Although you cannot edit the data, you can view your table here.

Input Tables: Information About

Information about Suppress:

Table Type:

Regular.

Suppression.

Intersection.

Self Purge.

No Purge.

Advanced...

Alias:

Priority:

Assign Priority:

Contents of Field: Advanced...

Custom Expression:

Status Marking:

Mark rejected records for deletion.

Status Field:

Count Field:

Table Type Select how the highlighted table will be handled:

- **Regular** This table will be deduped against itself and any other Regular tables.
- **Suppression** Records in Regular, No Purge, and Self Purge tables which also appear

in a Suppression list will be rejected. Ideal for "do not mail" lists.

- **Intersection** Records in Regular, No Purge, and Self Purge tables which do not appear on an Intersect list will be rejected. (opposite of Suppression).
- **Self Purge** This table will be matched against itself, Suppression, and Intersection tables, but not against Regular, No Purge, or other Self Purge tables. Self Purge tables are used when you would like to match several tables against a Suppression list, and purge internal dupes, but not match the tables against each other.
- **No Purge** This table will not be deduped against any other table. It will, however, be matched against Suppression and Intersection tables. No Purge tables are used when you would like to match several tables against a Suppression list, but not dedupe the tables internally or against each other.
- **Advanced** Click if the contents of a field dictate the Table Type.

Alias Enter a name you would like to refer to this table. This name will appear on all reports.

Priority Select a ranking method for the highlighted table:

- **Assign Priority** Enter a rank for this table. This option only appears if you selected I will assign priorities or specify a priority field under the General tab.
- **Contents of Field** Enter the name of the rank field. This option only appears if you selected I will assign priorities or specify a priority field under the General tab. This field must already exist in your table.
- **Advanced Rank** Check this option when the values in the specified Field are not easily compared.
- **Custom Expression** Enter a valid dBASE expression for evaluating rank. Use it when you need a tighter criteria than a single field or table will allow. For example, if you want to mail to the oldest female in a household, enter the expression: SEX + DtoS(BIRTHDATE). Only appears if you selected I will assign priorities or specify a priority field under the General tab.

Status Marking You can select methods of marking a record's disposition with these options:

- **Mark rejected records for deletion** If checked, rejected records will be marked for deletion. This can only be done with dBASE tables.
- **Status Field** (optional) Select a field name for capturing a record's disposition. During processing, this field will be populated with a code indicating whether this record is a dupe, has dupes, was suppressed, etc. See MatchUp Status Codes.
- **Count Field** (optional) Select a field name to receive a record's dupe count. During processing, this count field will be populated with a number indicating how many dupes a record has. See Counting Method.

Information About: Advanced Table Type

Sometimes tables are created by a Merge process and contain a code field specifying their original source. Before MatchUp, to process different source codes as different table types, users would have to 'undo' the merged file into separate files, then Merge/Purge them. Now you can specify the table type as Advanced and specify the field containing the code indicating how the record should be processed.

Advanced Table Type Settings for inHouse.dbf:

Base Table Type selection on contents of field: STATUS

Number of characters to use from field: 3 Ignore case
(when specifying multiple codes, separate each code with a comma)

Table Type 'Regular' when code is: SUB

Table Type 'Suppression' when code is: DMA

Table Type 'Intersection' when code is:

Table Type 'Self Purge' when code is:

Table Type 'No Purge' when code is:

Any codes not in any of the above lists should be processed:

Not Processed (filtered).

Table Type 'Regular'.

Table Type 'Suppression'.

Table Type 'Intersection'.

Table Type 'Self Purge'.

Table Type 'No Purge'.

Previous Next OK Cancel Help

Base Table Type selection on contents of field The field containing a code indicating how a record should be processed.

Number of characters to use from field How many characters in the above field should be used in considering how a record should be processed.

Ignore case Whether or not case should be considered when looking up a code in the lists below.

Table Type 'Regular' when code is A list of code(s) that indicate that a record should be processed as a regular record.

Table Type 'Suppression' when code is A list of code(s) that indicate that a record should be processed as a suppression record.

Table Type 'Intersection' when code is A list of code(s) that indicate that a record should be processed as an intersection record.

Table Type 'Self Purge' when code is A list of code(s) that indicate that a record should be processed as a self purge record.

Table Type 'No Purge' when code is A list of code(s) that indicate that a record should be processed as a no purge record.

Any codes not in any of the above lists should be processed Other codes will be processed in the following manner:

- **Not Processed (filtered)**
- **Table Type 'Regular'**
- **Table Type 'Suppression'**
- **Table Type 'Intersection'**
- **Table Type 'Self Purge'**
- **Table Type 'No Purge'**

Information About: Advanced Rank

When the field chosen to provide the ranking has values which can not easily be compared, Advanced Ranking can provide an order to those values.

For example, a rank field containing Months would wind up with unusual prioritization if the values were compared by character strings: Apr, Dec, Feb, Jan, Jul, Jun, etc.

Advanced Rank Settings for national.dbf:

Map the value in the Rank Field to an alternate set of priorities.

Number of characters to use from field: Ignore case

Use an external file listing ranks and priorities:



This list should be a two-field dBase file. The first field will be used as the Lookup Rank, the second field, the Desired Priority.

Prioritize ranks in this order:

The above list should list rank codes, from lowest to highest priority. Separate each code with a comma.

Codes that don't appear in the above list should be:

Not Processed (filtered).

Assigned the lowest priority.

Assigned the highest priority.

When the list of prioritized ranks is quite long, it can be get tedious to manually enter a list of ranks in the manner specified above. A more efficient method is to create a two-field lookup table (dBASE III format) and specify it using the first option, **Use an external file listing ranks and priorities**. In our example, the lookup table would look like the following:

Month	Priority
Jan	1
Feb	2
Mar	3
Apr	4
May	5
Jun	6
Jul	7
Aug	8
Sep	9
Oct	10
Nov	11
Dec	12

Number of characters to use from field How many characters in the specified rank field should be used in considering a record's priority.

Ignore case Whether or not case should be considered when looking up a rank in the list below.

Advanced rank assignment can be done in one of two ways:

- **Use an External File** If you have a pre-existing dBASE table with the desired ranks, or have many potential codes - in which case creating a table is a good idea, use this method. One advantage of this method is you can use the external table in other setups.
- **Rank in This Order** Enter a string of codes in ranking order from lowest to highest priority. When you don't have a lot of different codes in the selected field, or don't want to bother creating a new dBASE file, use this method.

Codes that don't appear on the above list should be Specify what to do when a record has a value in the Rank Field which is not present in the above list:

- **Not Processed (filtered)**
- **Assigned the lowest priority**
- **Assigned the highest priority**

4.3.3 Matchcode Mapping

Purge Setup

General | Input Tables | **Matchcode Mapping** | Advanced

Field Mapping for Matchcode: **Address, Company OR Last Name**

Component	inhouse	national	suppress	household
Zip5	ZIP	CITYSTZIP	CSZ	ZIP
Last Name	LAST	NAME		NAME
Company				
Address Line 1	STREET	STREET	ADDR1	STREET
Address Line 2			ADDR2	
Address Line 3				

Matchcode Component:

- Data Type: **Last Name**
- Label:
- Size: **10**
- Start: **Left**
- Match Strategy: **Exact**
- Short/Empty: **Both Fields**
- Swap: **None**

Input Field:

- Name: **NAME**
- Field Type: **Fixed Character**
- Size: **30** Decimals:
- Data Type: **Full Name**
- Conversion: **Full to LN**

View File | Field List

Change Data Type...

Previous | Next | OK | Cancel | Help

Matchcode Mapping: Field Mapping for Matchcode

Field Mapping for Matchcode: **Address, Company OR Last Name**

Component	inhouse	national	suppress	household
Zip5	ZIP	CITYSTZIP	CSZ	ZIP
Last Name	LAST	NAME		NAME
Company				
Address Line 1	STREET	STREET	ADDR1	STREET
Address Line 2			ADDR2	
Address Line 3				

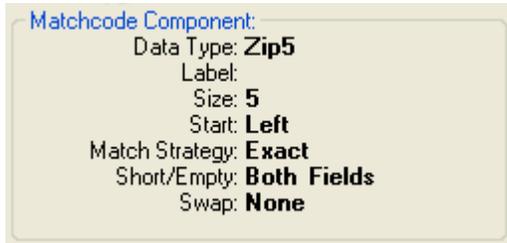
This screen tells MatchUp which field contains the data for each component of the matchcode.

These components are matched with fields from your input table. MatchUp tries to determine a field's type based on the field name. For example, MatchUp will default to "Address" if the field's name starts with "ADD", or "STR". You can customize these defaults in Field Naming. If MatchUp doesn't pick correctly, double-click on the incorrect field name and select the correct field.

The purple field names are fields that need to be parsed to get the needed matchcode component. So, even though this matchcode is using last name, MatchUp can use a full name field and find the information it needs.

And, although this matchcode uses street number, street name and PO Box, MatchUp does not ask for these components individually (as few databases provide data in that way). Instead, MatchUp asks for up to three address lines which it will parse internally.

Matchcode Mapping: Matchcode Component



Describes the currently highlighted matchcode component:

Data Type The currently selected matchcode component.

Size The number of characters this matchcode uses from the data.

Label A label that is attached to this component. MatchUp does not itself use this label, but it can be helpful in remembering what a particular General component was for. The label is specified in the Matchcode Editor.

Start Specifies where the matchcode extracts the data: from the Left, Right, or a specific position.

Match Strategy Lists the component's fuzzy matching capability (if any).

Short/Empty Should blank or short fields be considered as matching or not?

Swap If this component is part of a Swap Pair, the letter that was assigned to that pair ("A" through "H") will be listed.

See also: Matching Strategies

Matchcode Mapping: Input Field

Input Field:

Name: **ZIP**
Field Type: **Fixed Character**
Size: **10** Decimals:
Data Type: **Zip9**
Conversion: **Zip9 to Zip5**

Change Data Type...

Describes the currently highlighted input field:

Name Name of the field.

Field Type Valid types are: Fixed Character, Variable Character, Integer, Float, Decimal, Logical, and DateTime.

Size Size of the field.

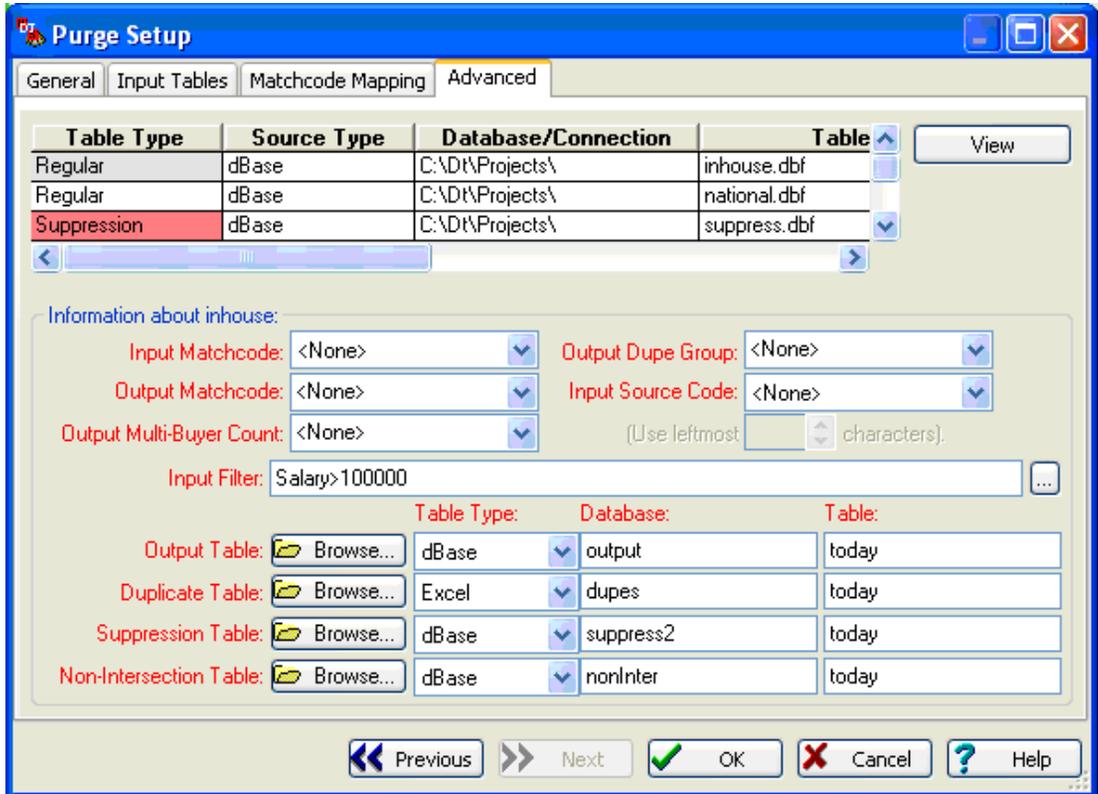
Decimals If the field type is numeric there may be a decimal.

Data Type The type of information contained in this source field. Make sure this is correct or the fields will not merge properly. MatchUp tries to determine a field's type based on the field name. For example, MatchUp will default to "Address" if the field's name starts with "ADD", or "STR". If MatchUp doesn't pick correctly, click **Change Data Type** and select the description which best suits this field. You can customize these defaults in Field Naming.

Conversion? Indicates what type of conversion will be done to get from the highlighted field's data type to the output field's data type. For example, an input field containing city, state and zip will have to be split to compare Zip codes.

Change Data Type Allows you to change the Data Type of an input field. Single right-click for a pop-up menu or double right-click for a dialog box.

4.3.4 Advanced



View File Although you cannot edit the data, you can view your file here.

Input Matchcode (Advanced) (optional) If you have used MatchUp to process this file before and saved the matchcode (See "Output Matchcode" below), you may use this matchcode instead of having the program build it again. Why? For lists that you process all the time, it's much faster to build a matchcode once and reuse it.

Input Matchcode Restrictions:

- The matchcode you used to build this matchcode must be the exact same one you are using now - MatchUp cannot verify this for you.
- This matchcode field does not automatically update if you change the contents of a record. You must adjust the matchcode accordingly. (With one exception - if you fill a record's Input Matchcode Field with "*****" (five asterisks), MatchUp will rebuild the Matchcode.

Output Matchcode (Advanced) (optional) Enter a field name for the Output Matchcode. During processing, MatchUp will put the record's matchcode in this field.

There are a few practical uses for the Output Matchcode Field:

- Reusing this matchcode speeds processing in future runs.
- You can use this matchcode as a lookup field in your own database programs.
- It can be used as one of those "secret codes" people sometimes print on "Anonymous" Business Reply Cards.

Output Multi-Buyer Count (Advanced) (optional) Enter a field name which will receive a record's multi-buyer count. During processing, this field will be populated with a number indicating how many inter-file dupes a record has. However, see Counting Method for alternate multi-buyer counting methods.

Output Dupe Group (Advanced) (optional) Enter a field name which will receive the dupe group number for this record. Each original record that MatchUp comes across is assigned a number. When a dupe is found, it is assigned the same number. (See Dupe Groups)

Input Source Code (Advanced) (optional) Specify a field that contains source information that should be used from this table when generating the Source Reports (See Reports).

Use leftmost ... characters (Advanced) Specify how many leading characters should be extracted from the Input Source Field and used in generating the Source Reports (See Reports).

Input Filter (Advanced) (optional) Enter a filter condition to limit records processed from the currently highlighted input table. This filter condition must be specified in dBASE Syntax or you can access the Expression Builder.

Result Tables Allows you to specify where you'd like to put Output, Duplicate, Suppressed and Non-Intersected records. You can use the **Browse** button for the easiest way to select a table. Records placed in these tables will be specific to the currently highlighted input table. You can specify result table(s) for any input table(s).

- **Output Table (Advanced)** (optional) Specify the table that will receive output records from the highlighted input table. Output records consist of unique records and one record from each duplicate group.
- **Duplicate Table (Advanced)** (optional) Specify the table that will receive duplicate records from the highlighted input table.
- **Suppression Table (Advanced)** (optional) Specify the table that will receive suppressed records from the highlighted input table. Suppressed records consist of Regular, Self Purge, and No Purge records that did not match against a Suppression list.
- **Non-Intersection Table (Advanced)** (optional) Specify the table that will receive non-intersected records from the highlighted input table. Non-intersected records consist of Regular, Self Purge, and No Purge records that did not match against an Intersection list.

4.4 File Update

File Update allows you to transfer data from one table to another. For example, if you have one table that contains addresses and phone numbers and would like to transfer these phone numbers to another table (containing matching addresses), you can do this through File Update.

Another popular usage for File Update is the updating of NCOA information. Say you send a table to the USPS for an NCOA or Move Update. Often, you won't send every field in your table, just the name and address fields. When you receive the updated list, File Update can be used to transfer addresses from the update list back into the original table. In these cases, it is advantageous to have a unique ID field in both tables, as address matching isn't a good idea for NCOA or Move updates.

4.4.1 General

Update Setup

General | Input Tables | Matchcode Mapping | Update

Matchcode:

Address, Company OR Last Name:

Data Type	Size	Start	Fuzzy	Short/Empty	Swap	1	2	3	4	5	6	7	8	9	10	11	1
Zip5	5	Left	Exact	Both Fields	None	X	X	X	X								
Last Name	10	Left	Exact	Both Fields	None	X	X										
Company	10	Left	Exact	Both Fields	None			X	X								
Street Number	5	Left	Exact	Both Fields	None	X		X									

Matching Rules:

- 1: Z5+LN+S#+SN
- 2: Z5+LN+POB
- 3: Z5+Co+S#+SN
- 4: Z5+Co+POB

Ranking:

- MatchUp will select the best record.
- I will assign priorities or specify a priority field.
- Prioritization will be random for each record.
- A uniform distribution of records will be selected.
- Doesn't matter.

Order:

- Ascending ("A" has a higher priority than "B").
A over B, 01 over 02, 3/10/68 over 4/10/68
- Descending ("A" has a lower priority than "B").
B over A, 02 over 01, 4/10/68 over 3/10/68

When matching records have the same rank...

- the first record processed will be selected.
- the last record processed will be selected.
- a random record will be selected.

General: Matchcode

Matchcode: Matchcode Editor...

Address, Company OR Last Name:

Data Type	Size	Start	Fuzzy	Short/Em...	S...	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Zip5	5	Left	Exact	Both Fields	None	X	X	X	X												
Last Name	10	Left	Exact	Both Fields	None	X	X														
Company	10	Left	Exact	Both Fields	None			X	X												
Street Number	5	Left	Exact	Both Fields	None	X		X													
Street Name	10	Left	Exact	None	None	X		X													
PO Box	10	Left	Exact	None	None		X		X												

Matching Rules:
 1: Z5+LN+S#+SN
 2: Z5+LN+POB
 3: Z5+Co+S#+SN
 4: Z5+Co+POB

Select the matchcode that best suits your needs. A selection is supplied, and you will probably find a suitable one from this collection. If not, you can click Matchcode Editor to create (or edit) a matchcode.

The table above lists the properties of the currently selected matchcode. It is for viewing purposes only.

Matching Rules: MatchUp uses up to 16 matchcode "combinations" which are evaluated simultaneously. In the table, the X's in the columns marked 1 through 16 define component combinations that should be considered a match. The Matching Rules section simply displays them in an alternate form.

General: Ranking

Ranking:	Order:
<input type="radio"/> MatchUp will select the best record. <input checked="" type="radio"/> I will assign priorities or specify a priority field. <input type="radio"/> Prioritization will be random for each record. <input type="radio"/> A uniform distribution of records will be selected. <input type="radio"/> Doesn't matter.	<input checked="" type="radio"/> Ascending ("A" has a higher priority than "B"). A over B, 01 over 02, 3/10/68 over 4/10/68 <input type="radio"/> Descending ("A" has a lower priority than "B"). B over A, 02 over 01, 4/10/68 over 3/10/68 When matching records have the same rank... <input checked="" type="radio"/> the first record processed will be selected. <input type="radio"/> the last record processed will be selected. <input type="radio"/> a random record will be selected.

Ranking Allows you to choose which records to favor when dupes are found:

- **MatchUp will select the best record** Take the record with the most complete information.
- **I will assign priorities or specify a priority field** Allows you to choose which table or records to favor when duplicate records are found. If you select this option, you will be able to assign a rank or rank field for each input table on the Input Tables tab.
- **Prioritization will be random for each record** MatchUp will randomly select a record to use as the best record when duplicate records are found.
- **A uniform distribution of records will be selected** Instructs MatchUp to try (as best it can) to assign output records uniformly from all input files.
- **Doesn't Matter** Take the first record you come across.

What's the difference between Random and Uniform?

Random ranking tends to take an appropriate proportion of output records from each source. Uniform Distribution tries to take an equal quantity of output records from each source. For example, if there were 100 output records:

	Total Records	Random	Uniform
File1	9,000	90	50
File2	1,000	10	50

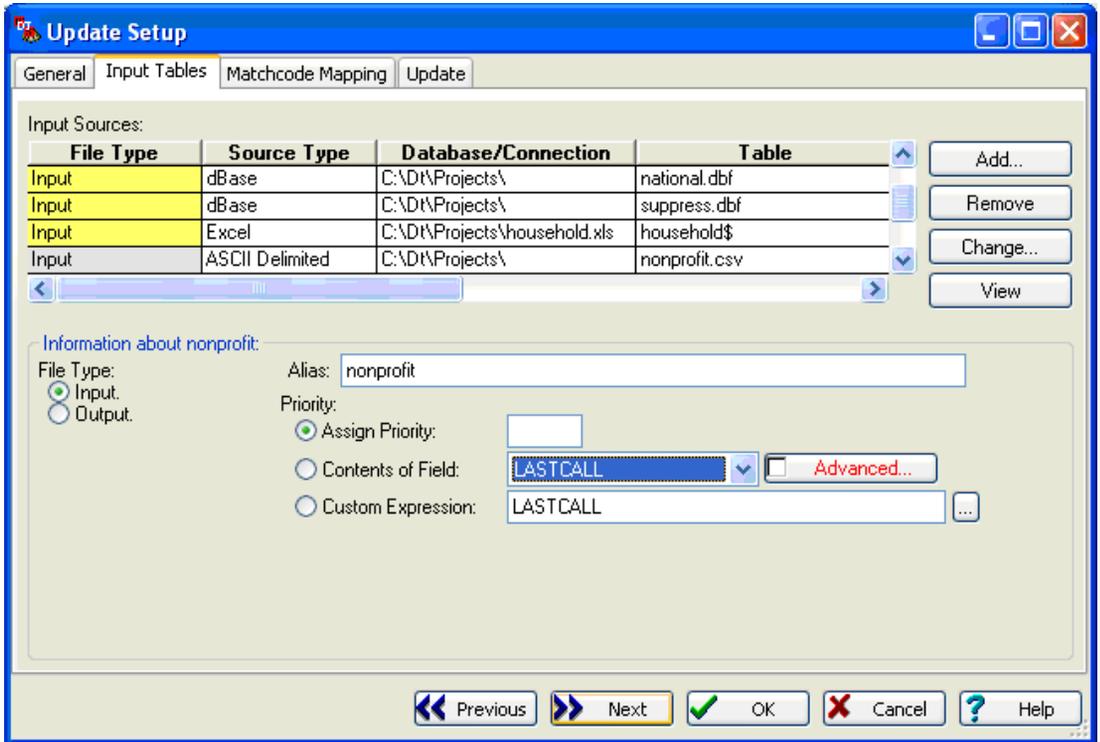
Ascending MatchUp will select records with a lower rank over records with a higher rank (i.e., "A" over "B", "1" over "2", and "03/10/68" over "04/10/68").

Descending MatchUp will select records with a higher rank over records with a lower rank (i.e., "C" over "B", "3" over "2", and "05/10/68" over "04/10/68").

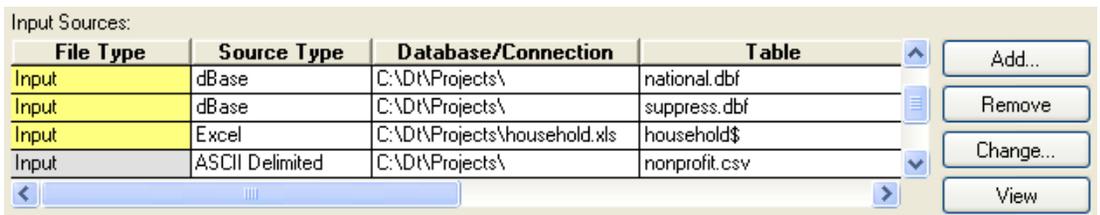
When matching records have the same rank If MatchUp can't decide which record to use because they both have the same rank (ie, a tie), a backup plan can be selected:

- **The first record processed will be selected**
- **The last record processed will be selected**
- **A random record will be selected**

4.4.2 Input Tables



Input Tables: Input Sources



This screen displays the tables you have added.

You can change the placement of an input table by clicking and dragging it into the desired position. Alternately, you can press CTRL+UP ARROW or CTRL+DOWN ARROW to change the table's position. Note that the order of input tables has no impact on processing, but is purely for your benefit.

Add... Allows you to add an additional table to the setup.

Remove Removes a table from the merge/purge setup.

Change... Allows you to replace a table in the merge/purge setup with another table.

View Although you cannot edit the data, you can view your table here.

Input Tables: Information About

Information about nonprofit:

File Type:
 Input.
 Output.

Alias: nonprofit

Priority:
 Assign Priority:
 Contents of Field: LASTCALL Advanced...
 Custom Expression: LASTCALL

Table Type Select how the highlighted table will be handled:

- **Input** This table will supply the data to update other tables.
- **Output** This table will updated with the data from the Input table(s). NOTE: You only get one(1) output table.

Alias Enter a name you would like to refer to this table. This name will appear on all reports.

Priority Select a ranking method for the highlighted table:

- **Assign Priority** Enter a rank for this table. This option only appears if you selected I will assign priorities or specify a priority field under the General tab.
- **Contents of Field** Enter the name of the rank field. This option only appears if you selected I will assign priorities or specify a priority field under the General tab. This field must already exist in your table.
- **Advanced Rank** Check this option when the values in the specified Field are not easily compared.
- **Custom Expression** Enter a valid dBASE expression for evaluating rank. Use it when you need a tighter criteria than a single field or table will allow. For example, if you want to mail to the oldest female in a household, enter the expression: SEX + DtoS(BIRTHDATE). Only appears if you selected I will assign priorities or specify a priority field under the General tab.

4.4.3 Matchcode Mapping

Update Setup

General | Input Tables | Matchcode Mapping | Update

Field Mapping for Matchcode: **Address, Company OR Last Name**

Component	national	suppress	household\$	nonprofit	local
Zip5	CITYSTZIP	CSZ	ZIP	ZIP	ZIP
Last Name	NAME		NAME	LAST	LAST
Company					
Address Line 1	STREET	ADDR1	STREET	STREET	STREET
Address Line 2		ADDR2			
Address Line 3					

↓

Matchcode Component:

- Data Type: **Zip5**
- Label:
- Size: **5**
- Start: **Left**
- Match Strategy: **Exact**
- Short/Empty: **Both Fields**
- Swap: **None**

↓

Input Field:

- Name: **CITYSTZIP**
- Field Type: **Fixed Character**
- Size: **30** Decimals:
- Data Type: **City/St/Zip**
- Conversion: **CSZ to Zip5**

View File Field List

Change Data Type...

Previous Next OK Cancel Help

Matchcode Mapping: Field Mapping for Matchcode

Field Mapping for Matchcode: **Address, Company OR Last Name**

Component	inhouse	national	suppress	household
Zip5	ZIP	CITYSTZIP	CSZ	ZIP
Last Name	LAST	NAME		NAME
Company				
Address Line 1	STREET	STREET	ADDR1	STREET
Address Line 2			ADDR2	
Address Line 3				

This screen tells MatchUp which field contains the data for each component of the matchcode.

These components are matched with fields from your input table. MatchUp tries to determine a field's type based on the field name. For example, MatchUp will default to "Address" if the field's name starts with "ADD", or "STR". You can customize these defaults in Field Naming. If MatchUp doesn't pick correctly, double-click on the incorrect field name and select the correct field.

The purple field names are fields that need to be parsed to get the needed matchcode component. So, even though this matchcode is using last name, MatchUp can use a full name field and find the information it needs.

And, although this matchcode uses street number, street name and PO Box, MatchUp does not ask for these components individually (as few databases provide data in that way). Instead, MatchUp asks for up to three address lines which it will parse internally.

Matchcode Mapping: Matchcode Component



Describes the currently highlighted matchcode component:

Data Type The currently selected matchcode component.

Size The number of characters this matchcode uses from the data.

Label A label that is attached to this component. MatchUp does not itself use this label, but it can be helpful in remembering what a particular General component was for. The label is specified in the Matchcode Editor.

Start Specifies where the matchcode extracts the data: from the Left, Right, or a specific position.

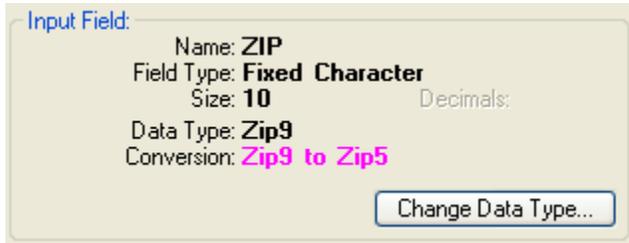
Match Strategy Lists the component's fuzzy matching capability (if any).

Short/Empty Should blank or short fields be considered as matching or not?

Swap If this component is part of a Swap Pair, the letter that was assigned to that pair ("A" through "H") will be listed.

See also: Matching Strategies

Matchcode Mapping: Input Field



Describes the currently highlighted input field:

Name Name of the field.

Field Type Valid types are: Fixed Character, Variable Character, Integer, Float, Decimal, Logical, and DateTime.

Size Size of the field.

Decimals If the field type is numeric there may be a decimal.

Data Type The type of information contained in this source field. Make sure this is correct or the fields will not merge properly. MatchUp tries to determine a field's type based on the field name. For example, MatchUp will default to "Address" if the field's name starts with "ADD", or "STR". If MatchUp doesn't pick correctly, click **Change Data Type** and select the description which best suits this field. You can customize these defaults in Field Naming.

Conversion? Indicates what type of conversion will be done to get from the highlighted field's data type to the output field's data type. For example, an input field containing city, state and zip will have to be split to compare Zip codes.

Change Data Type Allows you to change the Data Type of an input field. Single right-click for a pop-up menu or double right-click for a dialog box.

4.4.4 Update

Update Setup

General | Input Tables | Matchcode Mapping | **Update**

Input Sources				Output Source
national	suppress	household\$	nonprofit	local
NAME		NAME	FIRST	FIRST
NAME		NAME	LAST	LAST
STREET	ADDR1	STREET	STREET	STREET
CITYSTZIP	CSZ	CITY	CITY	CITY
CITYSTZIP	CSZ	STATE	ST	STATE
CITYSTZIP	CSZ	ZIP	ZIP	ZIP

Global Update...
View

↓

Input Field:

Name: **NAME**
Field Type: **Fixed Character**
Size: **30** Decimals:
Data Type: **Full Name**
Conversion: **Full to FN**

Change Data Type...

↓

Output Field:

Name: **FIRST**
Field Type: **Fixed Character**
Size: **20** Decimals:
Data Type: **First Name**
Update Method: **Join**

Change Data Type...

Previous Next OK Cancel Help

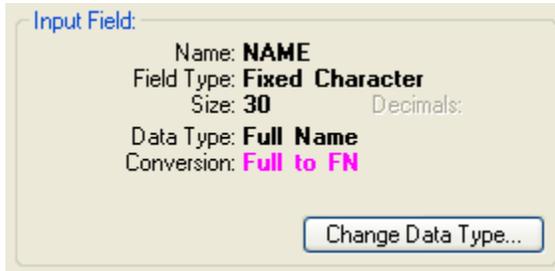
Input Sources These tables will supply the data to output table.

Output Source This table will be updated with the data from the Input table(s).

Global Update An easier way to make multiple update settings.

View Although you cannot edit the data, you can view your table here.

Update: Input Field



Describes the currently highlighted input field:

Name Name of the field.

Field Type Valid types are: Fixed Character, Variable Character, Integer, Float, Decimal, Logical, and DateTime.

Size Size of the field.

Decimals If the field type is numeric there may be a decimal.

Data Type The type of information contained in this output field. Make sure this is correct or the fields will not merge properly. MatchUp tries to determine a field's type based on the field name. For example, MatchUp will default to "Address" if the field's name starts with "ADD", or "STR". If MatchUp doesn't pick correctly, click **Change Output Field** and select the description which best suits this field. You can customize these defaults in Field Naming.

Conversion? Indicates what type of conversion will be done to get from the highlighted field's data type to the output field's data type. For example, an input field containing city, state and zip will have to be split if the output table has separate city, state and zip fields.

Change Data Type Allows you to change the Data Type of an input field. Single right-click for a pop-up menu or double right-click for a dialog box.

Update: Output Field

Output Field:

Name: **FIRST**

Field Type: **Fixed Character**

Size: **20** Decimals:

Data Type: **First Name**

Update Method: Join

Describes the currently highlighted output field:

Name Name of the field.

Field Type Valid types are: Fixed Character, Variable Character, Integer, Float, Decimal, Logical, and DateTime.

Size Size of the field.

Decimals If the field type is numeric there may be a decimal.

Data Type The type of information contained in this output field. Make sure this is correct or the fields will not merge properly.

Update Method Select the method of updating your data. See Gathering Methods for more information.

Change Data Type Allows you to change the Data Type of an output field. Single right-click for a pop-up menu or double right-click for a dialog box.

Update: Global Update

When you need to set several update settings, it can get quite tedious to set up each table. To make this process less painful, a global update dialog is provided.

Global Update

Step 1: Highlight the Fields below to be changed:

FIRST	(Join)
LAST	(No Gathering)
STREET	(No Gathering)
CITY	(No Gathering)
STATE	(No Gathering)
ZIP	(No Gathering)
LASTCALL	(No Gathering)
SALARY	(No Gathering)
AGE	(No Gathering)

Step 2: Select the Update Method to change them to:

Update Method:

Step 3: Click **Change!** to execute the changes.

Step 4: Return to step 1 or click **Close** if done.

Once you select a field(s) you want to update, check (or uncheck) **Update** and then select the **Method** you want to use. To execute your selections, click **Change!**

4.5 CASS

Our CASS module checks, corrects and verifies addresses, by comparing them to the USPS national database. The verification process standardizes your address to USPS specifications by correcting misspellings, fixing nonstandard abbreviations, and adding missing data.

4.5.1 Input/Output Fields

Input:	Output:	Size
Company: <None>	Company: COMPANY	35
Address Line 1: STREET	Address Line 1: ADDR1	35
Address Line 2: <None>	Address Line 2: ADDR2	35
Suite: <None>	Suite: STE	15
City or City/St/Zip: CITYSTZIP	City: CITY	20
State: <None>	State: ST	2
Zip: <None>	Zip: ZIP	5
Plus 4: <None>	Plus 4: ZIP4	4
Urbanization: <None>	Urbanization: URBAN	20
Carrier Route: <None>	Carrier Route: CARRIER	20
Delivery Point Code: <None>	Delivery Point Code: DPC	20
DP Check Digit: <None>	DP Check Digit: DPCD	20

Specify same Field for all 4 entries to join into single output.

Navigation: Previous, Next, OK, Cancel, Help

Input Table Click **Browse** to select the table that you would like to CASS verify.

Input MatchUp determines which field corresponds with which input line. You must provide at least one address line, a city and a state for CASS processing. The more input information you provide, the better the results.

- **Company** Very large companies sometimes get their own Plus 4, so providing a company can increase accuracy.
- **Address Line 1 & 2** The delivery address lines.

- **Suite** The suite name and number.
- **City or City/St/Zip** The city, or if it is not pre-split, the entire city/state/zip.
- **State** The 2 letter state abbreviation.
- **Zip** The 5-digit zip code.
- **Plus 4** The zip plus 4 extension.
- **Urbanization** A description of an area, sector, or development within a geographic area. It is commonly used in urban Puerto Rico, as it describes the location of a given street.
- **Carrier Route** Delivery carrier route.
- **Delivery Point Code** A two digit string which describe the 10th and 11th positions of a 12-digit POSTNet barcode.
- **Delivery Point Check Digit** A single digit which describes the 12th position of a 12-digit POSTNet barcode.

Output Specify the fields that will receive the updated address information. You can elect to use the input fields or new fields:

- **Company**
- **Address Line 1 & 2**
- **Suite**
- **City** If you desire the output city, state and zip code to all be in the same field, specify the same output field for **City, State, Zip** and **Plus 4**.
- **State**
- **Zip**
- **Plus 4** If the **Zip** and **Plus 4** are the same field, the field will contain both codes.
- **Urbanization**
- **Carrier Route**
- **Delivery Point Code**
- **Delivery Point Check Digit** If this field is the same as the **Delivery Point Code**, it will be appended to that value.

Carrier Route:

The carrier route is a letter followed by three numbers (for example "R009" and "C039"). The alphabetic character describes the type of delivery:

Code	Description
B	PO Box
C	City Delivery
G	General Delivery
H	Highway Contract
R	Rural Route

4.5.2 Additional Output Fields

MatchUp will populate any of these additional fields from the CASS database:

Demographics:		Size	Split Components:		Size
City Abbreviation:	CITY_ABBR	13	Number:	NUMBER	10
County:	COUNTY	30	Pre-Directional:	PREDIR	10
County FIPS:	FIPS	5	Name:	NAME	20
Time Zone:	TIMEZONE	20	Suffix:	SUFFIX	10
Time Zone Code:	TZC	2	Post-Directional:	POSTDIR	10
MSA Number:	MSA	4	Secondary:	SECONDARY	30
PMSA Number:	PMSA	4	PO Box:	POBOX	30
Congressional Dist:	DISTRICT	2	Address Status:		Size
Geographics:		Size	LACS:	LACS	1
Latitude:	LATITUDE	7	EWS:	EWS	1
Longitude:	LONGITUDE	9	Address Type:	ADDRTYPE	1
			DPV Footnotes:	FOOTNOTES	10

Navigation buttons: Previous, Next, OK, Cancel, Help

Demographics Geopolitical information:

- **City Abbreviation** City names longer than 13 letters will have the Postal Services' official abbreviation. If the city is 13 letters or less, the full city name is returned. For example, "Fort Lauderdale" for "Ft Lauderdale" and "Fredericksburg" for "Fredericksbrg".
- **County** The name of the County the verified address resides in.
- **County FIPS** Federal Information Processing Standard. A 5 digit code defined by the U.S. Census Bureau. The first two digits represent the state code, the last three representing the residing county.
- **Time Zone** A description of the zip code's time zone. example: "Eastern Time"

- **Time Zone Code A** 1 or 2 digit code representing the verified addresses' time zone (see below).

- **MSA Number** Metropolitan Statistical Area Number. A 4 digit code for a metropolitan area made up of one or more counties and meeting statistical criteria defined by the U.S. Office of Management and Budget (OMB). example: "1922"
- **PMSA Number** Primary Metropolitan Statistical Area. A 4 digit code for metropolitan regions with a population greater than one million.
- **Congressional Dist** A 2 digit number representing the congressional district associated with the verified address.

Geographics Note that latitude and longitude are accurate only to the 5-digit zip level, and not the plus 4.

- **Latitude** The 7-digit latitude coordinate (accurate to 4 decimal places) of the center of a Zip code.
- **Longitude** The 9-digit longitude of the center of a Zip code.

Split Components See Matchcode Components for more information on the components of a street address. The Split Components fields are populated regardless of whether or not an address has been successfully CASSed. If a record has been CASSed, these components will be populated with the parsed components as they exist in the USPS database. If not, MatchUp's internal street splitter will parse the components instead.

Address Status

- **LACS** Locatable Address Conversion Service. The total number of records which were flagged as having undergone a change to a city-style address (to allow emergency services to locate these addresses more efficiently). The address change is not reflected here.
- **EWS** Early Warning System Data. Count of new addresses scheduled to be included in the next release of the bimonthly USPS national database. New construction projects for example, are flagged by the EWS file.
- **Address Type** A single character code indicating the type of verified address.
- **DPV Footnotes** A set of one or more 2 character codes containing delivery point information.

Time Zone Code:

Code	Zone
(blank)	Military
4	Atlantic Time
5	Eastern Time
6	Central Time
7	Mountain Time
8	Pacific Time
9	Alaska time
10	Hawaii Time
11	Samoa Time
13	Marshall Island Time
14	Guam Time
15	Palau Time

Address Type:

Code	Description
(blank)	unverified
F	Company Address
G	General Delivery
H	High-Rise or business complex
P	PO Box
R	Rural Route
S	Residential Address

DPV Footnotes:

Code	Description
AA	Input address matched to the Zip+4 file
A1	Input address not matched to the Zip+4 file
BB	Input address matched to DPV (all components)
CC	Input address primary number (street number) matched to DPV but secondary number not matched
N1	Input address primary number matched to DPV but secondary number missing
M1	Input address primary number missing
M3	Input address primary number invalid
P1	Input address PO, RR or HC box number missing
P3	Input address PO, RR or HC box number invalid
RR	Input address matched to CMRA
R1	Input address matched to CMRA but secondary number not present
F1	Address was coded to a military address
G1	Address was coded to a General Delivery address
U1	Address was coded to a unique Zip Code

4.5.3 Options

The screenshot shows the 'CASS Setup' dialog box with the 'Options' tab selected. The 'Output Error Code' is set to 'ERROR' and the 'Output Status Code' is set to 'STATUS'. Under 'If there's a coding error...', the 'Clear the Input Carrier Route field' checkbox is checked. Under 'If joining City/State/Zip...', the 'Delimit City & State with a comma.' radio button is selected. The 'Input Filter' field is empty. The 'CASS database date' is 11/15/2006 and the 'CASS database expires' on 02/28/2007. Three checkboxes are checked: 'Perform Delivery Point Validation (DPV)', 'Use CASSMate enhanced processing', and 'Process via Zip Index'. The 'Form 3553 Information' section contains the following fields: List Name (Demo1.Dbf), Processor Name (Peoplesmith Software), Mailing Company (Peoplesmith Software), Address (50 Cole Parkway, Suite 34), City (Scituate), State (MA), and Zip (02066-1337). Navigation buttons at the bottom include Previous, Next, OK, Cancel, and Help.

Output Error Code Select the field you want to use for the error code.

Output Status Code Select the field you want to use for the status code.

If there's a coding error What should MatchUp do if it comes across a coding error (you can choose one or both):

- **Clear the Input Plus 4 field**
- **Clear the Input Carrier Route field**

City/State Delimiter If you elect to put city, state, and zip data into a single field, how would you like them delimited?

- **Delimit City & State with a space**

- **Delimit City & State with a comma**

Zip/Plus 4 Delimiter Same deal, but separating the Zip and Plus 4:

- **Delimit Zip & Plus 4 with a dash**
- **Delimit Zip & Plus 4 with a space**
- **Delimit Zip & Plus 4 with no delimiters**

Input Filter (Advanced) (optional) Enter the input filtering criteria for each source table. MatchUp will allow you to use a different filter for each input table. The filter must be entered in dBASE syntax. Instead of typing in a dBASE expression, or if you're unfamiliar with dBASE syntax, you can use the Expression Builder.

Perform Delivery Point Validation (DPV) Check if you want to verify each address's delivery point during CASS verification.

Use CASSmate enhanced processing Whether or not to use the enhanced power of CASSmate to get more CASS matches. Enhanced CASSmate processing is only used when normal CASSing attempts have yielded no results.

Process via Zip Index CASSmate generally processes faster in Zip order. But, often your table is not in Zip order (and you don't want it to be). With this option checked, MatchUp will create a temporary index that will cause your table to be CASSed in Zip order (but it's physical order will remain untouched). This option is disabled for all table types except dBASE (the speed gain is countered by random access slowdowns).

Form 3553 information This is the general information necessary to fill out the USPS Form 3553 for the CASS certification. Be sure to fill in this information or your Form 3553 will not print out complete.

Error Codes:

Code	Description
(blank)	No Error
M	Multiple Matches More than one record matches the address and there is not enough information available in the input address to break the tie between multiple records. Passing more complete information, such as city names or urbanization names, can help reduce the number of multiple match errors.
N	No Street Data for Zip The Zip Code exists but no streets begin with the same letter in that Zip Code.
R	Range Error The address was found but the street number in the input address was not between the low and high range in the CASS database.
T	Component Error Either the directionals or the suffix field did not match the CASS database, and there was more than one choice for correcting the address. For example, if the given address was "100 Main St" and the only addresses found were "100 E Main St" and "100 Main Ave" the error code "T" would be returned because we do not know whether to add the directional "E" or to change the suffix to "Ave".
U	Unknown Street An exact street name match could not be found and phonetically matching the street name resulted in either no matches or matches to more than one street name.
X	Non-Deliverable Address The physical location exists but there are no homes on this street. One reason might be that there are railroad tracks or a river running alongside this street, as they would prevent construction of homes in this location.
Z	Invalid Zip Code The Zip Code does not exist and could not be determined by the city and state.
C	Canadian Postal Code The Zip Code matches the format characteristics of a Canadian Postal Code.
W	EWS File Record The Zip Code was found in the Early Warning System Data. These are new addresses scheduled to be included in the next USPS national database.

EWS:

Having a EWS file prevents new addresses from being miscoded with information from the (soon-to-be-outdated) USPS database. For example, an address of 44 Legacy Drive might be changed to 44 Legacy Street if Legacy Drive does not exist in the EWS File. However, if Legacy Drive is in the EWS File, the address is left alone (no Plus 4) and coded with an EWS Error. The next CASS database update should contain the new information and will code the address correctly.

The EWS file is updated weekly every Thursday. To get the most current file, you can download it from <ftp://www.melissadata.com/updates/ews.txt>. Place the **ews.txt** file in the same folder as your other CASS Databases (mdAddr.dat, mdAddr.lic, etc.) overwriting the previous week's file.

Status Codes:

Code	Description
D	Demo Mode If processing with the demo CASS database, records not in the state of Nevada will be coded with "D".
E	Expired Database The CASS database has expired.
S	Standardized but Not Coded Standardization means that some conversion was done on the address (for example, changing "Post Office Box" to "PO Box").
V	Street Number Validated to DPV Level This record has been DPV validated. You can check the DPV Footnotes for more information about the level of validation.
X	Address Not Coded Check the Error code and/or DPV Footnotes for more information.
7	Multiple Matches There were multiple matches for the address that were all in the same Zip Code and Carrier Route. The returned Zip Code and Carrier Route will be correct but you will not get any Plus 4 information.
9	Fully Coded The address was fully CASS coded.

5 Processing

5.1 Merge

The screenshot shows a 'Processing' dialog box with a blue title bar. It contains three main sections: 'Processing Statistics', 'Merging Statistics', and 'Progress'. The 'Processing Statistics' section shows a progress bar at 100% and lists: Elapsed Time: 0:00:00.3, Estimated Time Remaining: 0:00:00, Completed at: 4:46:04 pm, Total Records: 1561, Records Processed: 1561, and Records per Hour: 17953993. The 'Merging Statistics' section contains a table with columns for Source, Total, Processed, and Filtered. The 'Progress' section shows two progress bars, both at 100%. At the bottom, there is a 'Ratio: Screen Updates vs. Processing Speed' slider, a 'Pause' button, a 'Close' button with a green checkmark, and a 'Reports...' button with a document icon.

Processing Statistics:

Elapsed Time: **0:00:00.3** Total Records: **1561**
Estimated Time Remaining: **0:00:00** Records Processed: **1561**
Completed at: **4:46:04 pm** Records per Hour: **17953993**

Merging Statistics:

Source	Total	Processed	Filtered
inhouse.dbf	500	488	12
newcontacts.dbf	500	491	9
suppress.dbf	61	61	0
Database	500	500	0
Total:	1561	1540	21

Progress:

Current Table: 100%

Total Progress: 100%

Ratio: Screen Updates vs. Processing Speed

Updates Speed

Pause Close Reports...

Processing Statistics:

Elapsed Time How much time has elapsed during processing.

Estimated Time Remaining Approximately how much longer processing will take.

Estimated Completion Approximately when the job will be done.

Total Records Total number of records in the table being processed.

Records Processed Number of records that have been processed.

Records per Hour Number of records processed per hour (your mileage may vary).

Merging Statistics:

Source Lists each table that was processed.

Total Total number of records in this table.

Processed The total number of records output from all source tables.

Filtered Number of records filtered from this table (Input Filter is specified under Information about).

Ratio: Screen Updates vs. Processing Speed:

Adjust the slider to your preference. When you set for more frequent screen updates, it appears that records are being processed faster, but it's an optical illusion. In reality, fewer records are being processed (because more time is being wasted updating the screen). Test it yourself!

5.2 Merge/Purge

Processing

Processing Statistics:

Elapsed Time: **0:00:32.0** Total Records: **22146**
 Estimated Time Remaining: **0:00:27** Records Processed: **1797**
 Estimated Completion: **10:40:06 am** Records per Hour: **1342849**

Matching Statistics:

Source	Total	Processed	Filtered	Output	Duplicates	Suppressions	Not Interse...
national	13130	13130	0	413	539	0	0
inhouse	7954	7954	0	278	294	0	0
suppress	61	61	0	0	0	0	0
nonprofit	501	501	0	30	5	0	0
household\$	500	500	0	1	35	0	0

Total:

- Output: **722**
- Duplicates: **873**
- Suppressed: **0**
- Not Intersected: **0**

CASS Coded: **13** of **875**
 DPV Coded: **11** of **875**



Progress:

Thread Progress: 1 CASSing Output 150/159

Current Pass: **74.8%**

Total Progress: **87.4%**

Ratio: Screen Updates vs. Processing Speed

Updates *Speed*

Processing Statistics:

- Elapsed Time** How much time has elapsed during processing.
- Estimated Time Remaining** Approximately how much longer processing will take.
- Estimated Completion** Approximately when the job will be done.
- Total Records** Total number of records in the table being processed.
- Records Processed** Number of records that have been processed.
- Records per Hour** Number of records processed per hour (your mileage may vary).

Matching Statistics:

- Source** Lists each table that was processed.
- Total** Total number of records in this table.
- Processed** The number of records processed in this table.
- Filtered** Number of records filtered from this table (Input Filter is specified under Advanced).
- Output** Number of records in this table that were unique or selected as the output record in a group of duplicates.

Duplicates Number of records in this table that were duplicates of other records - in a group of duplicate records, these records were not selected as the output record, i.e., didn't end up in the Output table.

Suppressions Number of records in this table that matched against records in any suppression list. Suppression records themselves don't contribute to this count.

Non-Intersections Number of records in this table that didn't match against any record in any intersection list. Intersection records themselves don't contribute to this count.

CASS Coded Number of records that were CASS certified.

DPV Coded Number of records that were Delivery Point Validated.

Ratio: Screen Updates vs. Processing Speed:

Adjust the slider to your preference. When you set for more frequent screen updates, it appears that records are being processed faster, but it's an optical illusion. In reality, fewer records are being processed (because more time is being wasted updating the screen). Test it yourself!

5.3 Purge

Processing

Processing Statistics:

Elapsed Time: 0:00:04.2	Total Records: 22585
Estimated Time Remaining: 0:00:00	Records Processed: 22585
Completed at: 10:56:26 am	Records per Hour: 19344753

Matching Statistics:

Source	Total	Processed	Filtered	Output	Duplicates	Suppressions	Not Interse...
national	13130	13130	0	5278	7211	641	0
inhouse	7954	7954	0	0	0	0	0
newcontacts	500	500	0	448	52	0	0
nonprofit	501	501	0	378	123	0	0
household\$	500	500	0	2	498	0	0

Total:

■ Output: 6106	
■ Duplicates: 7884	
■ Suppressed: 641	
■ Not Intersected: 0	

Progress:

Thread Progress:

1 <Idle>

Current Pass 100%

Total Progress 100%

Ratio: Screen Updates vs. Processing Speed

Updates



Speed

Reports...

Analyzer...

Pause

Close

Processing Statistics:

Elapsed Time How much time has elapsed during processing.

Estimated Time Remaining Approximately how much longer processing will take.

Estimated Completion Approximately when the job will be done.

Total Records Total number of records in the table being processed.

Records Processed Number of records that have been processed.

Records per Hour Number of records processed per hour (your mileage may vary).

Matching Statistics:

Source Lists each table that was processed.

Total Total number of records in this table.

Processed The number of records processed in this table.

Filtered Number of records filtered from this table (Input Filter is specified under Advanced).

Output Number of records in this table that were unique or selected as the output record in a group of duplicates.

Duplicates Number of records in this table that were duplicates of other records - in a group of duplicate records, these records were not selected as the output record, i.e., didn't end up in the Output table.

Suppressions Number of records in this table that matched against records in any suppression list. Suppression records themselves don't contribute to this count.

Non-Intersections Number of records in this table that didn't match against any record in any intersection list. Intersection records themselves don't contribute to this count.

Ratio: Screen Updates vs. Processing Speed:

Adjust the slider to your preference. When you set for more frequent screen updates, it appears that records are being processed faster, but it's an optical illusion. In reality, fewer records are being processed (because more time is being wasted updating the screen). Test it yourself!

5.4 File Update

Processing

Processing Statistics:
Elapsed Time: **0:00:04.2**
Estimated Time Remaining: **0:00:00**
Completed at: **10:56:26 am**
Total Records: **22585**
Records Processed: **22585**
Records per Hour: **19344753**

Matching Statistics:

Source	Total	Processed	Filtered	Output	Duplicates	Suppressions	Not Interse...
national	13130	13130	0	5278	7211	641	0
inhouse	7954	7954	0	0	0	0	0
newcontacts	500	500	0	448	52	0	0
nonprofit	501	501	0	378	123	0	0
household\$	500	500	0	2	498	0	0

Total:
Output: **6106**
Duplicates: **7884**
Suppressed: **641**
Not Intersected: **0**



Progress:
Thread Progress: 1 <Idle>
Current Pass: 100%
Total Progress: 100%

Ratio: Screen Updates vs. Processing Speed
Updates / *Speed*

Reports... Analyzer...
Pause Close

Processing Statistics:

Elapsed Time How much time has elapsed during processing.

Estimated Time Remaining Approximately how much longer processing will take.

Estimated Completion Approximately when the job will be done.

Total Records Total number of records in the table being processed.

Records Processed Number of records that have been processed.

Records per Hour Number of records processed per hour (your mileage may vary).

Matching Statistics:

Source Lists each table that was processed.

Total Total number of records in this table.

Processed The number of records processed in this table.

Filtered Number of records filtered from this table (Input Filter is specified under Advanced).

Output Number of records in this table that were unique or selected as the output record in a group of duplicates.

Duplicates Number of records in this table that were duplicates of other records - in a group of duplicate records, these records were not selected as the output record, i.e., didn't end up in the Output table.

Suppressions Number of records in this table that matched against records in any suppression list. Suppression records themselves don't contribute to this count.

Non-Intersections Number of records in this table that didn't match against any record in any intersection list. Intersection records themselves don't contribute to this count.

Ratio: Screen Updates vs. Processing Speed:

Adjust the slider to your preference. When you set for more frequent screen updates, it appears that records are being processed faster, but it's an optical illusion. In reality, fewer records are being processed (because more time is being wasted updating the screen). Test it yourself!

5.5 CASS

The screenshot shows a 'Processing' dialog box with a blue title bar. It is divided into several sections: 'Processing Statistics', 'CASS Statistics', 'CASS Errors', 'DPV Footnotes', and 'Progress'. The 'Processing Statistics' section shows elapsed time, estimated time remaining, and completion time, along with total records, records processed, and records per hour. The 'CASS Statistics' section lists various coding and matching metrics. The 'CASS Errors' section lists types of errors and their counts. The 'DPV Footnotes' section lists various coding and missing data metrics. The 'Progress' section shows a progress bar at 28.7% and a slider for the ratio of screen updates to processing speed. At the bottom right, there are buttons for 'Reports...', 'Pause', and 'Cancel'.

Processing Statistics:

Elapsed Time: 0:00:23.0	Total Records: 817
Estimated Time Remaining: 0:00:56	Records Processed: 235
Estimated Completion: 12:42:07 pm	Records per Hour: 36761

CASS Statistics:

5-Digit Coded: 231	
Zip+4 Coded: 219	
CASSMate Match: 3	
CRRT Coded: 219	
DPV Coded: 216	

CASS Errors:

Multiple Matches (M): 3	Unknown Street (U): 11
No Street Data (N): 0	Non-Deliverable (X): 0
Range Error (R): 1	Zip Code Error (Z): 1
Component Mismatch (T): 0	

DPV Footnotes:

Zip+4 Coded (AA): 219	Missing PO/RR/HC (P1): 0
Zip+4 Not Coded (A1): 16	Invalid PO/RR/HC (P3): 0
DPV Coded (BB): 119	CMRA Coded (RR): 0
Invalid Secondary (CC): 36	Missing Secondary (R1): 1
Missing Secondary (N1): 55	Military Coded (F1): 0
Missing Number (M1): 16	Genl Delivery Coded (G1): 0
Invalid Number (M3): 3	Unique Zip Coded (U1): 6

Progress:

Total Progress: 28.7%

Ratio: Screen Updates vs. Processing Speed

Updates Speed

Reports... Pause Cancel

Processing Statistics:

Elapsed Time How much time has elapsed during processing.

Estimated Time Remaining Approximately how much longer processing will take.

Estimated Completion Approximately when the job will be done.

Total Records Total number of records in the table being processed.

Records Processed Number of records that have been processed.

Records per Hour Number of records processed per hour (your mileage may vary).

CASS Statistics:

5-Digit Coded Records having a valid 5-digit Zip Code.

Zip+4 Coded Records coded with a Plus 4 extension.

CASSmate Match Records coded with a Plus 4 extension through the assistance of CASSMATE. This count is included in the **Zip+4 Coded** total.

CRRT Coded Records coded with a Carrier Route.

DPV Coded Records with a Delivery Point verified address.

CASS Errors:

Multiple Matches (M) More than one record matches the address and there is not enough information available in the input address to break the tie between multiple records. Passing more complete information, such as city names or urbanization names, can help reduce the number of multiple match errors.

No Street Data (N) The Zip Code exists but no streets begin with the same letter in that Zip Code.

Range Error (R) The address was found but the street number in the input address was not between the low and high range in the CASS database.

Component Mismatch (T) Either the directionals or the suffix field did not match the CASS database, and there was more than one choice for correcting the address. For example, if the given address was "100 Main St" and the only addresses found were "100 E Main St" and "100 Main Ave" the error code "T" would be returned because we do not know whether to add the directional "E" or to change the suffix to "Ave".

Unknown Street (U) An exact street name match could not be found and phonetically matching the street name resulted in either no matches or matches to more than one street name.

Non-Deliverable (X) The physical location exists but there are no homes on this street. One reason might be that there are railroad tracks or a river running alongside this street, as they would prevent construction of homes in this location.

Zip Code Error (Z) The Zip Code does not exist and could not be determined by the city and state.

DPV Footnotes:

Zip+4 Coded (AA) Input address matched to the Zip+4 file

Zip+4 Not Coded (A1) Input address not matched to the Zip+4 file

DPV Coded (BB) Input address matched to DPV (all components)

Invalid Secondary (CC) Input address primary number (street number) matched to DPV but secondary number not matched

Missing Secondary (N1) Input address primary number matched to DPV but secondary number missing

Missing Number (M1) Input address primary number missing

Invalid Number (M3) Input address primary number invalid

Missing PO/RR/HC (P1) Input address PO, RR or HC box number missing

Invalid PO/RR/HC (P3) Input address PO, RR or HC box number invalid

CMRA Coded (RR) Input address matched to CMRA

Missing Secondary (R1) Input address matched to CMRA but secondary number not present

Military Coded (F1) Address was coded to a military address

Genl Delivery Coded (G1) Address was coded to a General Delivery address

Unique Zip Coded (U1) Address was coded to a unique Zip Code

Ratio: Screen Updates vs. Processing Speed:

Adjust the slider to your preference. When you set for more frequent screen updates, it appears that records are being processed faster, but it's an optical illusion. In reality, fewer records are being processed (because more time is being wasted updating the screen). Test it yourself!

6 Analyze and Report

Analyzing gives you the chance to view and correct errors that occurred during processing. Analyze is not available for Merge or CASS setups.

6.1 Edit Results

Status	Table	Record	Key		DupeGro...	Zip5	Last Name
0	national	10898	44136SEITZ	38300DOW	4969	44136	Seitz
I	inhouse.	88	44139BRANSTETTE	391313	4970	44139	Branstett
0	national	5264	44139BRANSTETTE	391313	4970	44139	Branstett
I	inhouse.	2352	44139BRANSTETTE	4POBOX3913	4971	44139	Branstett
0	national	88	44139BRANSTETTE	2POBOX3913	4972	44139	Branstett
0	national	9470	44139BRANSTETTE	3POBOX3913	4973	44139	Branstett
I	inhouse.	7231	44143LINTON	6300 WILSONMILL	4974	44143	Linton
0	national	2511	44143LINTON	26300WILSONMILL	4975	44143	Linton
0	national	4453	44143LINTON	26300WILSONMILL	4975	44143	Linton
0	national	7687	44143LINTON	36300WILSONMILL	4976	44143	Linton
I	inhouse.	7420	44143PIUNNO	731 BETA	4977	44143	Piunno
I	inhouse.	7844	44143PIUNNO	731 BETA	4977	44143	Piunno
0	national	2700	44143PIUNNO	2731 BETA	4978	44143	Piunno
0	national	3124	44143PIUNNO	2731 BETA	4978	44143	Piunno

Key: 8489 of 21584 Output record Table: national.dbf Record: 496 10:11:33 AM

Status This information is also available in plain English across the bottom of the Analyze screen:

- **U** A unique record.
- **O** An output record that has dupes (i.e. is not unique).
- **D** A duplicate record that was rejected.
- **S** A suppression table record (neither output nor reject). A lowercase s means that this record was used to suppress records. An uppercase S means that this record didn't find any matching record to suppress.
- **X** A Regular, Self Purge, or No-Purge record that was suppressed (rejected).
- **I** An intersection table record (neither output nor reject). A lowercase i means that this record was used to intersect records. An uppercase I means that this record didn't find any matching record to intersect.
- **Y** A Regular, Self Purge, or No-Purge record that had no intersection hits (rejected).

Table The source table for this record.

Record The original record number for this record.

Matchcode Hits The matchcode combination(s) that caused this match. If you have the mouse over this field, a more verbose representation will be displayed in a tooltip.

Key The matchcode for this record.

Dupe Group Every original record encountered by MatchUp gets a unique number. When a duplicate record is found, it is assigned the same number as the original. This number is known as the dupe group.

Colors:

Output, Rejected, Suppressed, and Non-Intersected records come in two background colors. There's no difference between the two: we just use the color variation to "break up" duplicate groups, much like green bar paper. The color coding can be changed in Analyze | Options.

Analyze:

Next Dupe Group Take you to the next group of duplicate records.

Locate Row Search for a text string in the matchcode key or any displayed column(s).

Locate Again Search for the next record matching the previously specified search criteria.

Show/Hide Rows Select the type of records and matches you want to view in the analyzer.

Sort Rows Alter the sequence of the displayed rows.

Mark as Output Moves the highlighted record to the Output file. You can also press "O" on your keyboard to accomplish the same. The source column shows where the record is currently stored. This option is disabled (dimmed) if you specified no Output file or the record is already marked as output.

Mark as Duplicate Moves the highlighted record to the Duplicate file. You can also press "D" on your keyboard to accomplish the same. The source column shows where the record is currently stored. This option is disabled (dimmed) if you specified no Dupe file or the record is already marked as duplicate.

Mark as Suppressed Moves the highlighted record to the Suppression file. You can also press "S" on your keyboard to accomplish the same. The source column shows where the record is currently stored. This option is disabled (dimmed) if you specified no Suppression file or the record is already marked as suppressed.

Print Rows Print the analyzer results.

Vertical Display Display the analyzer results vertically.

Display Output Display the output record vertically.

Column:

Add Column Display a new column in the browser.

Remove Column Remove an existing column from the browser.

Change Column Change what data is displayed in a column.

Auto-Size Column Change the width of a column to fit the longest string.

Auto-Size All Columns Change the width of each column to fit its longest string.

All Columns This Size Change the width of each column to the size of a column.

Reset Columns Reset all column widths to their original sizes. Show/Hide settings are also reset to show all records.

Options:

Settings Modify the analyzer's display settings.

No Toolbar Hide the analyzer's toolbar.

Small Toolbar Display the small toolbar.

Medium Toolbar Display the medium toolbar.

Large Toolbar Display the large toolbar.

Overview Toggle the narrow browser overview band that appears just to the right of the vertical scrollbar.

6.1.1 Locate Row

Locate Row

Search for: Main

Ignore Case

Search:

Matchcode Key

Selected Column(s):

- Status
- Table
- Record
- Key
- DupeGroup
- Zip5
- First Name

Check All

Check None

Search from start Search from here Cancel Help

Search for The word or phrase that you want to locate.

Ignore case Check if you want to locate the text regardless of the case.

Search Select how you would like to search:

- **Matchcode Key** Search the matchcode keys that were generated during processing (ie, the **Key** column).
- **Selected Column(s)** The checked fields will be searched for the specified text.

Search from start Start searching from the first row.

Search from here Start searching from the current record.

Note:

Only rows that are currently being displayed are searched. If you have used Show/Hide to hide some of the rows, you may not find what you're looking for.

6.1.2 Show/Hide Rows

Show/Hide Rows

Records from Regular, Self-Purge, and No-Purge List(s):

Unique Records Suppressed Records
 Records having Duplicates Non-Intersected Records
 Duplicate Records

Records from Suppression List(s): **Records from Intersection List(s):**

None None
 Only Hits Only Hits
 All Records All Records

Matchcode Combinations:

Z5+FN+LN+S#+SN
 Z5+FN+LN+POB

Show matches having ANY of the above checked combinations.
 Show matches having ALL of the above checked combinations.
 Show matches having EXACTLY the above checked combinations.

Records from Sources:

inhouse.dbf
 national2.dbf
 local.dbf

Records from Regular, Self-Purge and No-Purge List(s):

- **Unique Records** Show records with no matches.
- **Records having Duplicates** Show all records that have matched other records.
- **Duplicate Records** Show duplicate records (this doesn't include suppressed or non-intersected records).
- **Suppressed Records** Show records that were matched against a suppression list(s).
- **Non-Intersected Records** Show records that didn't match against any intersection list(s).

Records from Suppression List(s):

- **None** Don't show any records from the suppression list(s).
- **Only Hits** Only show records that were actually used to suppress other records.
- **All Records** Show all records from the suppression list(s).

Records from Intersection List(s):

- **None** Don't show any records from the intersection list(s).
- **Only Hits** Only show records that were actually used to intersect other records.
- **All Records** Show all records from the intersection list(s).

Matchcode Combinations Check the matchcode combination(s) that you want to view. Matches are filtered based on which combinations are checked as well as the selection:

- **Show matches having ANY of the above checked combinations**
- **Show matches having ALL of the above checked combinations**
- **Show matches having EXACTLY the above checked combinations**

Records from Sources Check the source table(s) you want to view.

6.1.3 Sort Rows



This option lets you display the records in the analyzer in your preferred order.

Sort by Select the sorting method:

- **Dupe Group** Order the records in groups of duplicates so that you can easily see and compare them (this is the default).
- **Matchcode Component(s)** Order the records by the contents of the specified component(s). To change this order, click a component and drag it to the desired position. Sorting will be done first on the top component, then the second, etc.

6.1.4 Print Rows

Print Rows

Printed rows are filtered by the criteria specified in **Analyze | Show/Hide Rows**. They're ordered by the criteria specified in **Analyze | Sort Rows**.

Row Range:

Start at Row: 1

Number of Rows to Print: 0
(enter '0' for all)

Separate Dupe Groups:

- No separator
- Solid line
- Blank line

Additional Information:

- Page headings and page numbers
- Field headings
- Colors

Output Options:

Output to Printer:

Printer: \\PS\laserjet4

Font: Arial

Size: 11

Output to File:

File:

Format: dBase III

OK Cancel Help

Start at row Specify the number of the row that you want printing to start at.

Number of Rows to Print Specify the number of rows to be printed. If you enter "0", all rows from the starting row will be printed.

Separate Dupe Groups Select what should visually separate one group of matching records from the next:

- **No separator**
- **Solid line**
- **Blank line**

Additional Information A few more options:

- **Page headings and page numbers**
- **Field headings**
- **Colors**

Output Options Select the method of output:

- **Output to Printer** Output will be sent to the selected printer. You can select the **Font** that will be used for printing. Additionally, you can click **Properties** to change the printer's properties and **Page Setup** to change the page size and orientation.
- **Output to File** Output will be sent to the specified file. You can select the **Format** of the file (dBASE III, Excel, ASCII SDF, ASCII Comma Delimited and ASCII Tab Delimited).

Note:

Excel formatted files, while eye-appealing, can take a long time to print. If you're using this as a proof of delivery to your customer, you may consider only providing the first 100 or so rows.

6.1.5 Vertical Display

The screenshot shows a window titled "Vertical Display" with a blue title bar. The window content is as follows:

Key 7 of 22146 Suppress
Table national.dbf Record 3621

Status	X
Table	national.dbf
Record	3621
Key	DENNIS ROY ATT
DupeGroup	6
Zip5	
First Name	Dennis
Last Name	Roy
Address Line 1	25th Floor, AT&T Plaza
Address Line 2	

At the bottom of the window, there are four buttons: "Previous" (with left arrows), "Next" (with right arrows), "Close" (with a green checkmark), and "Help" (with a question mark).

Displays a record vertically.

Note:

You can't edit records here.

Keyboard Shortcuts:

PAGEUP, PAGEDOWN Move to the previous/next row.

CTRL+HOME, CTRL+END Move to the first/last row.

6.1.6 Output Display

The screenshot shows a window titled "Output Display" with a blue title bar. Below the title bar, it says "Table output.Db Record 16" and "Output". The main area contains a list of fields with their values:

FIRST	Randy
LAST	Collins
STREET	4545 East 114, Suite 1000
CITY	Irving
STATE	TX
ZIP	
LASTCALL	/ /
AGE	0
SALARY	0
STATS	B1
OUTPUT	RANDY COLLINS 4545 EAST
SC	INHOUSE
COUNT	2
MULTIB	B B

At the bottom of the window, there are four buttons: "Previous" (with left arrows), "Next" (with right arrows), "Close" (with a green checkmark), and "Help" (with a question mark).

Displays an output, duplicate, suppression or non-intersection record vertically. This displays the output record that was created during processing. This is an excellent way to ensure that your gather and scatter settings are correct.

Note:

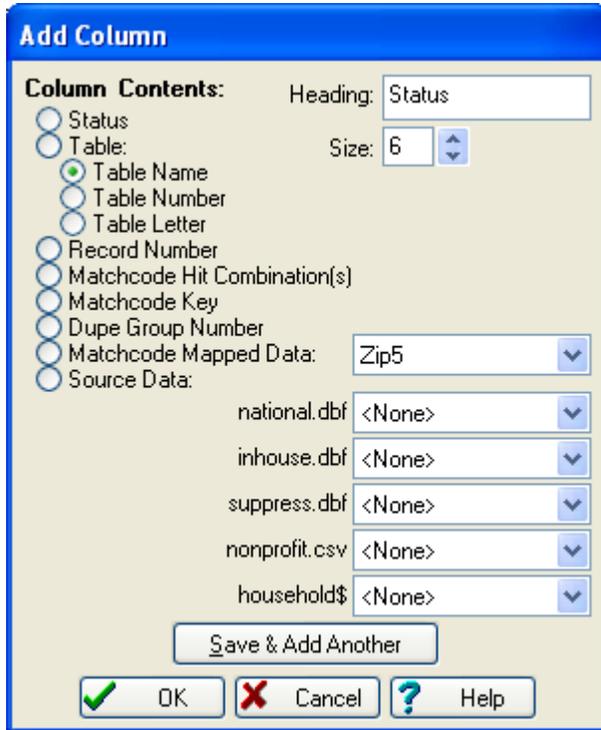
You can't edit records here.

Keyboard Shortcuts:

PAGEUP, PAGEDOWN Move to the previous/next row.

CTRL+HOME, CTRL+END Move to the first/last row.

6.1.7 Add Column



Heading/Size The name and size of the column you want to create.

Column Contents What should be displayed in the column:

- **Status** Display the disposition of the record.
- **Table** Display which table this record came from. You can elect to display the full table name, its number or letter.
- **Record Number** Display the source table's record number.
- **Matchcode Hit Combination(s)** Display the matchcode combination(s) which caused this record to match others (see also Matchcode Combinations).
- **Matchcode Key** Display the record's matchcode key.
- **Dupe Group Number** Display the record's dupe group number.
- **Matchcode Mapped Data** Display data as it was mapped into the matchcode. Select the matchcode component that contains the data you want to display. This is data as it is used in the matchcode (i.e., after any necessary conversion has been made).
- **Source Data** Display data directly from the source table(s). Select the field from each input table you want to display.

Status Codes:

- U** A unique record.
- O** An output record that has dupes (i.e. is not unique).
- D** A duplicate record that was rejected.
- S** A suppression table record (neither output nor reject). A lowercase s means that this record was used to suppress records. An uppercase S means that this record didn't find any matching record to suppress.
- X** A Regular, Self Purge, or No-Purge record that was suppressed (rejected).
- I** An intersection table record (neither output nor reject). A lowercase i means that this record was used to intersect records. An uppercase I means that this record didn't find any matching record to intersect.
- Y** A Regular, Self Purge, or No-Purge record that had no intersection hits (rejected).

6.1.8 Change Column

Change Column

Column Contents:

Status

Table: Heading: Table Size: 8

Table Name

Table Number

Table Letter

Record Number

Matchcode Hit Combination(s)

Matchcode Key

Dupe Group Number

Matchcode Mapped Data: Zip5

Source Data:

TARGET.DBF <None>

BUYER_.DBF <None>

<< Previous Next >>

OK Cancel Help

Heading/Size The name and size of the column.

Column Contents What should be displayed in the column:

- **Status** Display the disposition of the record.
- **Table** Display which table this record came from. You can elect to display the full table name, its number or letter.
- **Record Number** Display the source table's record number.
- **Matchcode Hit Combination(s)** Display the matchcode combination(s) which caused this record to match others (see also Matchcode Combinations).
- **Matchcode Key** Display the record's matchcode key.
- **Dupe Group Number** Display the record's dupe group number.
- **Matchcode Mapped Data** Display data as it was mapped into the matchcode. Select the matchcode component that contains the data you want to display. This is data as it is used in the matchcode (i.e., after any necessary conversion has been made).
- **Source Data** Display data directly from the source table(s). Select the field from each input table you want to display.

6.1.9 Settings

Analyzer Settings

Display Font: Size:

Printer Font: Size:

	Foreground	Background	Sample
Suppression List Records:	<input type="text" value="Black"/>	<input type="text" value="Other"/>	<input type="text" value="Sample"/>
Suppression List Records with Hits:	<input type="text" value="Yellow"/>	<input type="text" value="Red"/>	<input type="text" value="Sample"/>
Suppression List Records with Hits (alt color):	<input type="text" value="Yellow"/>	<input type="text" value="Other"/>	<input type="text" value="Sample"/>
Intersection List Records:	<input type="text" value="Black"/>	<input type="text" value="Other"/>	<input type="text" value="Sample"/>
Intersection List Records with Hits:	<input type="text" value="Yellow"/>	<input type="text" value="Bright Green"/>	<input type="text" value="Sample"/>
Intersection List Records with Hits (alt color):	<input type="text" value="Yellow"/>	<input type="text" value="Other"/>	<input type="text" value="Sample"/>
Unique Records:	<input type="text" value="Black"/>	<input type="text" value="White"/>	<input type="text" value="Sample"/>
Records having Duplicates:	<input type="text" value="Yellow"/>	<input type="text" value="Other"/>	<input type="text" value="Sample"/>
Records having Duplicates (alt color):	<input type="text" value="Yellow"/>	<input type="text" value="Other"/>	<input type="text" value="Sample"/>
Duplicate Records:	<input type="text" value="White"/>	<input type="text" value="Other"/>	<input type="text" value="Sample"/>
Duplicate Records (alt color):	<input type="text" value="White"/>	<input type="text" value="Other"/>	<input type="text" value="Sample"/>
Suppressed Records:	<input type="text" value="White"/>	<input type="text" value="Red"/>	<input type="text" value="Sample"/>
Suppressed Records (alt color):	<input type="text" value="White"/>	<input type="text" value="Other"/>	<input type="text" value="Sample"/>
Non-Intersected Records:	<input type="text" value="White"/>	<input type="text" value="Other"/>	<input type="text" value="Sample"/>

You can change both the display and printer fonts for the Analyzer at the top of this screen.

The rest of the screen is devoted to the color schemes you want to use on the Analyze screen. Many types of records have an alternate color that you can select. There is absolutely no difference between the original and the alternate. The only reason they exist is to provide alternating colors as a visual aid.

6.2 View Reports

Depending on your processing options, MatchUp produces up to 18 different reports. The reports can be viewed on screen through **Analyze | Reports**, as well as printed in a variety of ways.

Reports:

- Processing Summary

- File Summary

- Multiples

- Matrices:

 - Dupe Matrix

 - Multi-Buyer Dupe Matrix

 - Quality Matrix

 - Multi-Buyer Quality Matrix

- Matchcode:

 - Matchcode Matrix

 - Matchcode Quality

- Source Codes:

 - File Summary

 - Multiples

 - Matrices:

 - Dupe Matrix

 - Multi-Buyer Dupe Matrix

 - Quality Matrix

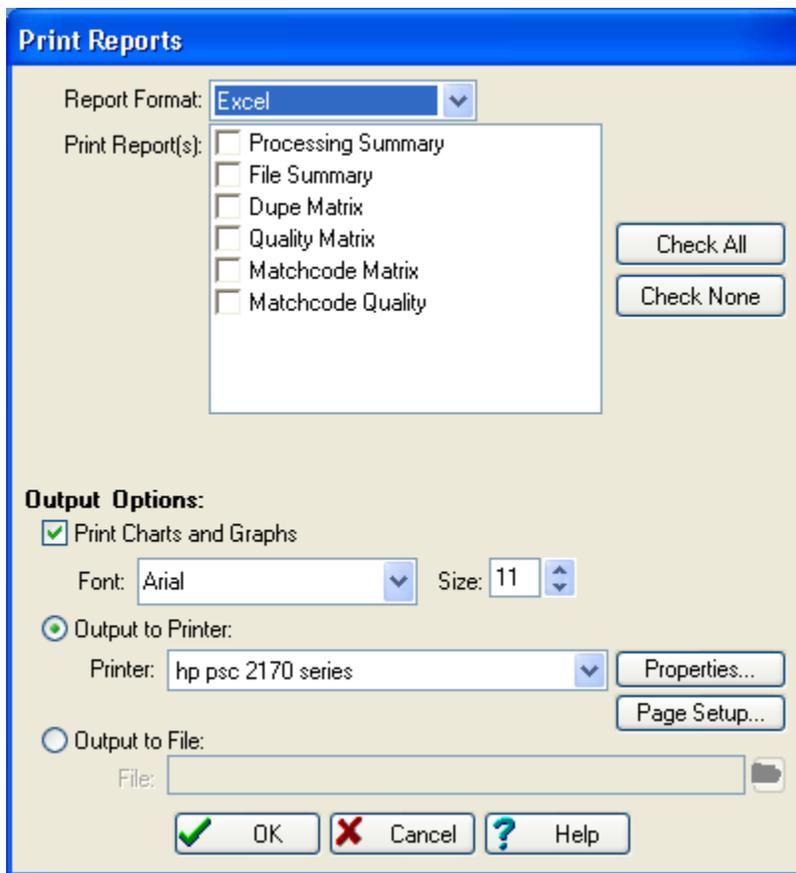
 - Multi-Buyer Quality Matrix

- Merge Matrix

- CASS

Printing:

When you click the **Print** button, the following dialog is displayed. It doesn't matter which report you are currently viewing, the Report list is always the same (i.e., you can print one report while viewing another).



The image shows a 'Print Reports' dialog box with a blue title bar. It contains the following elements:

- Report Format:** A dropdown menu currently set to 'Excel'.
- Print Report(s):** A list of reports with checkboxes:
 - Processing Summary
 - File Summary
 - Dupe Matrix
 - Quality Matrix
 - Matchcode Matrix
 - Matchcode Quality
- Check All** and **Check None** buttons.
- Output Options:**
 - Print Charts and Graphs
 - Font:** 'Arial' (dropdown) and **Size:** '11' (spinners).
 - Output to Printer:**
 - Printer:** 'hp psc 2170 series' (dropdown)
 - Properties...** button
 - Page Setup...** button
 - Output to File:**
 - File:** (text field with folder icon)
- OK** (with green checkmark), **Cancel** (with red X), and **Help** (with question mark) buttons at the bottom.

Report Format Select the type of report you want to create: Excel, Word or Template (see below).

Print Report(s) Check the report(s) that you want to create. The contents of this list will depend on the selected **Report Format** as well as the type of processing (CASS, Merge/Purge, etc).

Print Charts and Graphs Whether or not you want to waste your ink on the pretty charts and graphs.

Font/Size The font to be used for printing. Although Excel and Word formatted reports can be printed with any font, Template formatted reports should be printed with a mono-spaced (a.k.a. fixed-pitch) font such as Courier New.

Output to Printer Output will be sent to the selected printer. You can select the **Font** that will be used for printing. Additionally, you can click **Properties** to change the printer's properties and **Page Setup** to change the page size and orientation.

Output to File Output will be sent to the specified file. For Template formatted reports, this means no charts and graphs.

Report Formats:

- **Excel** The Excel format will produce a report generated through Microsoft Excel. If you send this report to a file, you will find that this produces a single spreadsheet, with tabbed sheets (one for each report). Excel 2000 or later must be installed on your computer.
- **Word** The Word format will produce a report generated through Microsoft Word. If you send this report to a file, you will find that this produces a single document, each report starting on a new page. The page orientation (portrait or landscape) may change from page to page. Word 2000 or later must be installed on your computer.
- **Template** The Template format will produce a report generated through MatchUp. This format is nearly the same as the one used in DoubleTake 2 (in fact, DoubleTake 2 report templates will still work with no modification in most cases). If your computer doesn't have Word or Excel installed, this is the way to go.

6.2.1 Processing Summary

Processing Summary
 Setup File: C:\DT\Projects\Merge Purge.Dt
 Date Processed: 08/11/2005 at 10:18:24 AM
 Elapsed Time: 0:07:49
 Total Records Processed: 22145 Records per Hour: 169716
 Maximum Records per Cluster: 264 Clusters Processed: 286
 Minimum Records per Cluster: 1 Clustering Key Size: 25
 Average Records per Cluster: 77

Output Tables:
 Output: C:\DT\Projects\ Table: output.Dbf
 Duplicate: C:\DT\Projects\ Table: dupe.dbf
 Suppressed: <None>
 Non-Intersected: <None>

Matchcode: Address, Last Name, First Name

Data Type	Size	Start	Fuzzy	Short/Empty	Swap	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Zip5	5	Left	Exact	Both Fields	None	X	X														
First Name	10	Left	Exact	Both Fields	None	X	X														
Last Name	10	Left	Exact	Both Fields	None	X	X														
Street Number	5	Left	Exact	Both Fields	None	X															

The **Processing Summary** contains processing statistics and information about your setup. Note that not all statistics are shown for all processes.

Setup File The name of the setup file used for processing.

Processing Statistics:

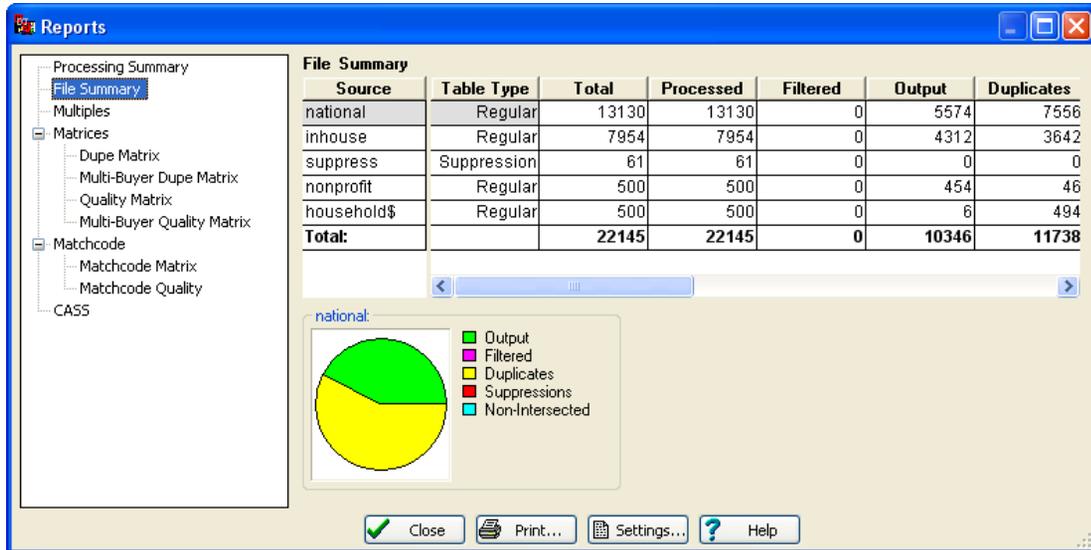
- **Date Processed** The latest processing date.
- **Elapsed Time** How long processing took.
- **Total Records Processed** The total number of records run through the process.
- **Records per Hour** The number of records processed in an hour.
- **Maximum Records per Cluster** Largest cluster.
- **Minimum Records per Cluster** Smallest cluster.
- **Average Records per Cluster** The average number of records in each cluster.
- **Clusters Processed** How many clusters were used in the process (see Concepts for information on clustering).
- **Clustering Key Size** The maximum number of common characters in each cluster load.

Output Tables:

- **Output**
- **Duplicate**
- **Suppressed**

- **Non-Intersected**

6.2.2 File Summary



The **File Summary** contains a detailed listing of what happened with each table. Counts for Output, Duplicates, Suppressions, etc. are given.

Source Lists each table that was processed.

Table Type Indicates the type of table processed: Regular, Suppression, Intersection, Self Purge, or No Purge.

Total Total number of records in this table.

Processed Number of records processed from this table.

Filtered Number of records filtered from this table.

Output Number of records in this table that were unique or selected as the output record in a group of duplicates.

Duplicates Number of records in this table that were duplicates of other records, i.e., didn't end up in the Output table.

Suppressions Number of records in this table that matched against records in a suppression list. Suppression records themselves don't contribute to this count.

Not Intersected Number of records in this table that didn't match against any record in any intersection list. Intersection records themselves don't contribute to this count.

Inter-File Number of dupe records from other tables that matched records output from this table.

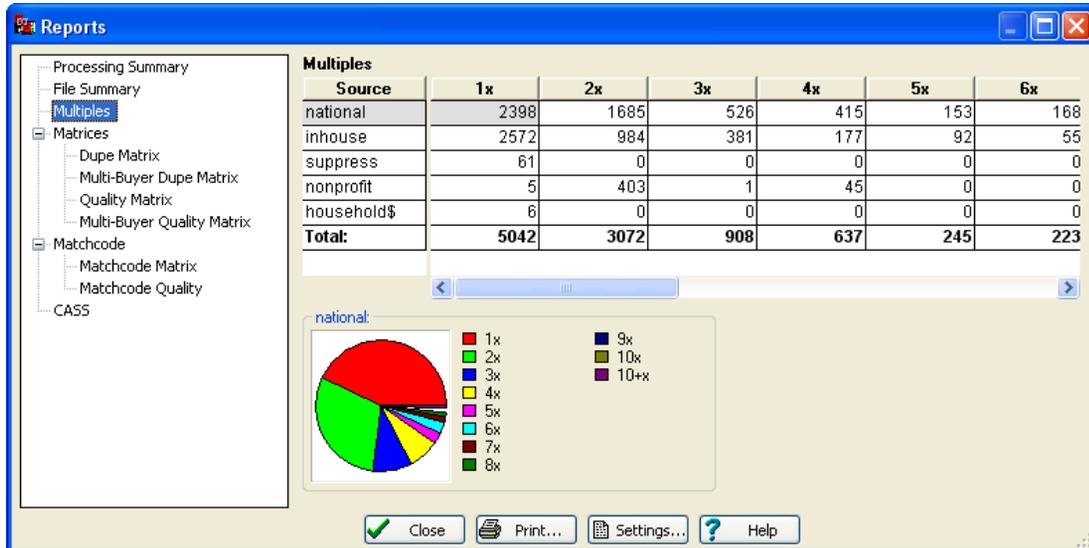
Intra-File Number of dupe records from this table that matched records output from this table.

Multi-Buyer Number of records output from this table that matched against records from another table. See Report Settings on the different ways that this statistic can be calculated.

How the Numbers Add Up:

- $\text{Output} + \text{Not Intersected} + \text{Suppressions} + \text{Duplicates} = \text{Processed}$
- $\text{Processed} + \text{Filtered} = \text{Total}$

6.2.3 Multiples



The **Multiples** report gives you counts of multiple dupe counts for each source table.

Source Lists each table that was processed.

1x Number of unique records (being unique, they ended up in the Output table).

2x Number of records that were selected as the output record (and ended up in the Output table) in a group of two duplicates.

3x-10x Number of records that were selected as the output record in a group of three, four, etc. duplicates.

10+x Number of records that were selected as the output record in a group of eleven or more duplicates (we had to stop sometime).

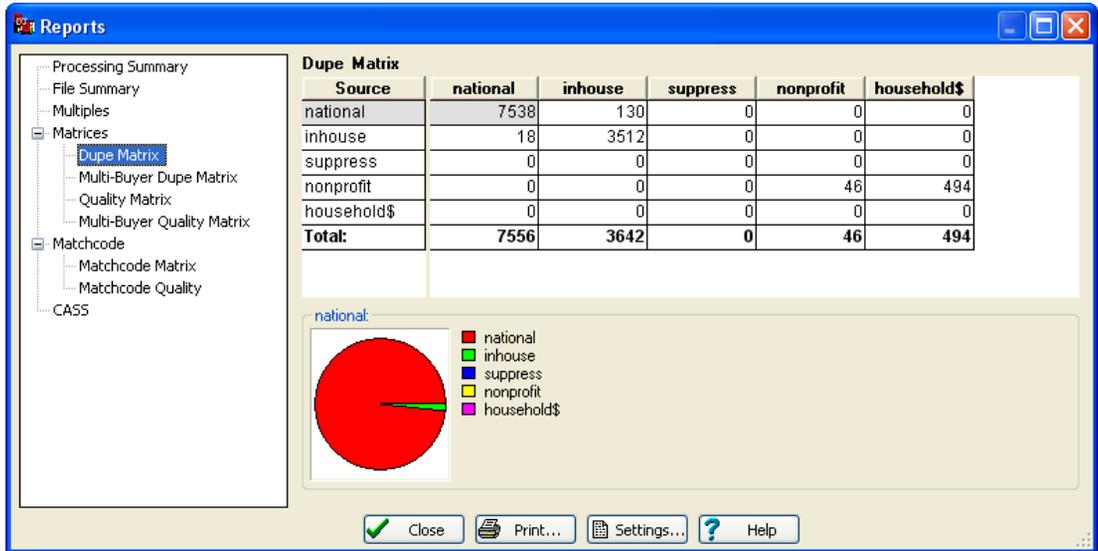
How the Numbers Add Up:

- $1x + 2x + 3x + 4x + 5x + 6x + 7x + 8x + 9x + 10x + 10+x = \text{Output}$
- $2x + (3x*2) + (4x*3) + (5x*4) + (6x*5) + (7x*6) + (8x*7) + (9x*8) + (10x*9) = \text{Total Dupes}$

The last equation only applies for the Total row. Also, if you have any 10+x counts, the formula can't be used, as we have no way of knowing its multiplier.

6.2.4 Matrices

Matrices: Dupe Matrix



The **Dupe Matrix** gives record counts of inter- and intra-file dupes. The pie chart illustrates the highlighted row.

Detailed explanation:

- For all records output from national, there were 7538 dupes of these records in national (intra-file dupes), 130 dupes in inhouse (inter-file dupes), and 0 in any other table.
- For all records output from inhouse, there were 18 dupes of these records in national (inter-file dupes), 3512 dupes in inhouse (intra-file dupes), and 0 in any other table.
- For all chosen records from nonprofit (ending up in the Output File), there were 46 dupes of these records in nonprofit, and 494 in household.

The important thing to remember about this report is that all dupes are counted with respect to which record was output. This is the key difference between this report and the Quality Matrix. The cells where a file intersects itself (the diagonal running from top left to bottom right) are commonly known as the intra-list counts, whereas the others are known as inter-list counts ("intra" meaning "within" and "inter" meaning "between")

Still don't get it? Let's try this example:

Demo1	Demo2	Demo3
Joe	Joe	Joe
Joe	Joe	Marge
Joe	Melissa	
Melissa	Melissa	
Marge		
Marge		

The bold entries indicate which records were selected as output records. A Dupe Matrix for this run would look like:

	Demo1	Demo2	Demo3	
Demo1	2	2	1	2 dupe Joes in 1, 2 in 2, 1 in 3
Demo2	1	1	0	1 dupe Melissa in 1, 1 in 2, 0 in 3
Demo3	2	0	0	2 dupe Marges in 1, 0 in 2, 0 in 3

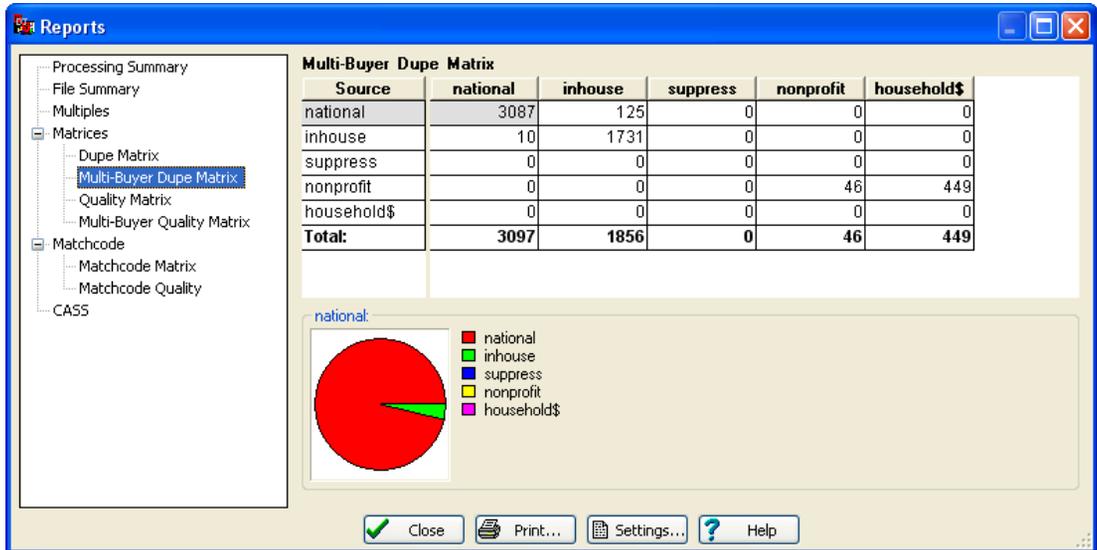
Notice that the output records themselves are not counted, just duplicates. That's why we called it the *Dupe Matrix*.

How the Numbers Add Up:

Each column of numbers add up to the Dupe count for that column's table in the File Summary.

One thing that is not immediately obvious is that a chart's counts are often "crowded" towards a corner. This is very typical if Demo1 is ranked highest in priority, Demo3 lowest. Why? Consider the case when there's a pair of duplicates, one from Demo1, one from Demo3. Because of the ranking, Demo1 gets output, Demo3 duped. So the Demo3 row will never have any counts in Demo1, because there will never be a situation where Demo3 is output and Demo1 is duped.

Matrices: Multi-Buyer Dupe Matrix



The **Multi-Buyer Dupe Matrix** gives record counts of inter- and intra-file dupes. The difference between this report and the Dupe Matrix is that in this report no more than one record is counted per source. The pie chart illustrates the highlighted row.

Detailed explanation:

- For all records output from national, there were 3087 records with at least one matching record from national. There were 125 records output from national that had at least one additional matching record from inhouse.
- For all records output from inhouse, there were 10 records with at least one matching record from national. There were 1731 records output from inhouse that had at least one additional matching record from inhouse.
- For all chosen records from nonprofit (ending up in the Output File), 46 of these records had one or more additional record also in nonprofit, and 449 in household.

The important thing to remember about this report is that all dupes are counted with respect to which record was output and that duplicates coming from the same source table are only tallied once per dupe group.

Still don't get it? Let's try this example:

Demo1	Demo2	Demo3
Joe	Joe	Joe
Joe	Joe	Marge
Joe	Melissa	
Melissa	Melissa	
Marge		
Marge		

The bold entries indicate which records were selected as output records. The dupe groups would look like:

Group 1	Group 2	Group 3
Joe (Demo1)	Melissa (Demo2)	Marge (Demo3)
Joe (Demo1)	Melissa (Demo1)	Marge (Demo1)
Joe (Demo1)	Melissa (Demo2)	Marge (Demo1)
Joe (Demo2)		
Joe (Demo2)		
Joe (Demo3)		

A Multi-Buyer Dupe Matrix for this run would look like:

	Demo1	Demo2	Demo3	
Demo1	1	1	1	Group 1 has matches in Demo1, Demo2 and Demo3
Demo2	1	1	0	Group 2 has matches in Demo1 and Demo2
Demo3	1	0	0	Group 3 has matches in Demo1

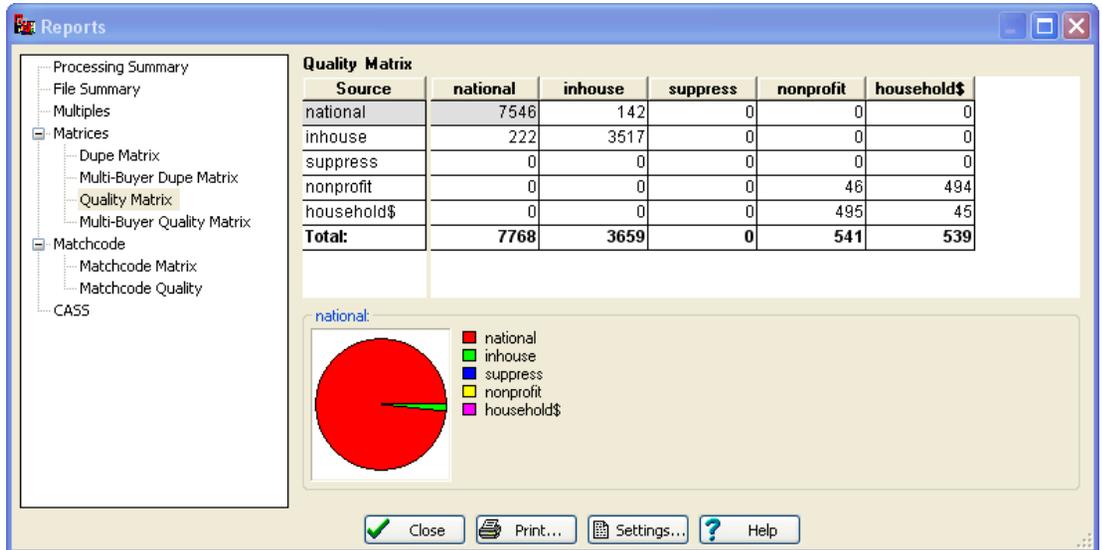
How the Numbers Add Up:

There's no correlation between these numbers and any other report's numbers. However, you can be sure that these numbers will be less than or equal to the numbers in the Dupe Matrix.

What's so Great About this Report?

The Multi-Buyer Dupe Matrix gives you an unbiased indication of how two lists intersect. A high number in a given cell would indicate that the two lists have a number of common records. Although the Dupe Matrix (non-Multi-Buyer) also reflects such a situation, the Dupe Matrix can be fooled by lists with high internal duplicate counts. For example, say Demo1 versus Demo2 has a Dupe Matrix count of 3000 (possibly indicating lots of common contacts between these two lists). However, the Multi-Buyer Dupe Matrix count is only 200. This would indicate that the lists are not very similar, and that the Demo2 list has a lot of internal dupes (which skewed the Dupe Matrix count).

Matrices: Quality Matrix



The **Quality Matrix** gives duplicate counts regardless of where an output record came from. The pie chart illustrates the highlighted row.

For example:

	Demo1	Demo2	Demo3
Demo1	39	69	22
Demo2	70	52	73
Demo3	22	72	46

- There are 39 records in Demo1 that match other records in Demo1, 69 records in Demo2 that match records in Demo1, and 22 records in Demo3 that match records in Demo1.
- There are 70 records in Demo1 that match records in Demo2, 52 records in Demo2 that match other records in Demo2, and 73 records in Demo3 that match records in Demo2.
- There are 22 records in Demo1 that match records in Demo3, 72 records in Demo2 that match records in Demo3, and 46 records in Demo3 that match other records in Demo3.

The intra- and inter- terminology *does not* apply to the Quality Matrix, as the counts are not generated with any regard to which record was output. What this report does tell you is the

"cross-saturation" between tables. Often, this is hidden in the Dupe Matrix. For example, using the sample report seen in the Dupe Matrix, a count of 0 is given in the Demo3 versus Demo2 cell. But here, we see a 72 - meaning that there are 72 records in common between the two tables. This count was "hidden" in the Dupe Matrix because the ranking caused all of these matches to accumulate into the Demo1 or Demo2 rows (because Demo1 was prioritized over Demo2 which was prioritized over Demo3)

This is a tough report to understand, so let's try this example (it's the same set of records used in the Dupe Matrix example):

Demo1	Demo2	Demo3
Joe	Joe	Joe
Joe	Joe	Marge
Joe	Melissa	
Melissa	Melissa	
Marge		
Marge		

No indicator of which record was selected as the output record is given for this report for the sake of clarity (you can assume that they are the same as the ones in the Dupe Matrix example). A Quality Matrix for this run would look like:

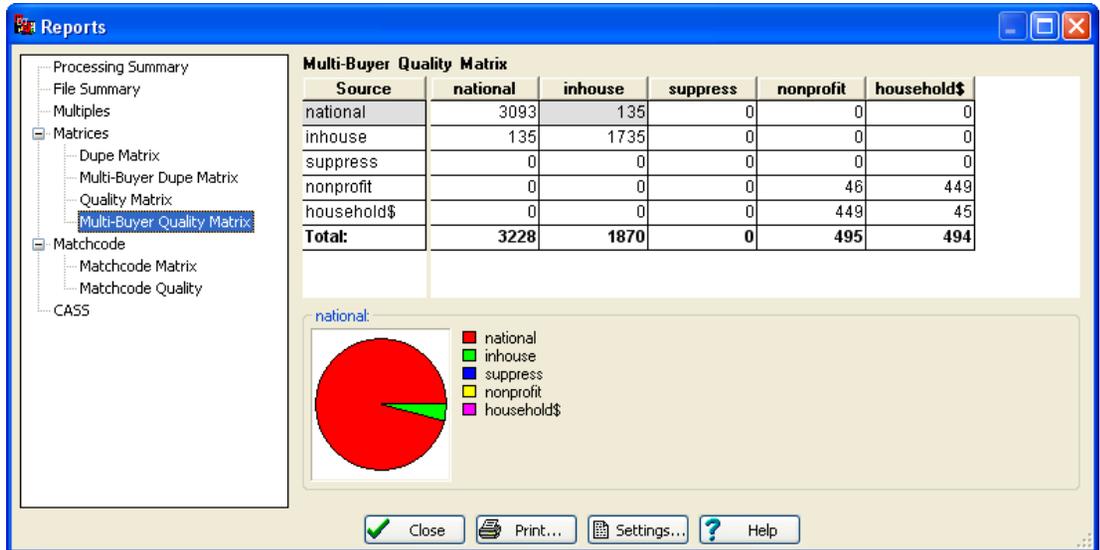
	Demo1	Demo2	Demo3	
Demo1	3	4	2	2 dupe Joes + 1 dupe Marge in 1 2 dupe Joes + 2 dupe Melissa in 2 1 dupe Joe + 1 dupe Marge in 3
Demo2	4	2	1	3 dupe Joes + 1 dupe Melissa in 1 1 dupe Joe + 1 dupe Melissa in 2 1 dupe Joe in 3
Demo3	5	2	0	3 dupe Joes + 2 dupe Marges in 1 2 dupe Joes in 2 No dupes of anything in 3

Basically, you look at each unique record in a table and dedupe it against everything else. The unique record does not get figured into the counts (just duplicates of it). Using Brand X software, you would have to run 3 dedupe runs, assigning different priorities each time: first to Demo1, then to Demo2, and finally to Demo3.

How the Numbers Add Up:

Good luck! Try as you might, there is no correlation between the numbers produced in the Quality Matrix and any other counts. Don't call us and ask us if we've "fixed" this report, or made some sort of scientific advance - the numbers don't add up because of what they represent; there's nothing wrong with your television.

Matrices: Multi-Buyer Quality Matrix



The **Multi-Buyer Quality Matrix** gives duplicate counts regardless of where an output record came from. The difference between this report and the Quality Matrix is that in this report no more than one record is counted per source. The pie chart illustrates the highlighted row.

For example:

	Demo1	Demo2	Demo3
Demo1	39	69	22
Demo2	69	52	72
Demo3	22	72	46

- There are 39 dupe groups with 2 or more records from Demo1. 69 dupe groups with records from Demo1 and Demo2, and 22 dupe groups with records from Demo1 and Demo3.
- There are 69 dupe groups with records from Demo2 and Demo1, 52 dupe groups with 2 or more records from Demo2, and 72 dupe groups with records from Demo2 and Demo3.
- There are 22 dupe groups with records from Demo3 and Demo1, 72 dupe groups with records from Demo3 and Demo2, and 46 dupe groups with 2 or more records from Demo3.

What this report does tell you is the "cross-saturation" between tables. Often, this is hidden in the Dupe Matrix. For example, using the sample report seen in the Dupe Matrix, a count of 0 is given in the Demo3 versus Demo2 cell. But here, we see a 72 - meaning that there are 72 records in common between the two tables. This count was "hidden" in the Dupe Matrix because the ranking caused all of these matches to accumulate into the Demo1 or Demo2 rows (because Demo1 was prioritized over Demo2 which was prioritized over Demo3)

In some ways, this report can be more useful than the Quality Matrix. Because intra-file dupes are not counted, lists with lots of internal dupes do not appear to have more 'real' matches than they actually do.

This is a tough report to understand, so let's try this example (it's the same set of records used in the Dupe Matrix example):

Demo1	Demo2	Demo3
Joe	Joe	Joe
Joe	Joe	Marge
Joe	Melissa	
Melissa	Melissa	
Marge		
Marge		

The dupe groups would look like:

Group 1	Group 2	Group 3
Joe (Demo1)	Melissa (Demo2)	Marge (Demo3)
Joe (Demo1)	Melissa (Demo1)	Marge (Demo1)
Joe (Demo1)	Melissa (Demo2)	Marge (Demo1)
Joe (Demo2)		
Joe (Demo2)		
Joe (Demo3)		

A Multi-Buyer Quality Matrix for this run would look like:

	Demo1	Demo2	Demo3	
Demo1	2	2	2	Groups 1 and 3 have double matches in Demo1 Groups 1 and 2 have records from both Demo1 and Demo2 Groups 1 and 3 have records from both Demo1 and Demo3
Demo2	2	2	1	Groups 1 and 2 have records from both Demo2 and Demo1 Groups 1 and 2 have double matches in Demo2 Group 1 has records from both Demo2 and Demo3
Demo3	2	1	0	Groups 1 and 3 have records from both Demo3 and Demo1 Group 1 has records from both Demo3 and Demo2 No group has double matches in Demo3

How the Numbers Add Up:

They don't; there's no correlation between these numbers and any other report's numbers. However, you can be sure that these numbers will be less than or equal to the numbers in the Quality Matrix.

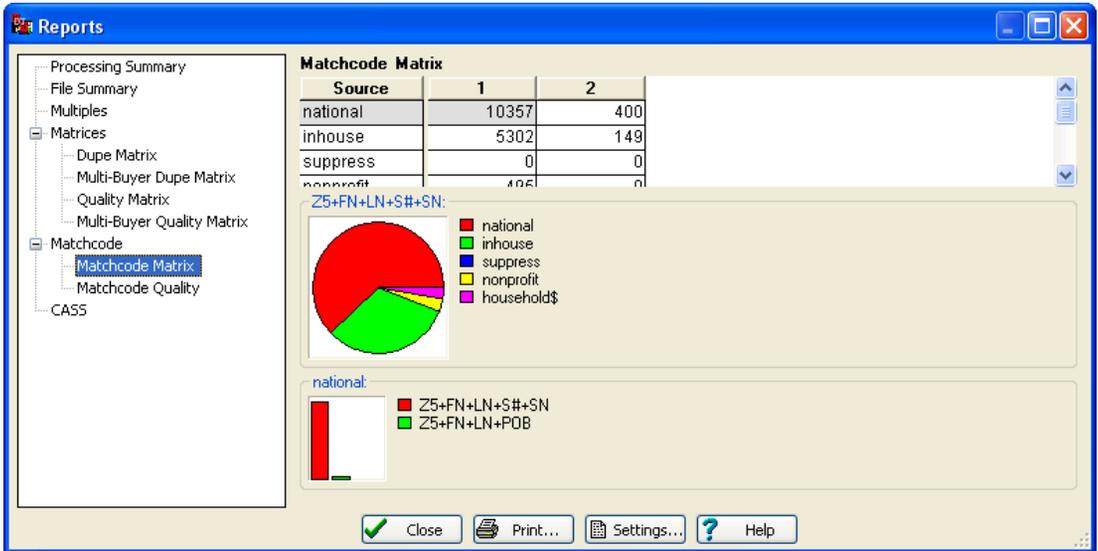
What's so Great About this Report?

The Multi-Buyer Quality Matrix gives you an unbiased indication of how two lists intersect. A high number in a given cell would indicate that the two lists have a number of common records. Although the Quality Matrix (non-Multi-Buyer) also reflects such a situation, the Quality Matrix can be fooled by lists with high internal duplicate counts. For example, say Demo1 versus Demo2 has a Quality Matrix count of 3000 (possibly indicating lots of common contacts between these two lists). However, the Multi-Buyer Quality Matrix count is only 100. This would indicate that the lists are not very similar, and that the Demo2 list has a lot of internal dupes (which skewed the Quality Matrix count).

This report also has an advantage over the Multi-Buyer Dupe Matrix, in that the output record doesn't skew the counts in any way. Because of this, commonality between two lists that is obscured by file ranking can be seen.

6.2.5 Matchcode

Matchcode: Matchcode Matrix



The **Matchcode Matrix** lists the number of matches made based on each matchcode combination. The bar chart illustrates the highlighted row, while the pie chart illustrates the highlighted column.

For example, say combination 1 is Zip+LN+POB and combination 2 is Zip+LN+S#+SN (combinations 3 through 16 weren't used). Using the example below:

	1	2
Demo1	20	174
Demo2	21	225
Demo3	15	175

- There were 20 records in Demo1 that matched other records (in any of the three tables) using the first combination (Zip+LN+POB). There were 174 records in Demo1 that matched other records using the second combination (Zip+LN+S#+SN).
- There were 21 records in Demo2 that matched other records using the first combination (Zip+LN+POB) and 225 records in Demo2 that matched other records using the second combination (Zip+LN+S#+SN).
- There were 15 records in Demo3 that matched other records using the first combination and 175 records in Demo3 that matched other records using the second combination.

Note that the output record is not used in these calculations. Also, if matches are made using both combinations, both columns' counts will reflect that match.

What this report can tell you is how well a matchcode combination is working. If a particular combination is generating low counts, that combination may not be working for you. Don't regard these counts as an exact indicator of performance, though. In our above example, we certainly should expect the PO Box combination to generate substantially less hits than the Street Address combination - after all, a large part of our population likes mail delivered to the door!

Here's an example of how this report is generated (bold indicating the output record). Warning: this is really involved:

Demo1	Demo2	Demo3
123 Main	123 Main	123 Main
POB 10	POB 10	POB 67
POB 67/10 Main	10 Main	POB 67/10 Main

During deduping, we end up with 3 dupe groups:

Group 1:

- 123 Main** from Demo1 (had Street # matches)
- 123 Main from Demo2 (matched because of Street #)
- 123 Main from Demo3 (matched because of Street #)

Group 2:

- POB 10** from Demo2 (matched because of PO Box)
- POB 10 from Demo1 (matched because of PO Box)

Group 3:

- POB 67/10 Main** from Demo3 (had both Street # and PO Box matches)
- POB 67/10 Main from Demo1 (matched because of both Street # and PO Box)
- 10 Main from Demo2 (matched because of Street #)
- POB 67 from Demo3 (matched because of PO Box)

The Matchcode Matrix for this run would look like (assume combination 1 is Zip+LN+POB and combination 2 is Zip+LN+S#+SN):

	1	2	
Demo1	2	2	Group 1: 1 Street # Group 2: 1 POB Group 3: 1 POB and Street #
Demo2	1	2	Group 1: 1 Street # Group 2: 1 POB Group 3: 1 Street #
Demo3	2	2	Group 1: 1 Street # Group 2: Group 3: 2 POBs and 1 Street #

Notice how the "POB 67/10 Main" produced a hit in both columns. Even if MatchUp finds a match due to combination 1, it will still check out combination 2.

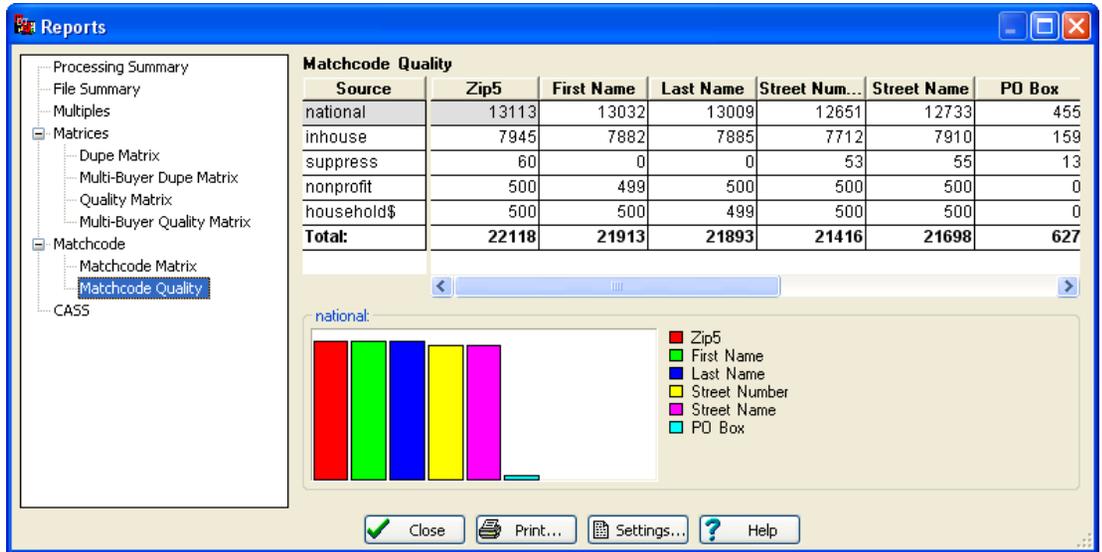
How the Numbers Add Up:

Sometimes they do. The main reason that they usually don't is the case when a record matches other record(s) using more than one combination - a "double hit" is accumulated. But, when this is not the case (and in the Street #/PO Box situation, this is often true), this will work:

- $\text{Total Output} - \text{Total 1x} + \text{Total Dupe} = \text{Sum of Combinations}$

If this equation doesn't apply, it means you have some multiple combination matches. The accumulation of multiple combination matches was intentional. If we wanted these numbers to always add up, we would have had to decide which combination(s) should be left uncounted in multiple combination matches. It was more important to us to generate accurate counts rather than give you that "everything adds up" warm and fuzzy feeling. If that bugs you, feel free to "fudge" the numbers. We won't tell (unless someone asks).

Matchcode: Matchcode Quality



The **Matchcode Quality** report gives you counts of how well your records were populated with the matchcode components. The bar graph illustrates the highlighted row.

This report is refreshingly simple. For each table, records having each matchcode component are counted. These counts can indicate problems with a database (like missing Zip Codes), as well as bad setup mappings (i.e., trying to map a First Name field as a Full Name).

The counts are generated *after* any on-the-fly splitting has been performed, so don't be alarmed because Street # and Street Name are listed even though your database doesn't have these fields - this information was extracted from the specified address lines.

How the Numbers Add Up:

You would love to see each count match the number of records processed for that table. That, of course, is a dream. Bad records are sure to screw things up and generate missing Zip Codes, Street Names, and so on. But you can dream!

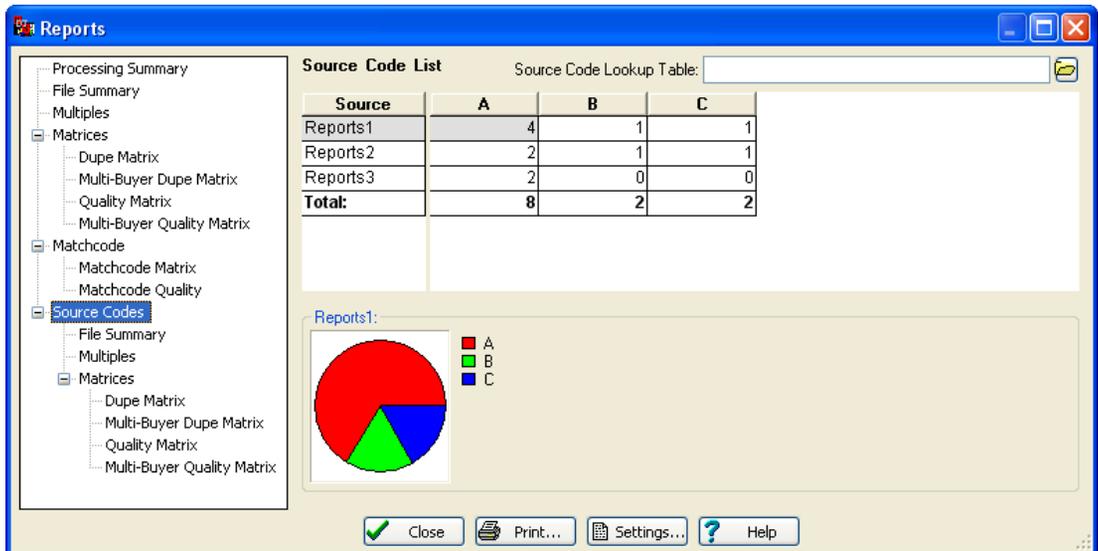
This is a great report to give to your customer when he claims that his data is pristine and had been input by a sect of data entry monks, when in reality it was entered by a roomful of monkeys trying to type *Macbeth*.

6.2.6 Source Codes

If you specified an Input Source Code Field for one or more of your input tables, Source Code reports will be available.

The Input Source Code Field is specified on the **Advanced** tab in Setup | Edit and is specific to each Input Table.

MatchUp uses the contents of your Input Source Code Field to generate report counts. This gives you counts based on a set of keycodes contained in your tables.



This report is generated whenever you have specified at least one Source Code Field (Merge/Purge Setup: Advanced or Purge Setup: Advanced). For each source code detected in the process, a count is accumulated for each table. The pie chart illustrates the highlighted column.

But, you ask, what about the times when I've only specified a Source Code Field for only some of my files? They get their own special source code: "**SF" followed by the number of the file from which the record came. Because of the leading "**", these codes will generally appear at the top of the list.

Source codes are counted during the Purge pass. For this reason, source codes are always based on post-filtered records (as filters are applied during the Key Building pass).

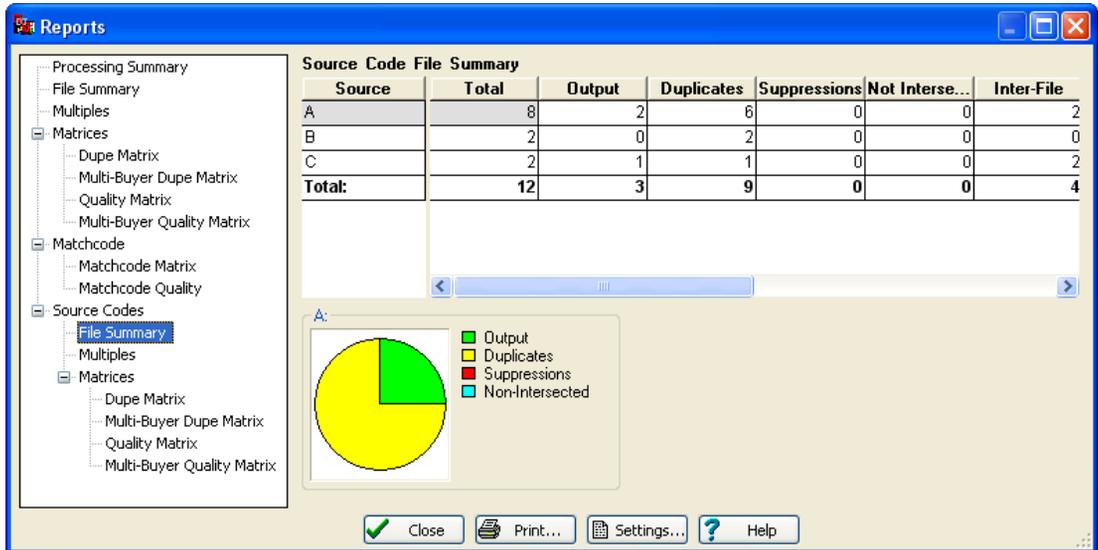
Source Code Lookup Table:

Sometimes source codes aren't very descriptive. You can provide MatchUp with a list of source codes and their descriptions in the "Source Code Lookup Table" prompt. This list must be a dBASE III table with at least two fields. The first field should contain the source codes (as they appear in your data), the second field should contain the source code's longer description. If you forget any source codes, the original (non-descriptive) source code will be used instead.

How the Numbers Add Up:

- Total Source Codes for a File = Records Processed for that File

Source Codes: File Summary



The **Source Code File Summary** shows processing results by source code. It is just like the File Summary with that one difference. Because this report is oriented to source codes, you will only get it when you have specified at least one Source Code Field (Merge/Purge Setup: Advanced or Purge Setup: Advanced)

Source Code Lists each source code that was processed.

Total Total number of records with this source code.

Output Number of records with this source code that were unique or selected as the output record in a group of duplicates.

Duplicates Number of records with this source code that were duplicates.

Suppressions Number of records with this source code that were suppressed. Suppression records themselves don't contribute to this count.

Not Intersected Number of records with this source code that didn't match against any record in any intersection list. Intersection records themselves don't contribute to this count.

Inter-File Number of dupe records from other sources that matched records from this source.

Intra-File Number of dupe records from this source that matched records output from this source.

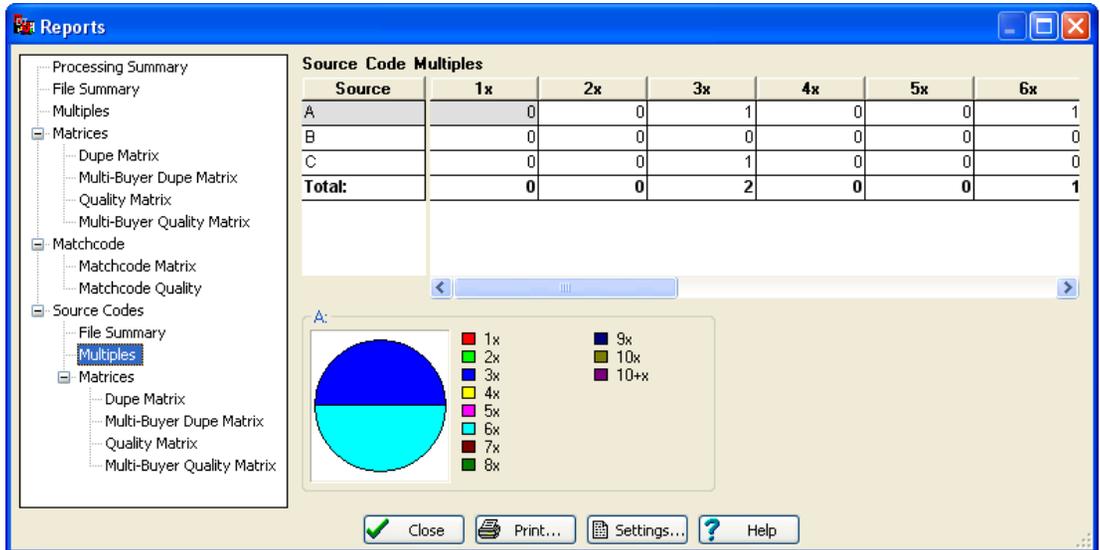
Multi-Buyer Number of records output from this table that matched against records from another table. See Report Settings on the different ways that this statistic can be calculated.

Why no Table Type and Data Type columns? Because source codes aren't table based - they can be scattered across many tables. How about the Processed and Filtered columns? Because source codes are accumulated during the Purge pass and filters are applied during the Key Building pass, all source codes counted have survived any filter conditions.

How the Numbers Add Up:

- Output + No Intersect + Suppress + Dupe = Total

Source Codes: Multiples



The **Multiples** report gives you counts of multiple dupe counts for each source table.

Source Code Lists each source that was processed.

1x Number of unique records (being unique, they ended up in the Output table).

2x Number of records that were selected as the output record (and ended up in the Output table) in a group of two duplicates.

3x-10x Number of records that were selected as the output record in a group of three, four, etc. duplicates.

10+x Number of records that were selected as the output record in a group of eleven or more duplicates (we had to stop sometime).

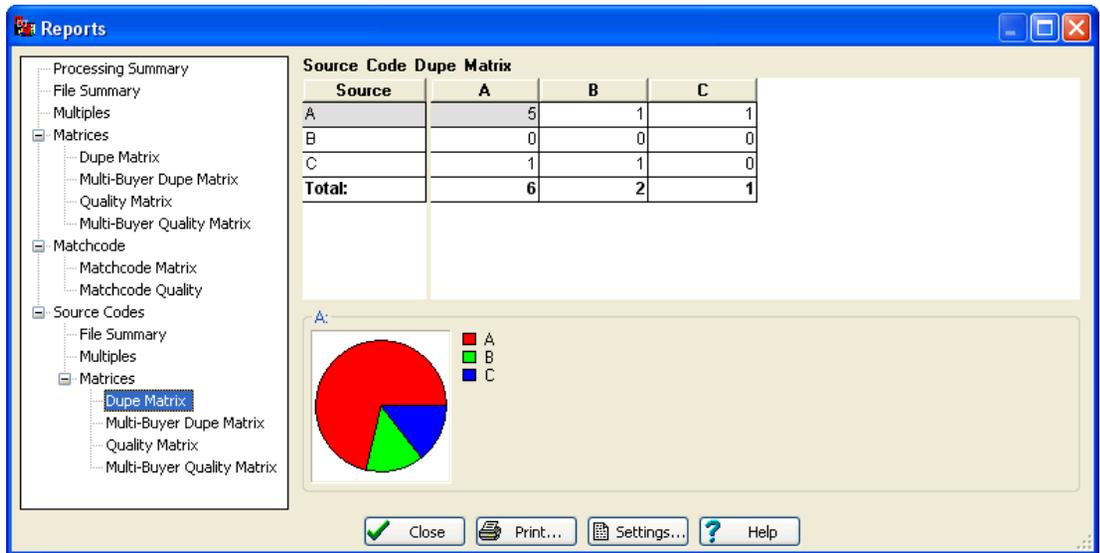
How the Numbers Add Up:

- $1x + 2x + 3x + 4x + 5x + 6x + 7x + 8x + 9x + 10x + 10+x = \text{Output}$
- $2x + (3x \cdot 2) + (4x \cdot 3) + (5x \cdot 4) + (6x \cdot 5) + (7x \cdot 6) + (8x \cdot 7) + (9x \cdot 8) + (10x \cdot 9) = \text{Total Dupes}$

The last equation only applies for the Total row. Also, if you have any 10+x counts, the formula can't be used, as we have no way of knowing its multiplier.

Source Codes: Matrices

Matrices: Dupe Matrix



The **Source Code Dupe Matrix** gives record counts of inter- and intra-source dupes. The pie chart illustrates highlighted row. This report is identical to the Dupe Matrix, except that counts are based on source codes, rather than source tables.

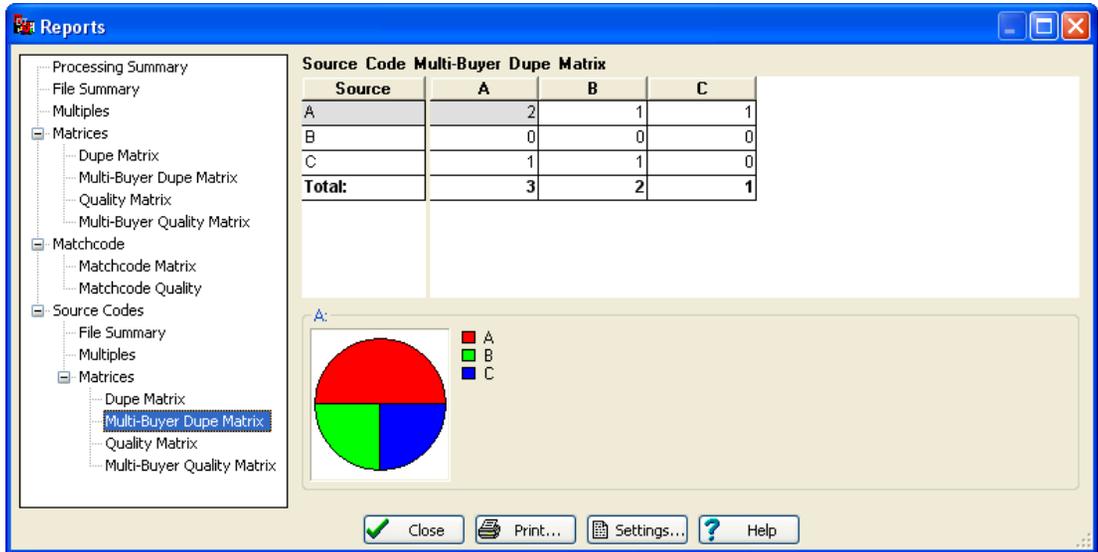
This report is generated whenever you have specified at least one Source Code.

The important thing to remember about this report is that all dupes are counted with respect to which record was output. This is the key difference between this report and the Quality Matrix. The cells where a table intersects itself (the diagonal running from top left to bottom right) are commonly known as the intra-list counts, whereas the others are known as inter-list counts.

How the Numbers Add Up:

Each column of numbers add up to the Dupe count for that column's source code in the Source Code File Summary.

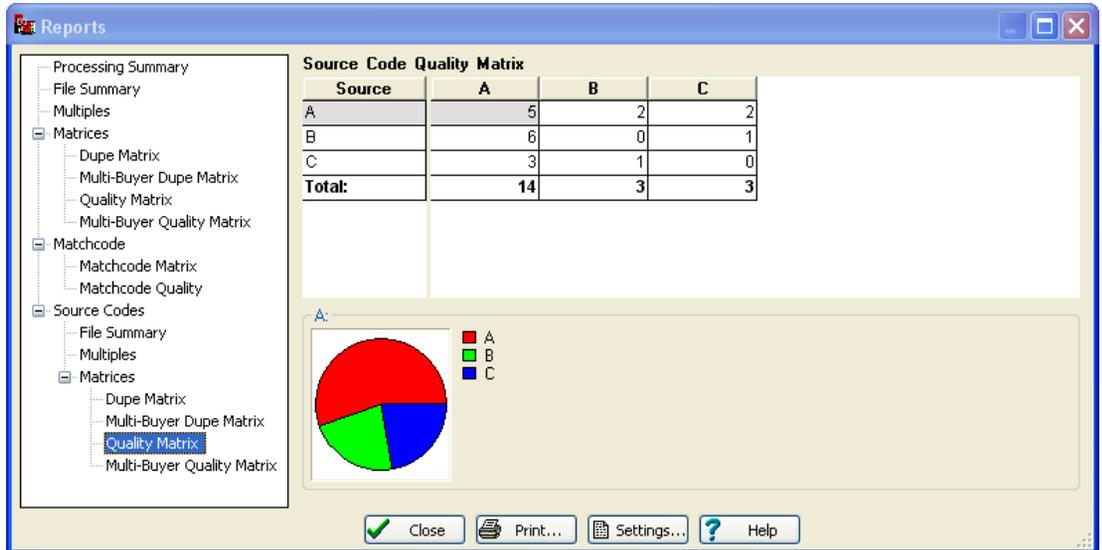
Matrices: Multi-Buyer Dupe Matrix



The **Source Code Multi-Buyer Dupe Matrix** gives record counts of inter- and intra-file dupes. The difference between this report and the Dupe Matrix is that in this report no more than one record is counted per source.

This report is generated whenever you have specified at least one Source Code. See Multi-Buyer Dupe Matrix for a detailed explanation about this report.

Matrices: Quality Matrix



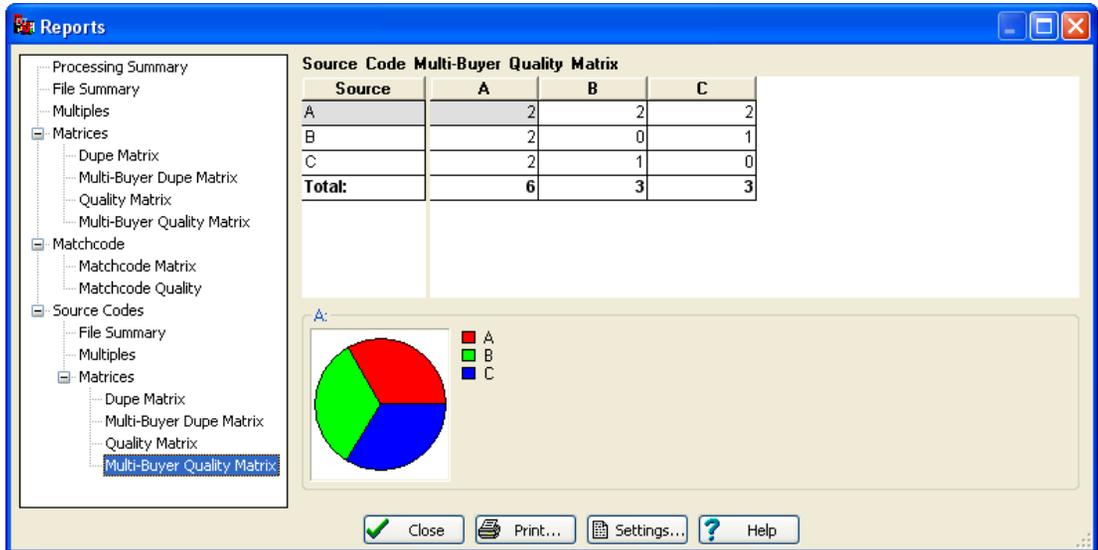
The **Source Code Quality Matrix** gives duplicate counts regardless of where an output record came from. The pie chart illustrates the highlighted row.

This report is identical to the Quality Matrix, except that counts are based on source codes, rather than source tables.

How the Numbers Add Up:

Like the Quality Matrix, the Source Code Quality Matrix does not produce counts that will "jibe" with any other numbers. They're just not supposed to.

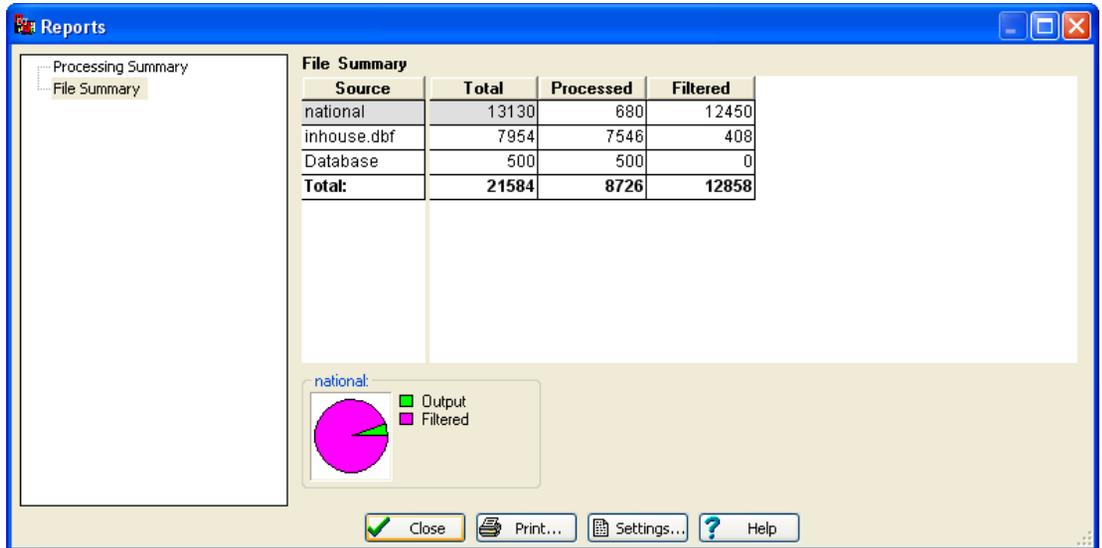
Matrices: Multi-Buyer Quality Matrix



The **Multi-Buyer Quality Matrix** duplicate counts regardless of where an output record came from. The difference between this report and the Quality Matrix is that in this report no more than one record is counted per source.

This report is generated whenever you have specified at least one Source Code. See Multi-Buyer Quality Matrix for a detailed explanation about this report.

6.2.7 Merge Matrix



The **Merge Matrix** contains a detailed listing of what happened with each table during a Merge process.

The pie chart illustrates the highlighted table.

Source Lists each table that was processed.

Total Total number of records in this table.

Processed Number of records processed from this table.

Filtered Number of records filtered from this table.

6.2.8 CASS

The screenshot shows a window titled "Reports" with a blue border. On the left is a small empty box labeled "CASS". The main area displays the following information:

Setup File: C:\Temp\CASS Test.dt
Table: C:\Temp\ Table: CASS Test.dbf
Date Processed: 04/13/2007 at 10:51:37 AM
Elapsed Time: 0:00:56
Records Processed: 1552 of 1552

Counts:

5-Digit Coded:	1390	CASSmate Match:	10
Zip+4 Coded:	1352	DPV:	1337
CRRT Coded:	0	LACS ^{link} Converted:	9

Errors:

Multiple Matches (M):	4	Unknown Street (U):	26
No Street Data (N):	0	Non-Deliverable (X):	0
Range Error (R):	14	Zip Code Error (Z):	43
Component Mismatch (T):	0		

DPV Footnotes:

Zip+4 Coded (AA):	1352	Missing PD/RR/HC (P1):	0
Zip+4 Not Coded (A1):	87	Invalid PD/RR/HC (P3):	0
DPV Coded (BB):	1054	CMRA Coded (RR):	0
Invalid Secondary (CC):	110	Missing Secondary (R1):	15
Missing Secondary (N1):	157	Military Coded (F1):	0
Missing Number (M1):	86	Genl Delivery Coded (G1):	0
Invalid Number (M3):	15	Unique Zip Coded (U1):	16

Statistics:

High Rise Default:	214	High Rise Exact:	314
Rural Route Default:	0	Rural Route Exact:	0
LACS:	9	EWS:	0

CASS database date: 03/15/2007
CASS database expires: 06/30/2007

At the bottom are buttons for Close, Print..., Settings..., and Help.

Processing Statistics:

Setup File The name of the setup file used for processing.

Table The table processed.

Date Processed The latest processing date.

Elapsed Time How long processing took.

Records Processed The total number of records run through the process.

Counts:

5-Digit Coded Records having a valid 5-digit Zip Code.

Zip+4 Coded Records coded with a Plus 4 extension.

CASSmate Match Records coded with a Plus 4 extension through the assistance of CASSMATE. This count is included in the **Zip+4 Coded** total.

CRRT Coded Records coded with a Carrier Route.

DPV Coded Records with a Delivery Point verified address.

Errors:

Multiple Matches (M) More than one record matches the address and there is not enough information available in the input address to break the tie between multiple records. Passing more complete information, such as city names or urbanization names, can help reduce the number of multiple match errors.

No Street Data (N) The Zip Code exists but no streets begin with the same letter in that Zip Code.

Range Error (R) The address was found but the street number in the input address was not between the low and high range in the CASS database.

Component Mismatch (T) Either the directionals or the suffix field did not match the CASS database, and there was more than one choice for correcting the address. For example, if the given address was "100 Main St" and the only addresses found were "100 E Main St" and "100 Main Ave" the error code "T" would be returned because we do not know whether to add the directional "E" or to change the suffix to "Ave".

Unknown Street (U) An exact street name match could not be found and phonetically matching the street name resulted in either no matches or matches to more than one street name.

Non-Deliverable (X) The physical location exists but there are no homes on this street. One reason might be that there are railroad tracks or a river running alongside this street, as they would prevent construction of homes in this location.

Zip Code Error (Z) The Zip Code does not exist and could not be determined by the city and state.

DPV Footnotes:

Zip+4 Coded (AA) Input address matched to the Zip+4 file

Zip+4 Not Coded (A1) Input address not matched to the Zip+4 file

DPV Coded (BB) Input address matched to DPV (all components)

Invalid Secondary (CC) Input address primary number (street number) matched to DPV but secondary number not matched

Missing Secondary (N1) Input address primary number matched to DPV but secondary number missing

Missing Number (M1) Input address primary number missing

Invalid Number (M3) Input address primary number invalid

Missing PO/RR/HC (P1) Input address PO, RR or HC box number missing

Invalid PO/RR/HC (P3) Input address PO, RR or HC box number invalid

CMRA Coded (RR) Input address matched to CMRA

Missing Secondary (R1) Input address matched to CMRA but secondary number not present

Military Coded (F1) Address was coded to a military address

Genl Delivery Coded (G1) Address was coded to a General Delivery address

Unique Zip Coded (U1) Address was coded to a unique Zip Code

Statistics:

High Rise Default The number of multi-delivery point addresses (apartment/suite) where the apartment/suite number is not present or is invalid.

High Rise Exact The number of multi-delivery point addresses (apartment/suite) where the apartment/suite number is present and matches a ZIP+4 range.

Rural Route Default The number of Rural or Highway Contract Route addresses where the box number is not present or is invalid.

Rural Route Exact The number of Rural Route or Highway Contract addresses where the box number is present and matches a ZIP+4 range.

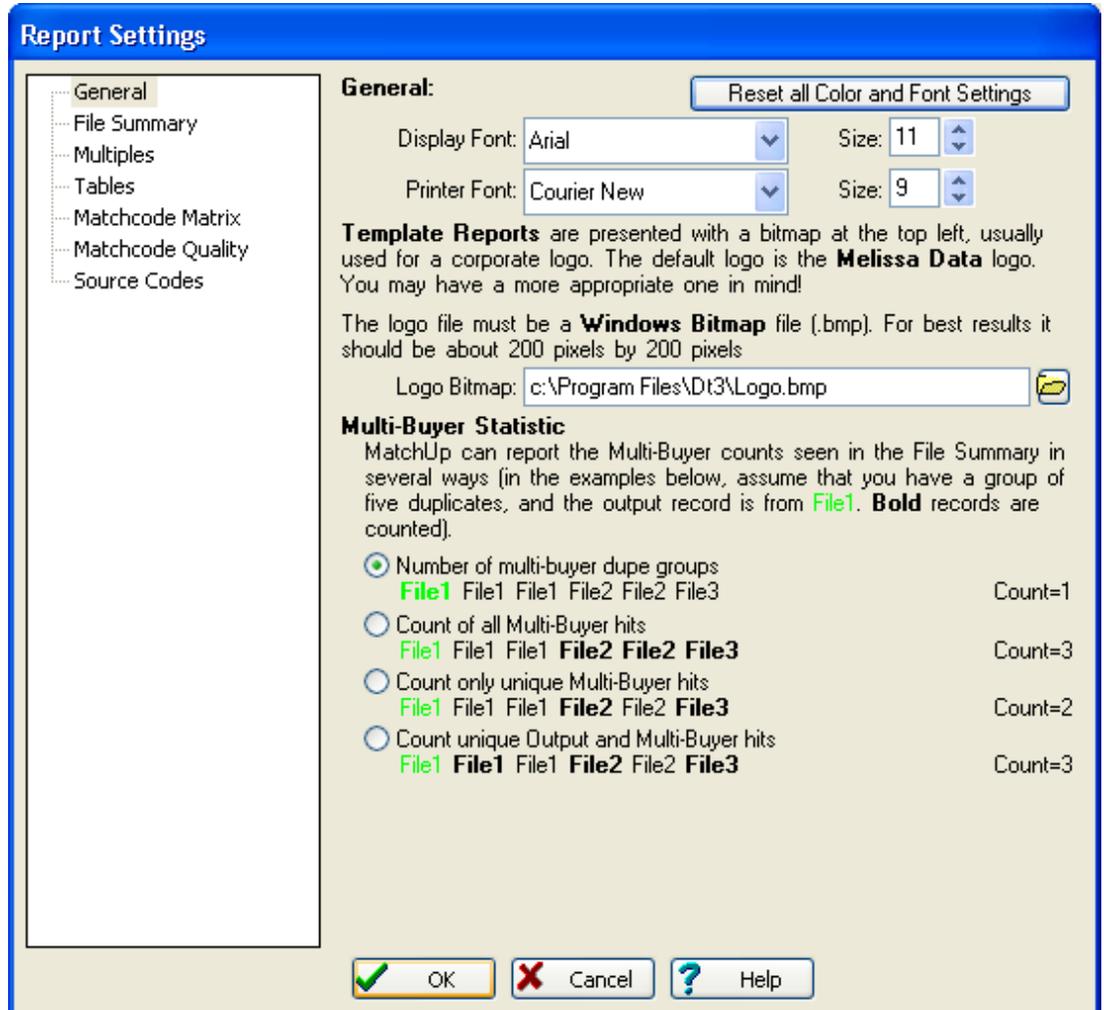
LACS Locatable Address Conversion Service. The total number of records which were flagged as having undergone a change to a city-style address (to allow emergency services to locate these addresses more efficiently). The address change is not reflected here.

EWS Early Warning System Data. Count of new addresses scheduled to be included in the USPS national database, new construction projects for example, which were flagged by the EWS file.

CASS database date The date of your CASS database files.

CASS database expires The expiration date of your CASS database files.

6.2.9 Report Settings



The report settings dialog gives you the ability to change many of the ways MatchUp presents its reports. Of most interest is the General branch, which allows you to change the display and printer font, as well as specify the bitmap file that should be used as a logo at the top of your Template format reports.

The Multi-Buyer count that appears on the File Summary and Source Code File Summary can be reported in a variety of ways, as shown in the examples on the screen.

The other branches on this dialog allow you to change the colors used in display the various charts and graphs on the reporting screen.

6.2.10 Report Templates

All of MatchUp's reports can be modified to a certain extent. Each format (Excel, Word, or Template) is handled differently. For more information on this topic, please see the Help files section on Report Templates.

Report Templates: Multi-Buyer Counting

The multi-buyer statistic is relatively new to MatchUp. One of the problems we had with introducing this statistic is that there are several differing opinions about exactly what this statistic should represent. When creating a Report Template, you should use the statistic that applies to your situation.

In the examples below, assume that we are talking about a single set of 6 matching records, from 3 different tables (FileA, FileB, and FileC). The output record is from FileA. **Bold** records indicates records that are counted in the Multi-Buyer statistic.

Multi-Buy 0:

This is what most people think of when you say "Multi-Buyer Count". It is simply a total of the number of output records from a table that have dupes from another table.

FileA FileA FileA FileB FileB FileC Multi-Buyer Count is 1.

Multi-Buy 1:

Records coming from the same table as the output record are not counted.

FileA FileA FileA **FileB FileB FileC** Multi-Buyer Count is 3.

Multi-Buy 2:

This is a popular alternative to Multi-Buy 1. It is generated the same way with the exception that records coming from the same table are not counted more than once.

FileA FileA FileA **FileB FileB FileC** Multi-Buyer Count is 2.

Multi-Buy 3:

This spin-off is similar to Multi-Buy 2 except that dupe records coming from the same table as the output record are counted.

FileA **FileA** FileA **FileB FileB FileC** Multi-Buyer Count is 3.

Multi-Buy 4:

Most people don't consider this count a multi-buyer count at all. It is generated the same way as Multi-Buy 3 except that multiple records from the same table are counted.

FileA **FileA FileA FileB FileB FileC** Multi-Buyer Count is 5

Multi-Buy 5:

This one isn't really a multi-buyer count either. Basically, all records in the dupe group are counted.

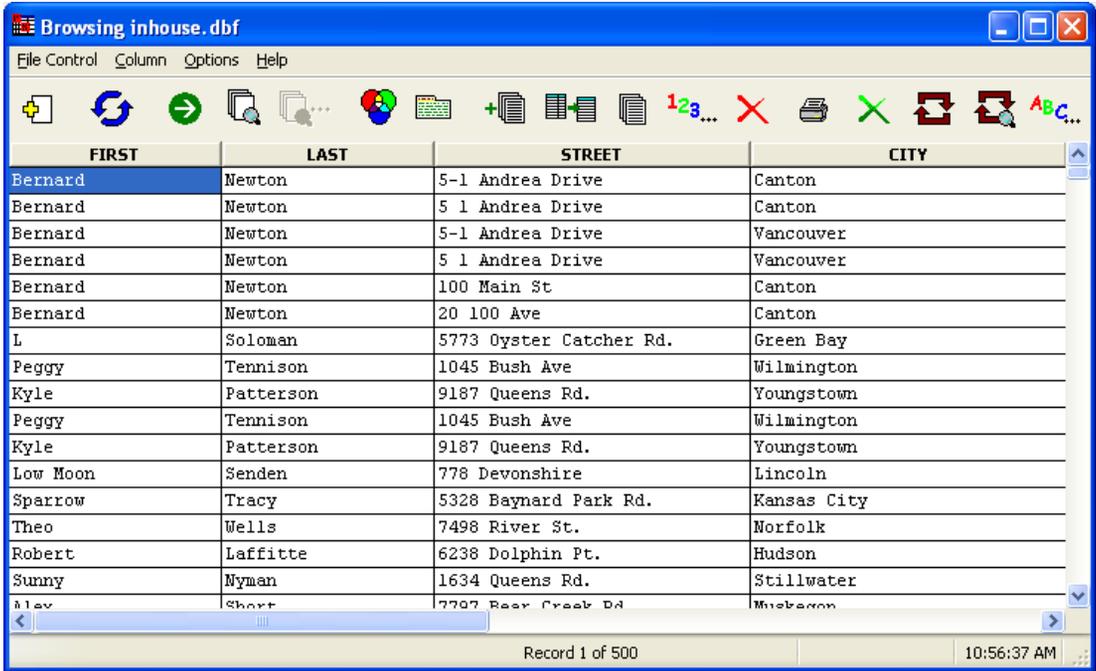
FileA FileA FileA FileB FileB FileC Multi-Buyer Count is 6.

Multi-Buy:

The MultiBuy macro having no number tells MatchUp to use whatever multi-buyer setting has been selected in **Analyze | Reports | Settings**. This is the selection provided in MatchUp's 'canned' templates.

7 Tools

7.1 Browse



FIRST	LAST	STREET	CITY
Bernard	Newton	5-1 Andrea Drive	Canton
Bernard	Newton	5 1 Andrea Drive	Canton
Bernard	Newton	5-1 Andrea Drive	Vancouver
Bernard	Newton	5 1 Andrea Drive	Vancouver
Bernard	Newton	100 Main St	Canton
Bernard	Newton	20 100 Ave	Canton
L	Soloman	5773 Oyster Catcher Rd.	Green Bay
Peggy	Tennison	1045 Bush Ave	Wilmington
Kyle	Patterson	9187 Queens Rd.	Youngstown
Peggy	Tennison	1045 Bush Ave	Wilmington
Kyle	Patterson	9187 Queens Rd.	Youngstown
Low Moon	Senden	778 Devonshire	Lincoln
Sparrow	Tracy	5328 Baynard Park Rd.	Kansas City
Theo	Wells	7498 River St.	Norfolk
Robert	Laffitte	6238 Dolphin Pt.	Hudson
Sunny	Nyman	1634 Queens Rd.	Stillwater
Alex	Short	7707 Bear Creek Rd	Wheaton

Tools | Browse allows you to view and modify a table's contents. Each row of the table represents one record.

Depending on the type of table you are browsing, you may or may not be allowed to edit its contents. If MatchUp cannot allow you to edit records, you will be warned when the browser window is first displayed.

Many of the Browse commands use a dBASE expression in one way or another. Even if your table is not a dBASE formatted table (i.e. Access, Excel, ASCII, etc), a dBASE expression should be used. MatchUp "corrects" the statement appropriately.

Keyboard Shortcuts:

LEFT, RIGHT	Move the highlight left/right one column.
CTRL+LEFT, CTRL+RIGHT	Pan the browser left/right one column.
HOME, END	Move the highlight to the first/last column.
UP, DOWN	Move the highlight up/down one row.
CTRL+UP, CTRL+DOWN	Pan the browser up/down one row.
CTRL+HOME, CTRL+END	Move the highlight to the first/last row.
PAGEUP, PAGEDOWN	Move the highlight up/down one screen.
CTRL+PAGEUP,CTRL+PAGEDOWN	Pand the browser up/down one screen.

Editing Records:

In the browser, double-click (or hit SPACE or ENTER) the desired field to edit. Hit TAB, ENTER or an arrow key to save changes, ESC to cancel. ENTER or TAB will keep you in a continuous editing state (see [Options | Settings](#) to modify this behavior).

Resizing a Column:

Move the pointer between the gray column headings until it becomes a , then drag the column's edge to the desired size.

Moving a Column:

Click the column's gray heading area, then drag it to a new location.

Changing what a Column Displays:

You can change what data is displayed in a column by double-clicking its gray heading area. You will then have the option to display a specific field or the result of a dBASE expression (see [Column | Change Column](#)). Note that the same field can appear as many times as you like!

Adding Records:

Select [File Control | New Record](#) or hit CTRL+N. You don't have to be at the bottom of the table to add records (but the record *will always* be added to the end). This option is not available for all database types.

Deleting Records:

To mark (or unmark) a dBASE record for deletion, either select [File Control | Toggle Delete](#) or hit CTRL+T. Records marked for deletion can be permanently removed with [File Control | Pack Table](#).

File Control:

New Record Add a new empty record to the end of the table (dBASE files only).

Toggle Delete Mark (or unmark) the current record for deletion (dBASE files only).

Goto Record Jump to a user specified record number.

Find Record Locate a record fitting a specified criteria.

Find Again Locate the next record fitting the previously specified search criteria.

Set Filter View only records that meet a specified criteria.

Set Index View records in a specified sequence.

Append Records Add records from one table into the current table.

Concatenate Fields Join up to eight fields into a single field.

Copy Records Copy records fitting a specified criteria to a new file.

Count Records Count the number of records that meet a specified criteria.

Delete Records Mark for deletion records that meet a specified criteria.

Print Records Print records fitting a specified criteria.

Recall Records Unmark for deletion records that meet a specified criteria.

Replace/With Replace the contents of a field with the result of a specified expression.

Search & Replace Search for text and replace it with something else.

Sort Records Sort the table by a specified criteria (dBASE files only).

Pack Table Physically and permanently remove records that have been marked for deletion (dBASE files only).

Split Table Split the table in a variety of ways.

Vertical Display Display the columns of the currently highlighted record in a vertical window.

Column:

Add Column Display a new column in the browser.

Remove Column Remove an existing column from the browser.

Change Column Change what data is displayed in a column.

Auto-Size Column Change the width of a column to fit the longest string.

Auto-Size All Columns Change the width of each column to fit its longest string.

All Columns This Size Change the width of each column to the size of a column.

Reset Columns Reset all column widths to their original sizes. Filter and Index settings are also reset.

Options:

Settings Modify the browser's display settings.

No Toolbar Hide the browser's toolbar.

Small Toolbar Display the small toolbar.

Medium Toolbar Display the medium toolbar.

Large Toolbar Display the large toolbar.

7.1.1 Find Record



Search for Text The word or phrase that you want to locate.

Ignore Case Check if you want to locate the text regardless of the case.

Search Select how you would like to search:

- **Any Field** Every field will be searched for the specified text.
- **Selected Field(s)** The checked fields will be searched for the specified text.
- **By Expression** Search until a record matches the specified criteria. The for expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.

Search from Start Start searching from the first record.

Search from Here Start searching from the current record.

Note:

Only rows that are currently being displayed are searched. If you have used Set Filter to hide some of the rows, you may not find what you're looking for.

7.1.2 Set Filter



Allows you to view records that meet certain criteria. A filter works just like a "for" condition except that it remains active until you either change or remove the filter. If you Copy, Sort, etc. once a filter is been set, the program will perform those actions only on the records that meet your filter criteria. You'll notice that subsequent File Control options will have the current filter condition listed in the "for" condition box. When a filter is set, the Set Filter tool will have a check next to its icon on the toolbar:



Show All Records (no filter) Removes the current filter.

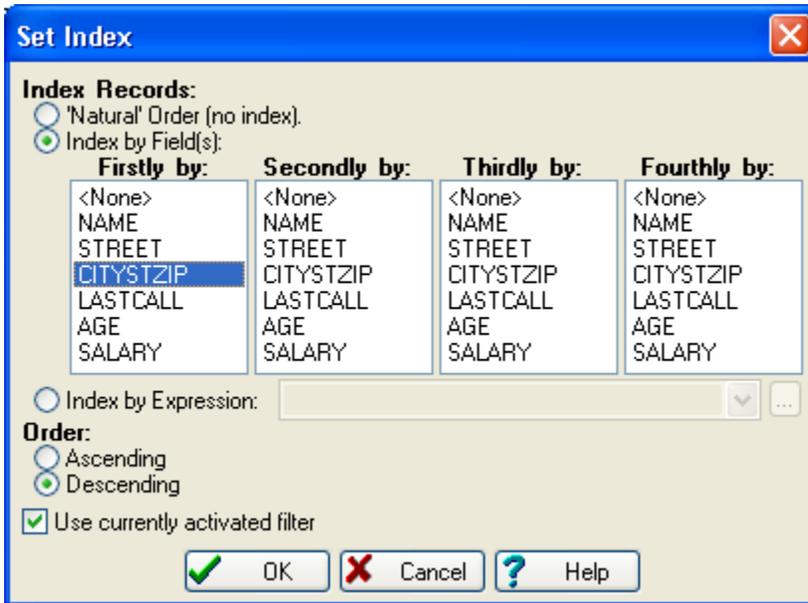
Filter by Expression Sets the filter to the expression specified to the right. The filter expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.

Use currently activated index If an index has been set, you can check this box to keep that index in effect. If not checked, the index will be reset to the table's natural order.

Display Filtered Records In what manner filtered records should be shown:

- **Don't Display Filtered Records** When browsing, only records meeting the filter expression will be displayed.
- **Show Filtered Records in an Alternate Color** When browsing, records not meeting the filter expression will be displayed in the selected color.

7.1.3 Set Index



Allows you to view records in a certain sequence. This only effects the order that records are displayed and not the physical record order. If you want to change the physical order of a table, use **File Control | Sort Records**. When an index is set, the Set Index tool will have a check next to its icon on the toolbar:



Index Records How you would like to order the records:

- **'Natural' Order (no index)** Removes the active index.
- **Index by Field(s)** Indexes by the field(s) you select.
- **Index by Expression** Indexes the file according to the expression specified to the right. The index expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.

Order Sequencing order:

- **Ascending** "A" before "B", "1" before "2", "03/10/68" before "04/10/68".
- **Descending** "C" before "B", "3" before "2", "05/10/68" before "04/10/68".

Use currently activated index If a filter has been set, you can check this box to keep that filter in effect. If not checked, the filter will be reset to show all records.

7.1.4 Append Records

Append records from another table onto the end of the current table. This option is not available for all table types.

When you first select this option, you are prompted for an input table. Once you have selected an input table, you will see the Append Records dialog:



The image shows a dialog box titled "Append Records" with a blue header. Below the title, the text "Append Records:" is followed by two red notes. The first note states: "Note: Field names and field lengths should match between the two files." The second note states: "Note: The settings below apply to the *Input Table*, not the table currently being browsed. This is unlike any of the other File Control functions." Below the notes, there are two radio button options: "All Records in the Table." (which is selected) and "Records meeting these conditions:". Under the second option, there are three checkboxes: "Record Range:" with "From: 1" and "To: 0" spinners; "For (Expression):" with a text field and a dropdown arrow; and "While (Expression):" with a text field and a dropdown arrow. At the bottom of the dialog are three buttons: "OK" with a green checkmark, "Cancel" with a red X, and "Help" with a blue question mark.

Note: Unlike any other File Control tool, the Record Range settings apply to the *input table*, not the current table.

Note Field names and lengths should match between the two tables. Data in fields from the input table that don't have a matching field (ie, a field with the same name) in the current table *will not* be added.

All Records in the Table All records in the input table will be appended.

Records meeting these conditions Only append records meeting the conditions:

- **Record Range** Append records between the **From** and **To** range. To process from a specific record all the way to the end of the file, specify "0" as the **To** record.
- **For (Expression)** Restrict appending to only records that meet the specified criteria. The for expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.
- **While (Expression)** Stop appending when records no longer meet the specified criteria. The while expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.

7.1.5 Concatenate Fields

'Concatenate' is just a fancy word for 'join'. It makes us feel superior when we use it.

Concatenate Fields

Input Fields:

Input Field 1: <None> Input Field 5: <None>
Input Field 2: <None> Input Field 6: <None>
Input Field 3: <None> Input Field 7: <None>
Input Field 4: <None> Input Field 8: <None>

Output Fields:

Output Field: <None>

If data exists in the output field:

Append new data to the existing data
 Overwrite the existing data

Remove leading/trailing spaces:

Trim leading spaces
 Trim trailing spaces

Separate fields with:

Single space
 No separator
 This separator:

Records:

All Records in the Table.
 Records meeting these conditions:

Record Range: From: 1 To: 0

For (Expression):

While (Expression):

OK Cancel Help

Input Fields Select the fields that you want to join together, in the order you want them joined.

Output Fields Select the field that will receive the joined result.

If data exists in the output field Select what should happen if there's already something in the **Output Field**:

- **Append new data to the existing data** The joined result will be tacked onto the end of the existing data.
- **Overwrite the existing data** The joined result will replace any existing data.

Separate fields with Select what should be inserted between the joined data:

- **Single space** "John" and "Smith" will become "John Smith".
- **No separator** "John" and "Smith" will become "JohnSmith".
- **This separator** "John" and "Smith" will become "John<your connector>Smith".

Remove leading/trailing spaces Select whether spaces should be trimmed:

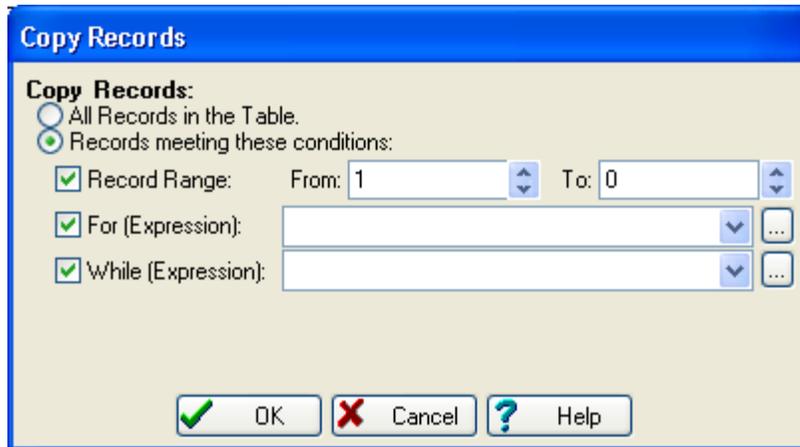
- **Trim leading spaces** Remove spaces from the beginning of the field.
- **Trim trailing spaces** Remove spaces from the end of the field.

Records You can process all or only part of the table:

- **All Records in the Table** To process all records.
- **Records meeting these conditions** To process only certain records:
 - **Record Range** Process records between the **From** and **To** range. To process from a specific record all the way to the end of the file, specify "0" as the **To** record.
 - **For (Expression)** Restrict processing to only records that meet the specified criteria. The for expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.
 - **While (Expression)** Stop processing when records no longer meet the specified criteria. The while expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.

7.1.6 Copy Records

Copy records fitting a specified criteria to a new table.



Copy Records

Copy Records:

All Records in the Table.

Records meeting these conditions:

Record Range: From: 1 To: 0

For (Expression):

While (Expression):

OK Cancel Help

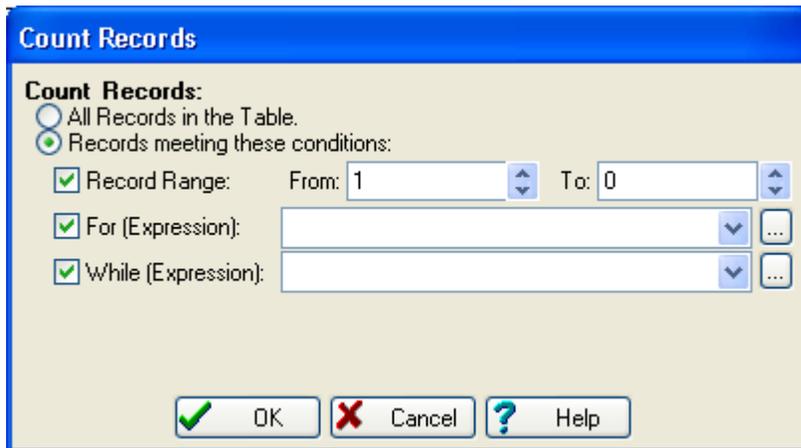
Copy Records You can copy all or only part of the table:

- **All Records in the Table** To copy all records.
- **Records meeting these conditions** To copy only certain records:
 - **Record Range** Copy records between the **From** and **To** range. To copy from a specific record all the way to the end of the file, specify "0" as the **To** record.
 - **For (Expression)** Restrict copying to only records that meet the specified criteria. The for expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.
 - **While (Expression)** Stop copying when records no longer meet the specified criteria. The while expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.

When you have completed this dialog and hit **OK**, you are prompted for the name of an output table.

7.1.7 Count Records

Count the number of records meeting a specified criteria.



Count Records You can count all or only some of the records:

- **All Records in the Table** To count all records. Not particularly useful when counting.
- **Records meeting these conditions** To count only certain records:
 - **Record Range** Count records between the **From** and **To** range. To count from a specific record all the way to the end of the file, specify "0" as the **To** record.
 - **For (Expression)** Restrict counting to only records that meet the specified criteria. The for expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.
 - **While (Expression)** Stop counting when records no longer meet the specified criteria. The while expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.

7.1.8 Delete Records

Mark for deletion records meeting a specified criteria. This option is only available for dBASE files.

Delete Records

Mark Records for Deletion:

All Records in the Table.

Records meeting these conditions:

Record Range: From: 1 To: 0

For (Expression):

While (Expression):

OK Cancel Help

Count Records You can delete all or only some of the records:

- **All Records in the Table** To delete all records.
- **Records meeting these conditions** To delete only certain records:
 - **Record Range** Delete records between the **From** and **To** range. To count from a specific record all the way to the end of the file, specify "0" as the **To** record.
 - **For (Expression)** Restrict deleting to only records that meet the specified criteria. The for expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.
 - **While (Expression)** Stop deleting when records no longer meet the specified criteria. The while expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.

7.1.9 Print Records

Print records meeting a specified criteria. The printed output will match the current browser's appearance: column contents, positions, sizes, filtering and ordering.

Print Records

Print Records:

Printer: Acrobat PDFWriter

Font: Arial Size: 10

Field Structure

Records

Page headings and page numbers

Field headings

Records:

All Records in the Table.

Records meeting these conditions:

Record Range: From: 1 To: 0

For (Expression):

While (Expression):

Printer Select the printer to output to.

Font/Size The font to be used for printing.

Properties Displays the printer's **Printer Setup** dialog. You can usually change settings such as Paper Orientation, Paper Size and Print Quality. This dialog is dependant on your printer driver, so your mileage may vary.

Page Setup Displays the Windows **Page Setup** dialog. You can change the Paper Size, Orientation and Margins.

Field Structure Check this box to print the table's structure.

Records Check to print the table's contents.

Page headings and page numbers Check to print each page with a title indicating the name of the table, as well as a page number.

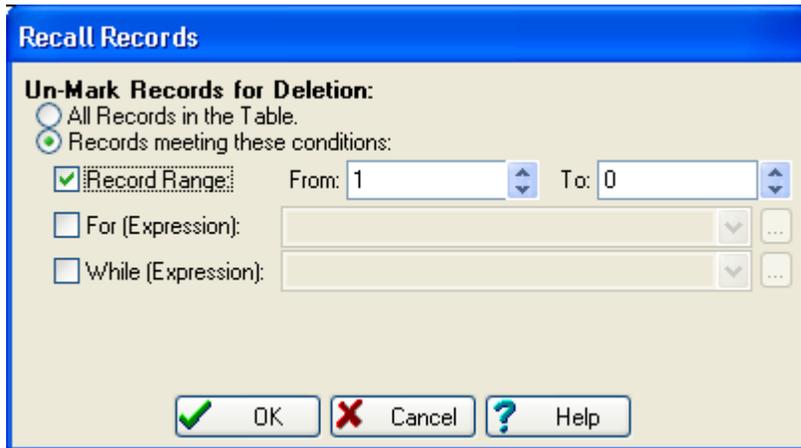
Field headings Check this box to print field names at the top of each page.

Records You can print all or only some of the records:

- **All Records in the Table** To print all records.
- **Records meeting these conditions** To print only certain records:
 - **Record Range** Print records between the **From** and **To** range. To count from a specific record all the way to the end of the file, specify "0" as the **To** record.
 - **For (Expression)** Restrict printing to only records that meet the specified criteria. The for expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.
 - **While (Expression)** Stop printing when records no longer meet the specified criteria. The while expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.

7.1.10 Recall Records

Remove deletion marks from records meeting a specified criteria. This option is only available for dBASE files.



Un-Mark Records for Deletion You can unmark all or only some of the records:

- **All Records in the Table** To unmark all records.
- **Records meeting these conditions** To unmark only certain records:
 - **Record Range** Unmark records between the **From** and **To** range. To count from a specific record all the way to the end of the file, specify "0" as the **To** record.
 - **For (Expression)** Restrict unmarking to only records that meet the specified criteria. The for expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.
 - **While (Expression)** Stop unmarking when records no longer meet the specified criteria. The while expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.

7.1.11 Replace/With

Replace the contents of a field with the result of a specified expression.

Replace Records

Replace Field:

With Expression:

Replace Records:

All Records in the Table.

Records meeting these conditions:

Record Range: From: To:

For (Expression):

While (Expression):

Replace Field Select the field that will receive the result of the **With Expression**.

With Expression Specify the dBASE expression that will determine the new contents of the **Replace Field**. The contents must be of the same type of data as the **Replace Field**. For example, if the **Replace Field** is a Numeric field, then this expression must result in a number. You can use the Expression Builder if you need help.

Replace Records You can process all or only some of the records:

- **All Records in the Table** To process all records.
- **Records meeting these conditions** To process only certain records:
 - **Record Range** Process records between the **From** and **To** range. To count from a specific record all the way to the end of the file, specify "0" as the **To** record.
 - **For (Expression)** Restrict processing to only records that meet the specified criteria. The for expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.
 - **While (Expression)** Stop processing when records no longer meet the specified criteria. The while expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.

When you have completed this dialog and hit **OK**, you are prompted for a Processing Mode.

7.1.12 Search & Replace

Search for specific text and replace it with another.

Search & Replace

Source:

Screen
 File:

Fields:

Search & Replace all fields
Search Field: Replace Field:
 Delete 'Search for' string from Search Field

Search & Replace:

Search for: Blank Field
Replace with: Blank Field

Options:

Ignore case Whole word Multiple occurrences
 Any occurrence First occurrence

Search & Replace Records:

All Records in the Table.
 Records meeting these conditions:

Record Range: From: To:

For (Expression):

While (Expression):

OK Cancel Help

Source How the Search & Replace should be directed:

- **Screen** The search and replace text will be entered in the **Search for** and **Replace with** prompts.
- **File** The search and replace text will be retrieved from the specified file. This file must be a dBASE database with two character fields. The first field is used as the search text, the second as the replace text.

Search & Replace all fields Every field in the table will be searched.

Search Field The field to search.

Replace Field The field to replace into. The default is to use the **Search Field**, which is usually what you want.

Delete 'Search for' string from Search Field If you chose a **Replace Field** that is different than the **Search Field**, you can remove the **Search for** text from the original by checking this box.

Search for The text to search for. If you don't elect to ignore case, then be sure to enter the data exactly as desired. Checking **Blank Field** searches for a blank **Search Field**.

Replace with The text to replace the **Search for** text with. It will be used exactly as you enter it (i.e., it is case dependant). Checking Blank Field will clear the **Replace Field** when the **Search for** text is found.

Ignore Case Check if you want to search regardless of the case.

Whole word/Any occurrence If **Whole word** is selected, "IT" will be found in "SMITH", "SMIT" or "ITH". In order to be considered a whole word, the search text must occur as a word or a series of words. If **Any occurrence** is selected, the text may appear anywhere in a field.

Multiple occurrences/First occurrence If **Multiple occurrences** is selected, each and every occurrence of the search text in a field is replaced. If **First occurrence** is select, only the first one is replaced.

Search & Replace Records You can process all or only some of the records:

- **All Records in the Table** To process all records.
- **Records meeting these conditions** To process only certain records:
 - **Record Range** Process records between the **From** and **To** range. To count from a specific record all the way to the end of the file, specify "0" as the **To** record.
 - **For (Expression)** Restrict processing to only records that meet the specified criteria. The for expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.
 - **While (Expression)** Stop processing when records no longer meet the specified criteria. The while expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.

When you have completed this dialog and hit **OK**, you are prompted for a Processing Mode.

Melissa Data has gone through several different iterations of methodology to determine leading and trailing spaces (DOS users will remember the degree symbol and "plaiditer" systems). With our Windows programs, you just need to type in the phrase just as you would like it to be searched or replaced. For example, if you want to search for "and", with trailing and leading spaces, enter it as " and " (note the leading and trailing spaces; don't type the

double quotes). The trailing space(s) won't appear unless you highlight the contents of the entry (in the field, hit CTRL+A).

If you are using a Search and Replace file, you can use the original method to denote trailing spaces, the degree symbol (ASCII character 248).

7.1.13 Sort Records

Physically sorts a table by a specified criteria. This option is only available for dBASE files.

Sort Records

Sort Records:

Sort by Field(s):

Firstly by:	Secondly by:	Thirdly by:	Fourthly by:
<None>	<None>	<None>	<None>
NAME	NAME	NAME	NAME
STREET	STREET	STREET	STREET
CITYSTZIP	CITYSTZIP	CITYSTZIP	CITYSTZIP
LASTCALL	LASTCALL	LASTCALL	LASTCALL
AGE	AGE	AGE	AGE
SALARY	SALARY	SALARY	SALARY

Sort by Expression: ...

Order:

Ascending

Descending

Records:

All Records in the Table.

Records meeting these conditions:

Record Range: From: To:

For (Expression): ...

While (Expression): ...

Records that don't meet the above conditions:

Should precede sorted records

Should follow sorted records

Should be removed from the table (CAUTION!!!!)

Sort Records How you would like to order the records:

- **Sort by Field(s)** Indexes by the field(s) you select.
- **Sort by Expression** Sorts the file according to the expression specified to the right. The for expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.

Order Sequencing order:

- **Ascending** "A" before "B", "1" before "2", "03/10/68" before "04/10/68".
- **Descending** "C" before "B", "3" before "2", "05/10/68" before "04/10/68".

Records You can sort all or only some of the records:

- **All Records in the Table** To sort all records.
- **Records meeting these conditions** To sort only certain records:
 - **Record Range** Sort records between the **From** and **To** range. To count from a specific record all the way to the end of the file, specify "0" as the **To** record.
 - **For (Expression)** Restrict sorting to only records that meet the specified criteria. The for expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.
 - **While (Expression)** Stop sorting when records no longer meet the specified criteria. The while expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.

Records that don't meet the above conditions If **Record Range**, **For** or **While** conditions are specified, you need to indicate what to do with records that don't satisfy the conditions:

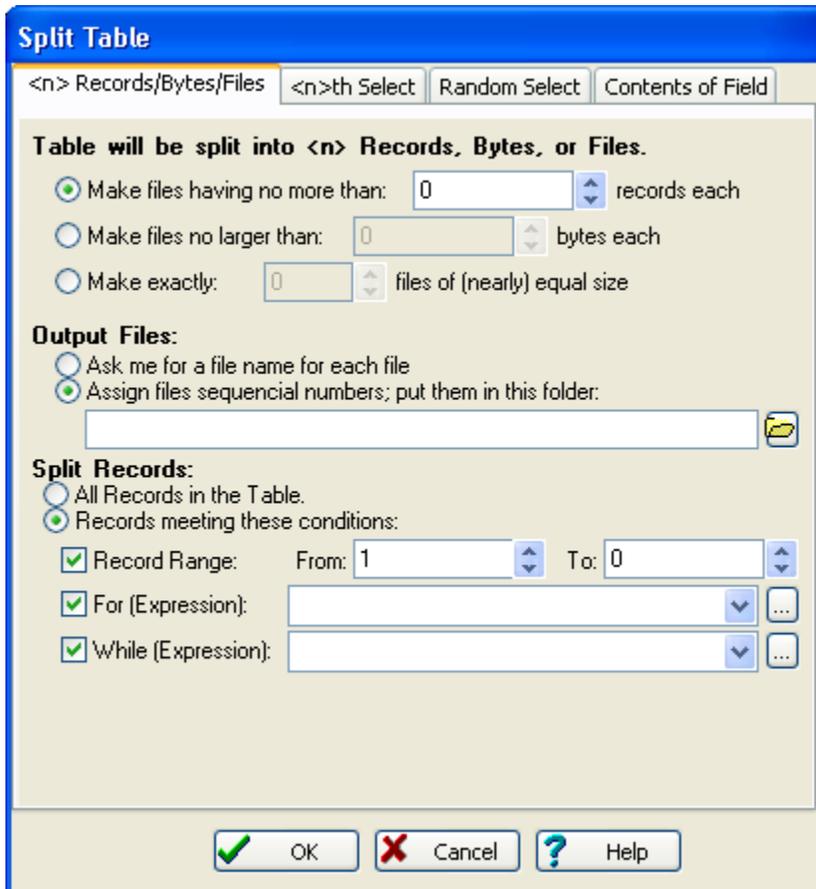
- **Should precede sorted records** Records not matching the criteria will be placed (in 'natural' order) before the sorted records.
- **Should follow sorted records** Records not matching the criteria will be placed after the sorted records.
- **Should be removed from the table** Records not matching the criteria will be **permanently removed**.

7.1.14 Split Table

Splits the table into one or more sub-tables using the method you select (indicated on the tabs at the top of the dialog):

- **<n> Records/Bytes/Files** Split the table so that the resultant tables are no larger than the specified size.
- **<n> th Select** Split the table so that every nth record is taken.
- **Random Select** Randomly select a specified number of records from the table.
- **Contents of Field** Use a source field to determine which resultant table a record should be copied. Commonly used for undoing a Merge/Purge or dividing a large table up by Zip Code or SCF.

Split Table: <n> Records/Bytes/Files



Split Table

<n> Records/Bytes/Files <n>th Select Random Select Contents of Field

Table will be split into <n> Records, Bytes, or Files.

Make files having no more than: 0 records each

Make files no larger than: 0 bytes each

Make exactly: 0 files of (nearly) equal size

Output Files:

Ask me for a file name for each file

Assign files sequential numbers; put them in this folder:



Split Records:

All Records in the Table.

Records meeting these conditions:

Record Range: From: 1 To: 0

For (Expression):  

While (Expression):  

OK Cancel Help

Table will be split into <n> Records, Bytes, or Files Select the splitting method:

- **Make files having no more than...records each** Each output table will have the specified number of records (the last table may have fewer).
- **Make files no larger than...bytes each** Each output table will be the same size (the last table may be smaller).
- **Make exactly...files of (nearly) equal size** Records will be evenly divided into a specified number of tables.

Output Files Select how output tables will be named:

- **Ask me for a file name for each file** Each time a file is created, you will be prompted for a name.
- **Assign files sequential numbers; put them in this folder** Files will be named 0000001.dbf, 0000002.dbf, etc, and stored in the specified folder.

Split Records You can process all or only some of the records:

- **All Records in the Table** To process all records.
- **Records meeting these conditions** To process only certain records:
 - **Record Range** Process records between the **From** and **To** range. To count from a specific record all the way to the end of the file, specify "0" as the **To** record.
 - **For (Expression)** Restrict processing to only records that meet the specified criteria. The for expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.
 - **While (Expression)** Stop processing when records no longer meet the specified criteria. The while expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.

Split Table: <n>th Select

Split Table

<n> Records/Bytes/Files <n>th Select Random Select Contents of Field

Take every: 0 th record from the table

Also mark selected records for deletion

Split Records:

All Records in the Table.

Records meeting these conditions:

Record Range: From: 1 To: 0

For (Expression):

While (Expression):

OK Cancel Help

Take every...th record from the table Specify the gap between each record selected. For example, "4" will select every fourth record or 25% of the table.

Also mark selected records for deletion Selected records are marked for deletion. This option is provided so that you can provide incremental selections. For example, if you later wanted to process only the records that were previously selected, you could specify a **For Expression** of `deleted()`. This option is only available for dBASE files.

Split Records You can process all or only some of the records:

- **All Records in the Table** To process all records.
- **Records meeting these conditions** To process only certain records:
 - **Record Range** Process records between the **From** and **To** range. To count from a specific record all the way to the end of the file, specify "0" as the **To** record.
 - **For (Expression)** Restrict processing to only records that meet the specified criteria. The for expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.
 - **While (Expression)** Stop processing when records no longer meet the specified criteria. The while expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.

Split Table: Random Select

The screenshot shows the 'Split Table' dialog box with the 'Random Select' tab selected. The dialog has a blue title bar and a light beige background. At the top, there are four tabs: '<n> Records/Bytes/Files', '<n>th Select', 'Random Select', and 'Contents of Field'. The 'Random Select' tab is active. Below the tabs, there is a text field labeled 'Randomly select:' containing the number '0', followed by a small up/down arrow icon and the text 'records from the table'. Below this is a checkbox labeled 'Also mark selected records for deletion' which is currently unchecked. Underneath is the section 'Split Records:' with two radio buttons: 'All Records in the Table.' (unchecked) and 'Records meeting these conditions:' (checked). Below the radio buttons are three checked checkboxes: 'Record Range:' with 'From: 1' and 'To: 0' (both with up/down arrows), 'For (Expression):' with an empty text box and a dropdown arrow, and 'While (Expression):' with an empty text box and a dropdown arrow. At the bottom of the dialog are three buttons: 'OK' with a green checkmark icon, 'Cancel' with a red X icon, and 'Help' with a blue question mark icon.

Randomly select...records from the table Specify how many records should be selected.

Also mark selected records for deletion Selected records are marked for deletion. This option is provided so that you can provide incremental selections. For example, if you later wanted to process only the records that were previously selected, you could specify a **For Expression** of `deleted()`. This option is only available for dBASE files.

Split Records You can process all or only some of the records:

- **All Records in the Table** To process all records.
- **Records meeting these conditions** To process only certain records:
 - **Record Range** Process records between the **From** and **To** range. To count from a specific record all the way to the end of the file, specify "0" as the **To** record.
 - **For (Expression)** Restrict processing to only records that meet the specified criteria. The for expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.
 - **While (Expression)** Stop processing when records no longer meet the specified criteria. The while expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.

Split Table: Contents of Field

The screenshot shows the 'Split Table' dialog box with the 'Contents of Field' tab selected. The dialog has a blue title bar and a light beige background. At the top, there are four tabs: '<n> Records/Bytes/Files', '<n>th Select', 'Random Select', and 'Contents of Field'. The 'Contents of Field' tab is active. Below the tabs, there are three input fields: 'Base split on contents of field:' with a dropdown menu showing 'STREET', 'Start at character position:' with a text box containing '0' and up/down arrows, and 'Number of characters to use:' with a text box containing '0' and up/down arrows. Under the heading 'Output Files:', there are three radio button options. The first is 'Ask me for a file name for each file'. The second is 'Use the contents of the field as the file name; put them in this folder:', which is selected, and has an empty text box and a folder icon to its right. The third is 'Assign files sequential numbers; put them in this folder:', which has an empty text box and a folder icon to its right. Under the heading 'Split Records:', there are three radio button options. The first is 'All Records in the Table'. The second is 'Records meeting these conditions:', which is selected. Below this are three checked checkboxes: 'Record Range:' with 'From:' and 'To:' text boxes containing '1' and '0' respectively; 'For (Expression):' with a text box and a dropdown arrow; and 'While (Expression):' with a text box and a dropdown arrow. At the bottom of the dialog are three buttons: 'OK' with a green checkmark icon, 'Cancel' with a red X icon, and 'Help' with a blue question mark icon.

Base split on contents of field Select the field whose contents will determine to which table a record will be copied.

Start at character position Specify the point in the field at which data should be extracted for the split.

Number of characters to use Specify how many characters should be used to determine the split (counting starts at the **Start Position**, not the start of the field).

Output Files Select how output tables will be named:

- **Ask me for a file name for each file** Each time a file is created, you will be prompted for a name.
- **Use the contents of the field as the file name; put them in this folder** The contents of the field will be used as the file name (illegal characters are stripped) and stored in the specified folder.
- **Assign files sequential numbers; put them in this folder** Files will be named 0000001.dbf, 0000002.dbf, etc, and stored in the specified folder.

Split Records You can process all or only some of the records:

- **All Records in the Table** To process all records.
- **Records meeting these conditions** To process only certain records:
 - **Record Range** Process records between the **From** and **To** range. To count from a specific record all the way to the end of the file, specify "0" as the **To** record.
 - **For (Expression)** Restrict processing to only records that meet the specified criteria. The for expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.
 - **While (Expression)** Stop processing when records no longer meet the specified criteria. The while expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.

7.1.15 Add Column

A column will be added to the browser at the highlighted column position. Its contents will be what you specify below:

Add Column

Heading: Size: 10

Column Contents:

Expression: ...

Field:

- ADDED
- ADVERTISER
- ADDITIONS
- BUSINESS
- CITY
- CO
- EMAIL

Save & Add Another

OK Cancel Help

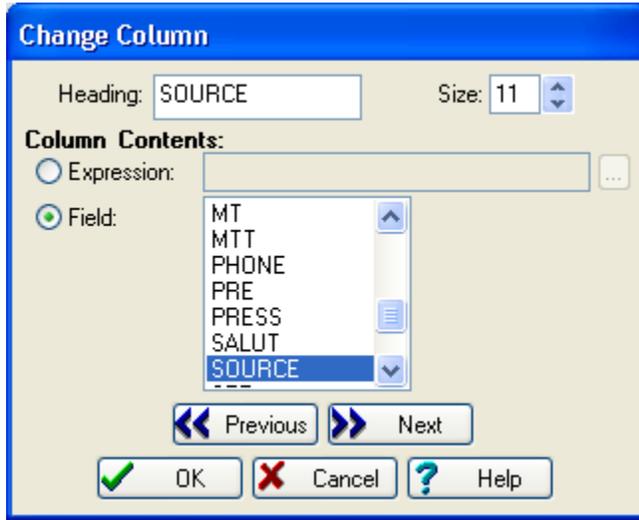
Heading/Size The name and size of the column you want to create.

Column Contents What should be displayed in the column:

- **Expression** Display the result of the specified expression as the column contents. The expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.
- **Field** Displays the contents of an existing field.

7.1.16 Change Column

A column will be added to the browser at the highlighted column position. Its contents will be what you specify below:

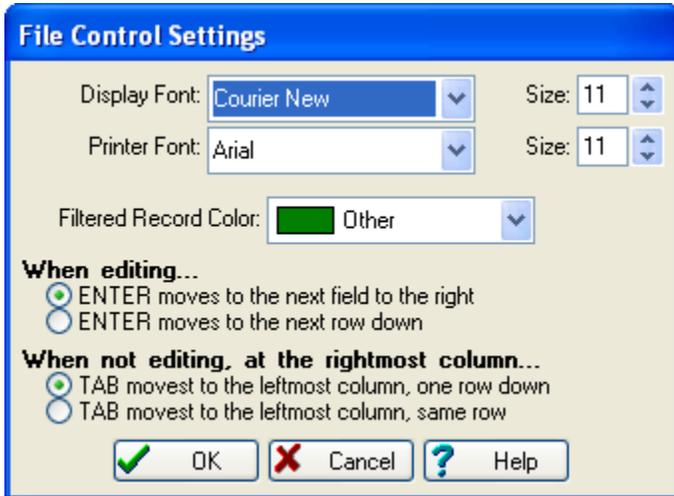


Heading/Size The name and size of the column.

Column Contents What should be displayed in the column:

- **Expression** Display the result of the specified expression as the column contents. The expression must be entered in dBASE syntax. You can use the Expression Builder if you need help.
- **Field** Displays the contents of an existing field.

7.1.17 Settings



Display Font/Size The browser's font.

Printer Font/Size The font used when printing.

Filtered Record Color The default color used for records that are being filtered.

When editing... Control the behavior of the ENTER key:

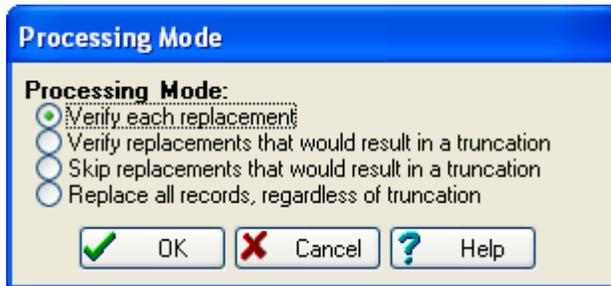
- **ENTER moves to the next field to the right** When editing a field, the ENTER key will move the cursor to the next field to the right.
- **ENTER moves to the next row down** When editing a field, the ENTER key will move the cursor to the same field, next row down.

When not editing, at the rightmost column... Control the behavior of the TAB key:

- **TAB moves to the leftmost column, one row down** When in the rightmost column, the TAB key will move the cursor one row down and circle to the first column.
- **TAB moves to the leftmost column, same row** When in the rightmost column, the cursor will circle back to the first column of the same row.

7.1.18 Processing Mode

The Processing Mode dialog appears on the Search & Replace, Replace/With and Concatenate Fields commands:



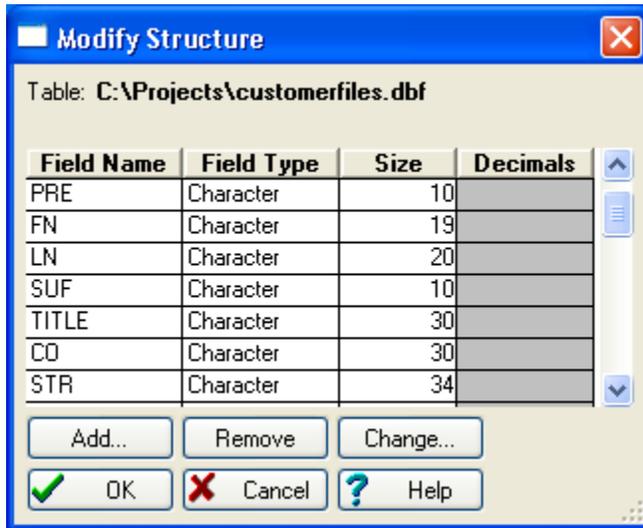
Processing Mode Specify what records should be brought to your attention for review:

- **Verify each replacement** The screen will display the current contents of the field and what it will look like after replacement. The processor will ask if you want to process the record every time.
- **Verify only when replacement that would result in a truncation** The field is displayed for review only when processing will result in a truncation. If the replacement string is longer than the search string (so a truncation would result), the processor will pause to verify whether or not to proceed with the replacement.
- **Skip replacements that would result in a truncation** In all cases the search string is automatically replaced with the replacement string *unless* the replacement will cause a truncation. No replacements are displayed to the screen.
- **Replace all records, regardless of truncation** In all cases, the field is automatically replaced with the replacement string or as much of the replacement string as will fit - no exceptions. These replacements are not displayed to the screen.

7.2 Modify Structure

Modify Structure allows you to change the structure of a dBASE table: to add new fields, delete fields, move fields, resize fields, and rename fields.

With the exception of renaming fields, all structure changes require enough disk space to make a copy of the current database.

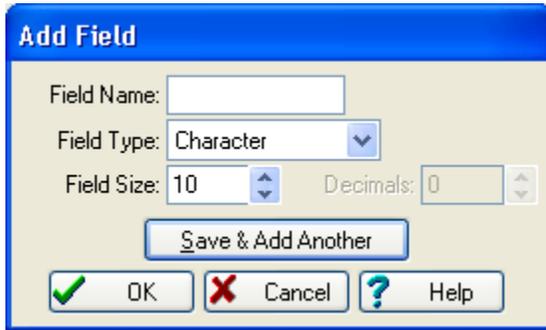


Add Add a new field at the highlight's position.

Remove Remove the currently highlighted field.

Change Modify the currently highlighted field. You can also modify fields by double-clicking the Field Name, Type, Size or Decimals entry that you want to change.

Adding Fields:



The screenshot shows a dialog box titled "Add Field". It contains the following elements:

- Field Name:** An empty text input field.
- Field Type:** A dropdown menu currently showing "Character".
- Field Size:** A spinner box set to "10".
- Decimals:** A spinner box set to "0".
- Buttons:** A "Save & Add Another" button, and three standard buttons: "OK" (with a green checkmark icon), "Cancel" (with a red X icon), and "Help" (with a blue question mark icon).

Field Name Field names must be between 1 and 10 characters. The first character must be alphabetic, but the others may be letters, numbers, or the underline (_). Embedded spaces are not allowed.

Field Type There are four options: Character, Date, Logical, and Numeric.

Field Size Date fields are automatically sized to 8 and Logical fields to 1. Character field sizes must be between 1 and 254. Numeric field sizes must be between 1 and 20. Remember that if you shorten a field, you will lose any data that exceeds the new field width.

Decimals (Numeric fields only) Decimal places must be between 0 and the field's size minus 2 (i.e., a numeric field of size 10 can have between 0 and 8 decimal places).

7.3 ASCII Conversion

7.3.1 Import from ASCII

Most database applications can convert to and from ASCII, since this is the most universal data format available. MatchUp can process files in ASCII delimited, fixed field and flat files if you specify a structure. If you select an ASCII file when you are in a setup, you will specify the structure, but you won't actually import the file.

Upon selecting this option, you are first prompted for the name and location of the ASCII file. Then, the following screen is presented:



Use the Interactive Importer Use the Interactive Importer to determine the ASCII file's structure.

Manually Enter Manually type in the ASCII file's structure.

Inherit currently selected table's structure The structure of the currently open table will be used as a starting point for the ASCII file's structure. The Interactive Importer is then displayed for you to fine-tune the results.

Append to the currently selected table The ASCII file is appended directly to the end of the current table. The ASCII file's structure must match the current or the ASCII file's records will not be imported correctly.

Copy another table's structure Use another table (that you will select) as a starting point for the ASCII file's structure. The Interactive Importer is then displayed for you to fine-tune the results.

Once the ASCII file's structure has been specified and/or reviewed (using one of the above methods), you are prompted for an output table name. The file is then imported.

Import from ASCII: Interactive Import

Import from ASCII (C:\dt\Projects\nonprofit.csv)

ASCII File Information

ASCII File Type: **Delimited** Delimiter: , (44) Record Size: **100** Fields: **9**

Currently Selected Field

Name: **FIRST** Type: **Fixed Character** Size: **10** Offset: **0**

FIRST	LAST	STREET	CITY	STA...	ZIP
FIRST	LAST	STREET	CITY	STATE	ZIP
Ronald	Robinson	306 Big Woods Dr.	Brooklyn	NC	16907-07
Phil	Scott	3391 Mockingbird Ln.	El Paso	MA	23678-98
Spencer	Lewis	5700 Mockingbird Ln.	Scarsdale	IN	09276-96
Beverly	Stewart	8624 Ketch Dr.	Hackensack	IL	54903-69
Pierre	Findley	5533 Pinto Ct.	Lowell	KY	50507-83
Emily	Harper	2672 The Greens	Urbana	OK	34781-47
Bo	Hunt	5017 Hope Rd.	Panama City	WA	47652-16
Ellis	Francois	7039 Stones Throw Dr.	Moultrie	MD	65834-14
Mercedes	Yoder	5063 Hunter Dr.	Spokane	CT	63278-78
Shirley	Hall	4219 Wisteria Lane	Los Angeles	OR	47810-56
Maxine	Brennan	5955 Ivy Court	Amarillo	AK	12729-49
Mickola	Harris	640 Cutter Ct.	Fort Dix	WA	70903-63
Mercedes	Sabo	9166 Woodbine Pl.	Staten Island	AR	92585-01

Exclude Field Clear Fields Get field names from first record.

File Info... Field Info... OK Cancel ? Help

Split Field (SDF and Flat File formats only) You can split the highlighted field into two smaller fields by clicking this button. The field will be split in half and then you can adjust the fields' sizes by dragging the dividing line to the left or right. Note that this is a positional split, not an intelligent split. You cannot use Split Field to break up a name field ("Mr. John Smith") into its parts. For this type of work, you should use the Name Splitter in Personator or MatchUp.

Join Field (SDF and Flat File formats only) You can join two adjacent fields into a single larger field by clicking this button. The highlighted field and its neighbor to the left will be joined.

Exclude Field You can intentionally exclude a field during importing by clicking this button. Of course, this field won't be removed from the original ASCII file, just the imported database. Excluded fields are shown in gray.

Clear Fields You can wipe out all of the fields (so you can start all over) by clicking this

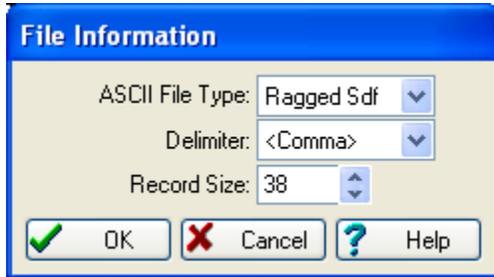
button.

Get field names from first record If the ASCII file's first record contains field names (as opposed to data), then check this box to use that data as field names (you can still change them to whatever you like). During the import, the first record will not be imported as data.

File Info Click this button if you need to change the ASCII file's type. Occasionally, the program will incorrectly guess the file's type and/or record size. Changing an ASCII file's type will cause the importer to reassign fields, so any changes made to field names, positions, etc., will be lost.

Field Info Click this button or double-click on the field's heading to change the information about the highlighted field. You can change the field's name, type, size and decimal places.

Interactive Import: File Information



ASCII File Type The type of ASCII file being imported:

- **SDF** Often called "fixed field" or "standard data format". Fields are of fixed length (padded with spaces) and each record is delimited with a carriage return and a line feed.
- **Ragged SDF** A spin-off of the SDF format, usually made by custom or in-house programs. The difference is that all trailing blanks at the end of each record are trimmed off.
- **Flat File** A common mainframe format. Fields are of fixed length, just like SDF, but there are no record delimiters of any kind.
- **Delimited** A common ASCII format. Fields are of varied length and delimited with a single character, usually a comma or a tab. Sometimes (particularly with character fields), fields are surrounded with double quotes. Each record ends with a carriage return and a line feed.

Delimiter (Delimited format only) The character that separates fields. Everybody's favorites, <Comma> and <Tab>, are listed first.

Record Size (SDF and Flat File formats only) The number of bytes used to store each record. Remember to count the carriage return and/or line feed in this number.

Interactive Import: Field Information



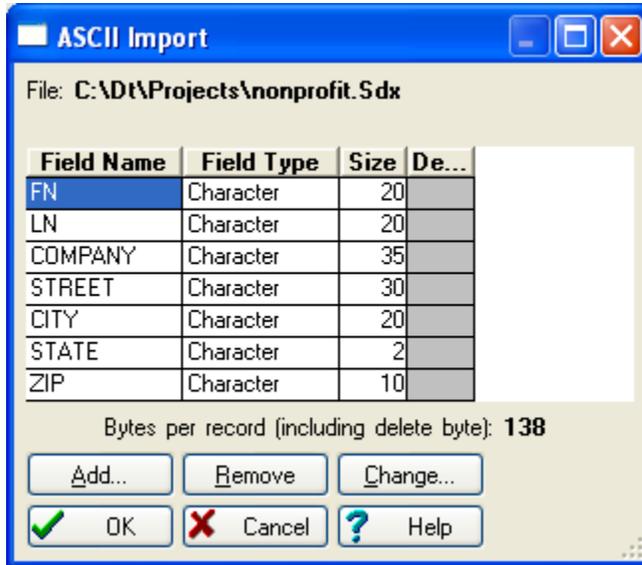
Field Name The field name must be between 1 and 10 characters. The first character must be alphabetic, but the others may be letters, numbers, or the underscore (_). Embedded spaces are not allowed.

Field Type There are seven options: Fixed Character, Variable Character, Integer, Float, Decimal, Logical, and DateTime. The program will automatically change the size of some of these types.

Field Size Some fields have fixed sizes, so you cannot edit these sizes. Character field sizes must be between 1 and 254.

Decimals (Numeric fields only) Decimal places must be between 0 and the field's size minus 2 (i.e., a numeric field of size 10 can have between 0 and 8 decimal places).

Import from ASCII: Manually Enter



Add (or CTRL+INSERT) Add a new field to the bottom of the list.

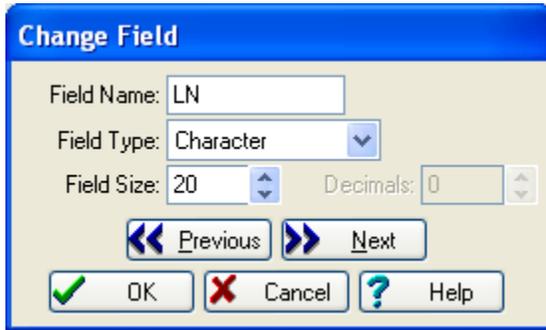
Remove (or CTRL+DELETE) Delete the currently highlighted field.

Change Modify the currently highlighted field. You can also modify fields by double-clicking the Field Name, Type, Size or Decimals entry that you want to change.

Moving Fields:

To move a field, just click and drag it to the desired location. Alternately, you can press CTRL+UP or CTRL+DOWN to accomplish this.

Adding Fields:



Field Name Field names must be between 1 and 10 characters. The first character must be alphabetic, but the others may be letters, numbers, or the underline (_). Embedded spaces are not allowed.

Field Type There are seven options: Fixed Character, Variable Character, Integer, Float, Decimal, Logical, and DateTime. The program will automatically change the size of some of these types.

Field Size Date fields are automatically sized to 8 and Logical fields to 1. Character field sizes must be between 1 and 254. Numeric field sizes must be between 1 and 20. Remember that if you shorten a field, you will lose any data that exceeds the new field width.

Decimals (Numeric fields only) Decimal places must be between 0 and the field's size minus 2 (i.e., a numeric field of size 10 can have between 0 and 8 decimal places).

7.3.2 Export to ASCII

Export to ASCII

Export Fields:

Source Table:

NAME	Fixed Character	30
STREET	Fixed Character	30
CITYSTZIP	Fixed Character	30
LASTCALL	DateTime	
AGE	Decimal	2.0
SALARY	Decimal	6.0

Add >>
<< Remove
All >>
<< None

Output File:

Export Options:

File Type:
 Delimiter:

Add first record with field names.
 Macintosh format (CRs, no LFs).
 Export records marked for deletion.
 Strip embedded CRs and LFs.

Delimit data with double quotes:
 Always.
 Only character fields.
 Only when there is a comma.
 Never.

OK Cancel Help

Source Table This lists each field in the table. By default, all fields will be exported. If you wish to export just a few fields, first click **None**, then click the field(s) to export and click **Add**.

Output File This lists each field to be exported. To remove field(s) from the list, click the field(s) to remove and click **Remove**. Note that the output field order of the **Output File** will always follow the same order as the **Source Table**. The only way to change the order is to physically modify the table's structure before exporting.

File Type The type of ASCII file to be created:

- **SDF** Often called "fixed field" or "standard data format". Fields are of fixed length (padded with spaces) and each record is delimited with a carriage return and a line feed.
- **Flat File** A common mainframe format. Fields are of fixed length, just like SDF, but there are no record delimiters of any kind.
- **Delimited** A common ASCII format. Fields are of varied length and delimited with a single character, usually a comma or a tab.

Delimiter (Delimited format only) The character that separates fields. Everybody's favorites, <Comma> and <Tab>, are listed first.

Add first record with field names Check this box to create an initial record that contains the field names (your ASCII file will be one record longer).

Macintosh format (CRs, no LFs) (SDF and Delimited formats only) Check this box to terminate each record with only a carriage return. Some Apple applications require this format.

Export records marked for deletion (dBASE tables only) Check this box to export records, regardless of whether or not the record's been marked for deletion. Deletion marks don't exist in ASCII files, so all records will appear as 'normal' in the ASCII file.

Strip embedded CRs and LFs Sometimes you will come across tables where field(s) have carriage returns (CRs) and/or line feeds (LFs) embedded (i.e., the characters aren't being used to delimit records - Excel seems to be the most common source of this problem.). These embedded CR/LFs can cause big problems with some database applications, so we offer the option of stripping them here.

Delimit data with double quotes There are several different standards as to how fields should be delimited:

- **Always** Every field gets delimited with double quotes.
- **Only Character Fields** Only character fields get double quotes - this is the most popular standard.
- **Only when there is a comma** The field is delimited with double quotes if a comma is embedded in the field.
- **Never** No double quote delimiters ever.

7.4 Matchcode Editor

Very Important Note:

A common source of confusion is the relationship between matchcode components and field names. Note that nowhere in the Matchcode Editor do you specify an actual table or field name. You only supply component field *types*. So how does the program know to use a field called "NAME" as the source for first name? You specify it on the Matchcode Mapping setup tab. This may seem like an extra step, but it gives you great flexibility. First, you create matchcodes that work for all of your databases, regardless of their structure. Then you "map" the fields in the Matchcode Mapping tab of your Merge/Purge or Purge setup.

The Matchcode Editor lets you maintain your personal matchcode database. Once you've used MatchUp for a while, you will probably have developed custom matchcodes that work well with your data. When you've developed the "Killer Matchcode", let us know. We're always looking for hot new prospects.

Matchcode Editor

Matchcodes

Address
Address, Last Name
Address, Last Name, First Name
Address, Last Name, Gender
Address, Company

Add...
Remove
Copy...
Rename...

Data Type	S...	Start	Fuzzy	Short/Empty	Swap	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
Zip5	5	Left	Exact	Both Fields	None	X	X															
First Name	10	Left	Exact	Both Fields	None	X	X															
Last Name	10	Left	Exact	Both Fields	None	X	X															
Street Number	5	Left	Exact	Both Fields	None	X																
Street Name	10	Left	Exact	None	None	X																
PO Box	10	Left	Exact	None	None		X															

Add... Remove
Created: 12/14/04 Modified: 12/14/04 I don't understand these X's!!!

Matchcode Component:
Data Type: First Name
Size: 10 Label:
 Maximum number of words:

Start:
 Left
 Right
 Position:
 Word:

Trim:
 Left
 Right

Custom File:

Matching Strategy:
 Exact Match
 Phonetex
 SoundEx
 Containment
 Frequency
 Fast Near:
 Accurate Near:
 Frequency Near:

Vowels Only
 Consonants Only
 Alphas Only
 Numerics Only
Tight Loose

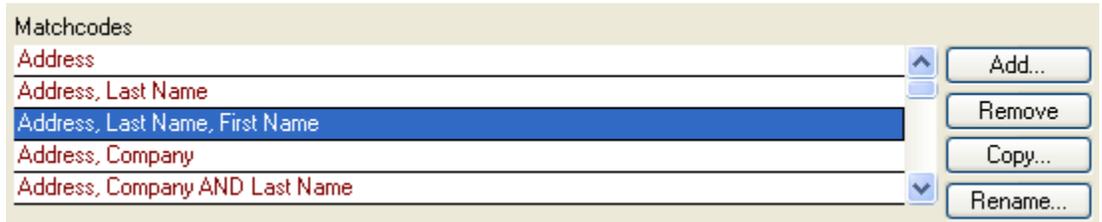
Short/Empty Settings:
 Match if both fields are blank
 Match if one field is blank
 Match initial to full field

Combinations:
 1 5 9 13
 2 6 10 14
 3 7 11 15
 4 8 12 16
 Advanced...

Swap Match Pairs:
Pair A Pair D Pair G
Pair B Pair E Pair H
Pair C Pair F

OK Cancel Help Print... Optimize Import from Double Take 2...

7.4.1 Matchcode List



This is a list of the matchcodes in your matchcode database. The middle part of the screen will list each component in the highlighted matchcode.

Our "canned" matchcodes are shown in red and are marked as "Read-only". This was done so that our technical support staff can rely on a consistent benchmark when troubleshooting users' problems. You can make your own matchcodes read-only by highlighting the matchcode and pressing CTRL+R to toggle the read-only status. This can come in handy in maintaining control over operating procedures.

The order of the matchcodes can be changed by dragging and dropping matchcodes. Note that the order of this table has no impact on any processing, it's just for your benefit.

Add Creates a new empty matchcode. If there is a matchcode that is close to what you need, it is usually easier to highlight that matchcode and click **Copy**.

Remove Delete the highlighted matchcode.

Copy Clone the highlighted matched with a new name that your specify. Now you can edit this new matchcode to your heart's delight.

Rename Change the name of the highlighted matchcode. You can also rename a matchcode by double-clicking it.

7.4.2 Matchcode Components

Data Type	Size	Start	Fuzzy	Short/Em...	Swap	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Zip5	5	Left	Exact	Both Fields	None	X	X														
First Name	10	Left	Exact	Both Fields	None	X	X														
Last Name	10	Left	Exact	Both Fields	None	X	X														
Street Number	5	Left	Exact	Both Fields	None	X															
Street Name	10	Left	Exact	None	None	X															
PO Box	10	Left	Exact	None	None		X														

Created: **08/13/03** Modified: **03/12/04**

This table lists the components in the currently selected matchcode. As you move from component to component, the properties in the lower left corner of your screen, will change to reflect the currently selected component.

Add Create a new component to the list. A dialog box pops up prompting you for this new component's properties. New components are always added to the bottom of the list; you can change a component's position by dragging and dropping it in the list.

Remove Delete the highlighted component from the matchcode.

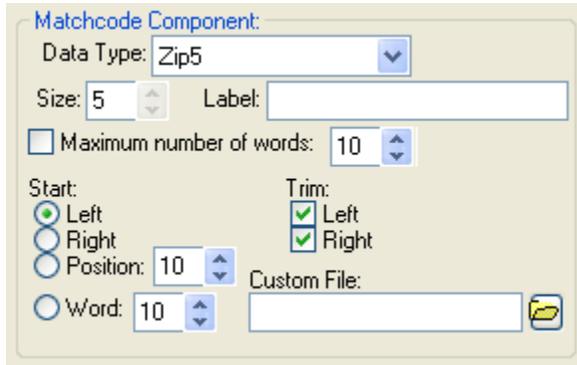
I don't understand these X's Well, you've come to the right place! The Matchcode Combinations topic is probably a good place to start.

Component Order:

The order of the components can be changed by dragging and dropping them or CTRL+UP or CTRL+DOWN. The order of this table can have some impact on processing in two ways:

- The first component has special restrictions that the others don't (see below).
- The order of components can effect processing speed. See Optimizing Matchcodes for more information.

7.4.3 Component



The screenshot shows a configuration window titled "Matchcode Component:". It contains several settings:

- Data Type:** A dropdown menu set to "Zip5".
- Size:** A spinner box set to "5".
- Label:** An empty text input field.
- Maximum number of words:** A checkbox is unchecked, and a spinner box is set to "10".
- Start:** Three radio buttons: "Left" (selected), "Right", and "Position".
- Trim:** Two checked checkboxes: "Left" and "Right".
- Position:** A spinner box set to "10".
- Word:** A spinner box set to "10".
- Custom File:** A text input field with a folder icon to its right.

You can change properties by double-clicking or right-clicking a property or by highlighting the component and changing the properties below.

Data Type The type of data represented by this component.

Size The number of characters that will be used from this component. See Component Properties for more information.

Label A label that can be attached to this component. MatchUp does not itself use this label, but it can be helpful in remembering for what a particular General component was intended. If a label is present, it will be displayed on the Matchcode Mapping screen.

Maximum number of words The maximum number of words that will be used. See Component Properties for more information.

Start Where character extraction should begin:

- **Left** From the left end of the data (most common choice).
- **Right** From the right end of the data.
- **Position** From the specified character position in the data.
- **Word** From the specified word in the data.

Trim Whether leading and/or trailing white space should be stripped from the data. See Component Properties for more information.

Custom When processing custom data types, a custom lookup table is specified here. See Matchcode Components for more information.

First Component Requirements:

For accurate clustering, MatchUp places some special restrictions on the first component in a given matchcode (clustering is discussed in Concepts).

The following must be true for a matchcode's first component:

- It must be used in every condition.
- It cannot use certain Matching Strategies: Containment, Frequency, Fast Near, Frequency Near or Accurate Near (other types are okay, though).
- It cannot use "Match if one field is blank".
- It cannot use "Match initial to full field".
- It cannot use Swap Matching.

MatchUp will display all components that meet these conditions with a pale green background - until a component not meeting these conditions is encountered. Having additional components that meet these criteria will usually make the matchcode process faster (see [Optimizing Matchcodes](#) for more information).

7.4.4 Matching Strategy

Matching Strategy:

<input checked="" type="radio"/> Exact Match	<input type="radio"/> Vowels Only
<input type="radio"/> Phonetex	<input type="radio"/> Consonants Only
<input type="radio"/> SoundEx	<input type="radio"/> Alphas Only
<input type="radio"/> Containment	<input type="radio"/> Numerics Only
<input type="radio"/> Frequency	<i>Tight</i> <i>Loose</i>
<input type="radio"/> Fast Near:	_____
<input type="radio"/> Accurate Near:	_____
<input type="radio"/> Frequency Near:	_____

Matching Strategies allow for matching of non-exact components. These options are mutually exclusive (i.e., you can only select one at a time). The following settings are available:

- **Exact Match** Data must be identical to be seen as a match.
- **Soundex** An auditory matching algorithm originally developed by the INS and later adopted by the USPS.
- **Phonetex** An auditory matching algorithm.
- **Containment** Match when one record's component is contained in another record's.
- **Frequency** Match the characters in one record's component to the characters in another, without any regard to the sequence.
- **Fast Near** A typographical matching algorithm. Faster than **Accurate Near**.
- **Accurate Near** A typographical matching algorithm. More accurate than **Fast Near**.
- **Frequency Near** A typographical matching algorithm. A combination of **Frequency** and **Accurate Near**.
- **Vowels Only** Only vowels will be compared. Consonants will be ignored.
- **Consonants Only** Only consonants will be compared. Vowels will be ignored.
- **Alphas Only** Only alphabetic characters will be compared.
- **Numerics Only** Only numeric characters will be compared. Decimals and signs are considered numeric.

Matching Strategy: Soundex

Soundex is an algorithm developed in 1917 by the Immigration Service and later adopted by the USPS. It is identical to the Soundex() function found in any of the dBASE languages. Its basic operations are to (1) keep the first letter of the string, (2) remove any vowels, (3) ignore all H's, W's and Y's, (4) change all double letters to singles, and (5) substitute numbers for the next 3 consonants using the following table (if there are no more consonants, 0's are used):

Character(s) Code

FPV	1
CSKQXZ	2
DT	3
L	4
MN	5
R	6
B	7
GJ	8

For example, "Peoplesmith" would be Soundexed to "P142".

Although Soundex has been in use over 80 years and has a wide following, we cannot understand how "S", "K", and "Q" can be grouped together as the same sound. Consequently, we greatly prefer Phonetex.

Matching Strategy: Phonetex

Phonetex (pronounced "Fo-net-icks") is similar to Soundex, but there are some key differences in its operation: (1) scan the string for certain letter combinations, and substitute alternate representations. For example, hard C's are converted to K's and soft C's are converted to S's. Leading GN's, PN's, and MN's are converted to N's, and so on. (2) certain phonemes like "PH" and "IGHT" are recognized. (3) Vowels are then removed from the string, and then numbers are substituted for the remaining consonants using the following table:

Character(s)	Code
X	1
KQ	2
L	3
R	4
MN	5
CSZ	6
FVW	7
BDPT	8
GJ	9

For example, "Peoplesmith" would be Phonetexed to "P83658".

Note that Phonetex doesn't have the 4 character restriction that Soundex has. Additionally, consonant combinations are specially treated, and letter groupings are more consistent with English speech patterns.

Matching Strategy: Containment

A match is found when one record's component is contained in another record. For example, "no" is contained in "innovation".

Matching Strategy: Frequency

The characters in one record's component are matched against the characters in another without any regard to the sequence. For example "abcdef" would match "badcfe".

Matching Strategy: Fast Near

A typographical matching algorithm. It works best in matching words that don't match because of a few typographical errors. Exactly how many errors is specified on a scale from 1 to 4 (1 being the tightest). The Fast Near algorithm is a speedy approximation of the Accurate Near algorithm. The tradeoff for speed is accuracy; sometimes Fast Near will find false matches or miss true matches.

Matching Strategy: Accurate Near

A typographical matching algorithm. The Accurate Near algorithm produces more accurate results than Fast Near, but at the cost of speed.

Matching Strategy: Frequency Near

The characters in one record's component are matched against the characters in another without any regard to the sequence. For example "abcdeg" would match "badcfe". You can specify the number of allowed errors with the slider.

7.4.5 Short/Empty Settings

Short/Empty Settings:

- Match if both fields are blank
- Match if one field is blank
- Match initial to full field

Match if both fields are blank If two records have the same empty component, that component will be counted as matching. So, two records with the first name missing will match. See Component Properties for more information.

Match if one field is blank Allows MatchUp to match missing data with the full data. For example, "Smith" matches "John Smith". However, two records with the same component missing will not match. See Component Properties for more information.

Match initial to full field Allows MatchUp to match abbreviated data with the full data. For example, "J Smith" matches "John Smith". See Component Properties for more information.

7.4.6 Combinations

Combinations:

<input checked="" type="checkbox"/>	1	<input type="checkbox"/>	5	<input type="checkbox"/>	9	<input type="checkbox"/>	13
<input type="checkbox"/>	2	<input type="checkbox"/>	6	<input type="checkbox"/>	10	<input type="checkbox"/>	14
<input checked="" type="checkbox"/>	3	<input type="checkbox"/>	7	<input type="checkbox"/>	11	<input type="checkbox"/>	15
<input type="checkbox"/>	4	<input type="checkbox"/>	8	<input type="checkbox"/>	12	<input type="checkbox"/>	16

Advanced...

The 1 through 16 check boxes define component combinations that should be considered a match. It is easier to visualize the effects of these boxes if you look at the Component Table part of the screen:

Data Type	S...	Start	Fuzzy	Short/Empty	Swap	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Zip5	5	Left	Exact	Both Fields	None	X	X	X	X												
Last Name	10	Left	Exact	Both Fields	None	X	X														
Company	10	Left	Exact	Both Fields	None			X	X												
Street Number	5	Left	Exact	Both Fields	None	X		X													
Street Name	10	Left	Exact	None	None	X		X													
PO Box	10	Left	Exact	None	None		X		X												

It is important to note that each **vertical** column of checkmarks designates one acceptable matchcode. For example, the illustration above shows a combination that is made up of 4 matchcodes:

1. Zip5, Last Name, Street #, and Street Name.
2. Zip5, Last Name, and PO Box
3. Zip5, Company, Street#, and Street Name.
4. Zip5, Company, and PO Box

Since Street Number is highlighted, the Combinations box displays check marks for the component combination Street Number appears in: 1 and 3.

One final comment: due to a requirement of Clustering, at least the first component must be used in every combination (which is why we provide the check marks for the first row).

Combinations: Advanced Combination Settings

Warning: Advanced Topic!

The Advanced Combination Settings allow you to specify whether or not a particular combination should be evaluated in every situation. In most cases, you will want to evaluate all combinations in all situations (the default).

This powerful option can be used in situations where an awkward workaround would otherwise be needed.

Advanced Combination Settings

Use Advanced Combination Settings

The **Advanced Combination Settings** allow the user to specify which matchcode combinations should be used in which situations.

This is a very advanced topic. Be sure that you understand it fully before using it on production data.

1: Z5+LN+S#+SN	Reset this combination
2: Z5+LN+POB	
3: Z5+Co+S#+SN	
4: Z5+Co+POB	Reset all combinations

Use in comparisons with records from Regular Files.
 Use in comparisons with records from Suppression Files.
 Use in comparisons with records from Intersection Files.

AND this is true: ...
 OR this is true: ...
 (no additional conditions).

Important: The above AND condition is evaluated whenever two records are compared. Should the AND condition be required for **both** records or **either** record?

Use this combination if either record satisfies the condition.
 Use this combination only if both records satisfy the condition.

Used Advanced Combination Settings Check if you need to set advanced combination settings for any combination in your matchcode.

Combination Select the combination that you would like to configure.

Reset this combination Reset the highlighted combination to its default (non-advanced) configuration.

Reset all combinations Reset all combinations to the default (non-advanced) configuration.

Use in comparisons with records from Regular Files This combination should be evaluated when dealing with regular records.

Use in comparisons with records from Suppression Files This combination should be evaluated when dealing with suppression records.

Use in comparisons with records from Intersection Files This combination should be evaluated when dealing with intersection records.

AND this is true If the above selections are satisfied and the specified expression is true. The expression is a dBASE expression, but instead of field names, component names should be used when needed (see below).

OR this is true If the above selections are satisfied or the specified expression is true. The expression is a dBASE expression, but instead of field names, component names should be used when needed (see below).

(no additional conditions) Only the above selection(s) needs to be satisfied.

When MatchUp evaluates the above AND or OR conditions, it performs this check on both records being compared. You must tell MatchUp if both records need to satisfy the condition or if only one needs to:

- **Use this combination if either record satisfies the condition**
- **Use this combination only if both records satisfy the condition**

dBASE Expressions:

The dBASE expressions used for the AND and OR conditions are nearly the same as the type that you would use for an Input Filter condition. The one difference is that these expressions can't use field names (because we don't know what tables you are planning on using with this matchcode). Instead, you can use the matchcode's component names. For component names with embedded spaces, the spaces should be removed (ie, the First Name component should be specified FirstName). If you have several components that are the same type, or you would like more specific names, assign the component(s) a Label in the Matchcode Editor, and refer to that label in the expression.

In some instances, you will want to refer to a field in your table that isn't in the matchcode.

You can accomplish this by adding a component to the matchcode (usually a General will do), but not assigning it any X's (so it will not be directly required by any combination). You can then refer to this field in your expression.

When you use a component in an expression, the following applies:

- It will always be a character data type.
- It will always be of whatever size you specified for the component.
- It will be padded with spaces so that it is the specified size (ie, you may need to `RTrim()` when doing string comparisons.

It will receive any early matching treatments which are described in Optimizing Matchcodes.

Combinations: Advanced Combination Example

Advanced Combinations may seem like such an abstract concept that you couldn't imagine why someone would ever use them.

Say you would like to do a mailing to potential MatchUp customers. The target would consist of Data Processing workers in large and small companies. Now, the logic may be this:

- For small companies (less than 20 employees), a loose matchcode (more matches) could be used, because chances are that if the mail piece landed in the hands of the inappropriate individual, it would be passed on
- For large companies (over 20 employees), a tighter matchcode (less matches) could be used, because information doesn't exchange hands so readily in Cubical City.

A matchcode like the following would be created:

Component	Label	Size	Fuzzy	Short/Empty	1	2	3	4
Zip5		5	Exact	None	X	X	X	X
Last Name		5	Exact	Both Fields	X	X		
Company		10	Exact	Both Fields			X	X
Street Number		5	Exact	Both Fields	X		X	
Street Name		10	Exact	None	X		X	
PO Box		10	Exact	None		X		X
General	Employees	5	Exact	None				

This sets up 4 matching conditions:

Large Companies:

1. Zip5 + Last Name + Street Number + Street Name
2. Zip5 + Last Name + PO Box

Small Companies:

3. Zip5 + Company + Street Number + Street Name
4. Zip5 + Company + PO Box

However, as it stands right now, all conditions will be evaluated in all situations, large or small company. That's where the Advanced settings come in...

Advanced Combination Settings

Use Advanced Combination Settings

The **Advanced Combination Settings** allow the user to specify which matchcode combinations should be used in which situations.

This is a very advanced topic. Be sure that you understand it fully before using it on production data.

1: Z5+LN+S#+SN
2: Z5+LN+POB
3: Z5+Co+S#+SN
4: Z5+Co+POB

Reset this combination

Reset all combinations

- Use in comparisons with records from Regular Files.
- Use in comparisons with records from Suppression Files.
- Use in comparisons with records from Intersection Files.

- AND this is true:
- OR this is true:
- (no additional conditions).

Important: The above AND condition is evaluated whenever two records are compared. Should the AND condition be required for **both** records or **either** record?

- Use this combination if either record satisfies the condition.
- Use this combination only if both records satisfy the condition.

Here, we've set a rule for condition one that states that when Employees is greater than 20, Condition 1 should be used. Condition 2 gets the same rule. Conditions 3 and 4 get a different rule:

Advanced Combination Settings

Use Advanced Combination Settings

The **Advanced Combination Settings** allow the user to specify which matchcode combinations should be used in which situations.

This is a very advanced topic. Be sure that you understand it fully before using it on production data.

1: Z5+LN+S#+SN
2: Z5+LN+POB
3: Z5+Co+S#+SN
4: Z5+Co+POB

Reset this combination
Reset all combinations

Use in comparisons with records from Regular Files.
 Use in comparisons with records from Suppression Files.
 Use in comparisons with records from Intersection Files.

AND this is true: VAL(Employees) <= 20
 OR this is true:
 (no additional conditions).

Important: The above AND condition is evaluated whenever two records are compared. Should the AND condition be required for **both** records or **either** record?

Use this combination if either record satisfies the condition.
 Use this combination only if both records satisfy the condition.

Close Help

Which states the exact opposite of the rule for Conditions 1 and 2.

By now, you're probably wondering about this General field with the "Employees" label. Unlike most dBASE expressions, the Matchcode Editor cannot use field names when evaluating an expression. The reason is simple: there are no tables! But you can use the contents of components. So we throw a General component in to use for our evaluation. We give it a label ("Employees") so that we know what we're talking about in our expressions and also later when we're on the Matchcode Mapping tab. So when we use this matchcode in a setup, we will need to map an Employee Count field into the "Employee" component.

When you use a component in an expression, the following applies:

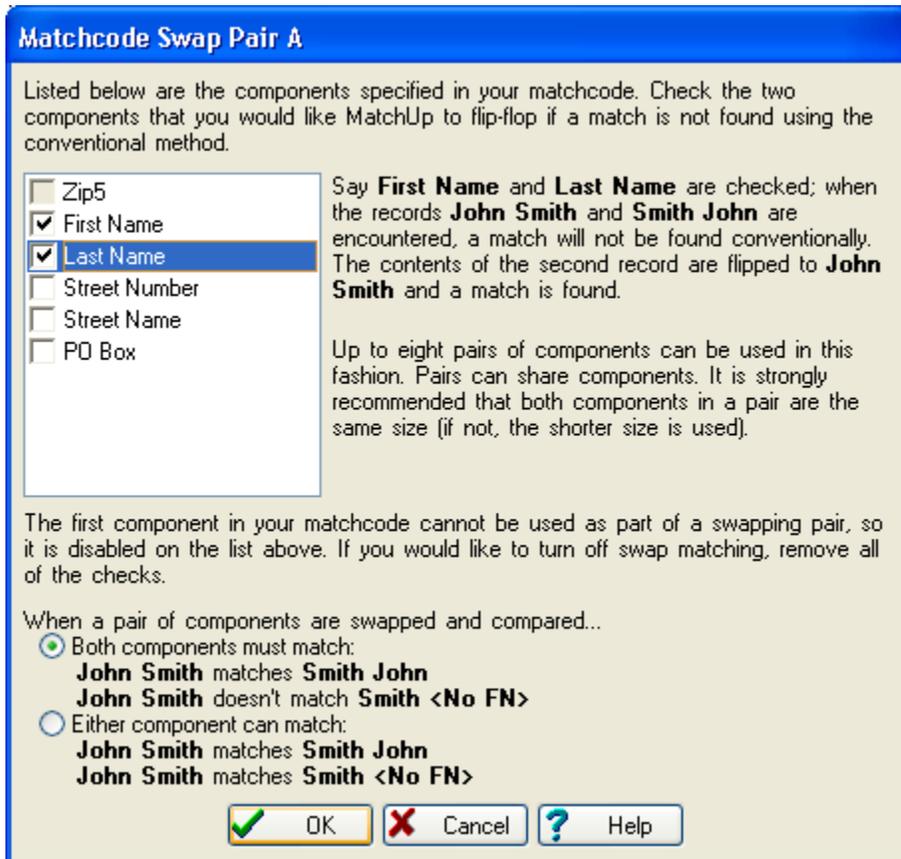
- It will always be a character data type.
- It will always be of whatever size you specified for the component.
- It will be padded with spaces so that it is always the specified size (ie, you will want to use `RTrim()` when doing string comparisons).

One other thing to note is that the General field isn't included in any combinations. The Matchcode Editor is okay with this and will not remove this field during any optimizations.

7.4.7 Swap Match Pairs



Swap Matching allows you to match "John Smith" with "Smith John". Essentially, you indicate in the Matchcode Editor two components whose contents can be swapped (in this case, First Name and Last Name). The components must be of the same size and should have the same set of matching options (i.e. don't Phonetex one and SoundEx the other). You can define up to eight pairs.



Both components must match When a pair of components is flipped, both components

must match in the reversed configuration. For example, "John Smith" would match "Smith John", but not "Smith <No FN>" nor "Smith Bob". We call this option a "full swap".

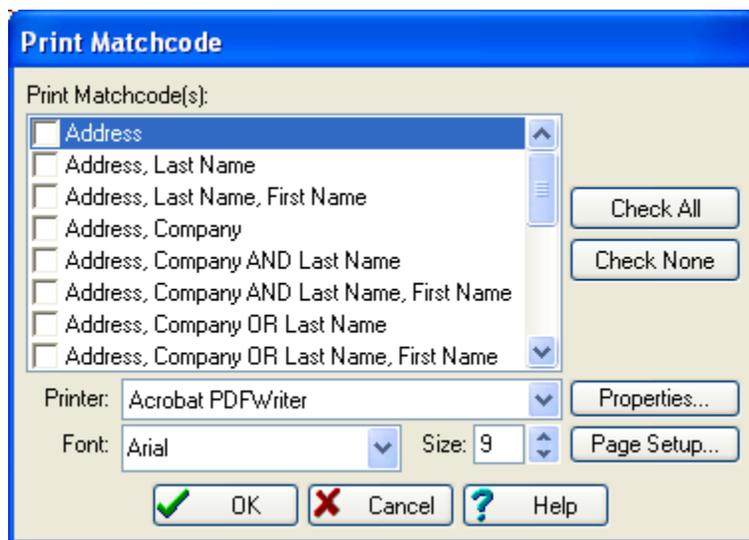
Either component can match When a pair of components is flipped, either component can match in the reversed configuration. For example, "John Smith" would match "Smith John", "Smith <No FN>", and "Smith Bob". This option is sometimes referred to as a "half swap".

Full Swap or Half Swap?

It depends on the situation. If you're matching two components that have the potential for being reversed (First and Last Name, for example), a full swap is probably what you want.

However, if you're matching several items appearing on a list, a half swap is the most likely option. For example, if you have a list that contains several household members (ie, FIRST1, FIRST2, FIRST3), you would like to catch matches where FIRST1 matches FIRST1, FIRST2 or FIRST3. In this situation, a Full Swap may not always work, as the ordering of names is inconsistent. Half Swap is the way to go.

7.4.8 Print Matchcode



You can print any or all of your matchcodes using this dialog box.

7.4.9 Optimize

The Optimize option tries to re-arrange your Matchcode's components in the most efficient order. It will also remove redundant combinations and arrange them in an optimal order.

In some (rare) cases, the existing component order may be critical to your process (for example, if you're using the Output Matchcode in another program). Sometimes the existing combination numbering may be critical (for example, if you're using the Output Matchcode in another program). In both of these cases, it's best to not use the Optimize option, or at least double-check the changes that MatchUp has made to your matchcode.

See [Optimizing Matchcodes](#) for more information on some of the more advanced optimizations that this option does and does not perform.

7.5 User Settings

If you invest three minutes now to set up MatchUp with your personal taste and style, it will pay off handsomely in the hours you will save in the future!

7.5.1 General

General: Interface

General: Interface

ENTER key:

- ENTER moves to next dialog prompt, CTRL+ENTER hits OK
- TAB moves to next dialog prompt, ENTER hits OK

Screen Saver:

- Don't suppress screen saver
- Suppress screen saver when processing

Tip of the Day:

- Always show tips at startup
- Show tip once per day
- Sometimes show tip
- Never show tip

Advanced: dBASE Emulation

If you don't have any idea what **SET EXACT** means or even what a dBASE is, leave this option unchecked.

If this option is checked, the string comparison

"BLACKWELL"="BLACK" returns **True**. Note that you can use the Exact Equals operator (==) when you temporarily need SET EXACT ON behavior.

SET EXACT OFF

Advanced: ADO/DAO Usage

Access and Excel files can be accessed using either DAO (older) or ADO (newer).

You may find that one technology performs better than the other in certain situations.

- Use ADO for Access & Excel tables
- Use DAO for Access & Excel tables.

ENTER key Control the behavior of the ENTER key in dialog boxes:

- **ENTER moves to the next dialog prompt, CTRL+ENTER hits OK** When navigating a dialog box, hitting ENTER will proceed to the next prompt. Technically, ENTER is supposed to hit the **OK** button, but many people find this an unnatural behavior, so we give you a more conventional interpretation.
- **TAB moves to the next dialog prompt, ENTER hits OK** When navigating a dialog box, hitting ENTER will immediately hit the **OK** button. This is standard Windows behavior.

Screen Saver Control the screen saver's behavior:

- **Don't suppress screen saver** The Windows screen saver's behavior will not be changed.
- **Suppress screen saver when processing** When processing, MatchUp will temporarily disable the screen saver. This ensures that all processing power is given to the program rather than rendering flying toasters.

Tip of the Day Control how often you want to see the Tip of the Day dialog:

- **Always show tips at startup** You'll get them every time.
- **Show tip once per day** Only the first time, so pay attention.
- **Sometimes show tip** Only when you least expect it.
- **Never show tip** Don't like them?

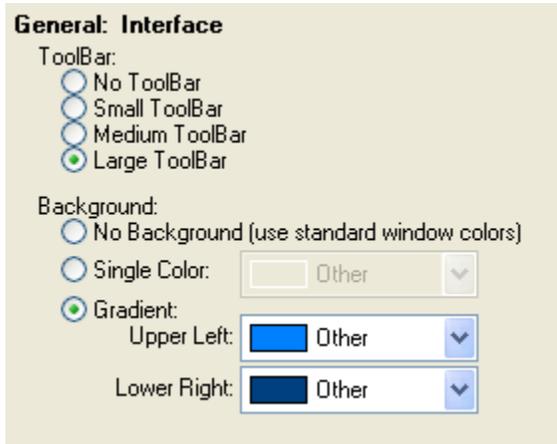
Advanced: dBASE Emulation:

SET EXACT OFF This advanced option allows you to control how the = operator behaves in dBASE expressions. If this is checked, "Shortcuted" string comparisons are performed. For example, "BLACKWELL"="BLACK" will return **true** (but "Black" will not equal "Blackwell"). This behavior is the opposite of SET EXACT ON. See = Operator, and == Operator in the help file for more information.

Advanced: ADO/DAO Usage

This advanced option allows you to select which Microsoft technology to use when accessing Access and Excel files. You may find that one technology performs better than the other in certain situations. Also, see the Frequently Asked Questions if you are having difficulties with Access or Excel files.

General: Main Window



ToolBar Change the toolbar's appearance:

- **No Toolbar** Hide the browser's toolbar.
- **Small Toolbar** Display the small toolbar.
- **Medium Toolbar** Display the medium toolbar.
- **Large Toolbar** Display the large toolbar.

Background Change the main window's background:

- **No Background (use standard window colors)** If your computer was made during the cold war, this is for you.
- **Single Color** Nice and plain, nothing to see here.
- **Gradient** Good Lord, the colors!

General: User File Locations

General: User File Locations:

Default location of setup (.dt) files:



Default location of data files:



Default location of setup (.dt) files: Lets MatchUp know where to start looking for your .dt setup files.

Default location of data files: Lets MatchUp know where to look for your data files.

The above two options can make work a bit easier for you. If you always store your setup files and/or data files in the same location, it is worth your while to specify the folder(s) above. If specified, **Add Table** and **File | Open Setup** will always start in the specified location. If no folder is specified, these commands will start in the last file location (which is very useful for some people).

General: Auto-Update

General: Auto-Update

At specific intervals, MatchUp can check to see whether or not there's a code update on our internet web site.

- Don't ever check for updates.
- Periodically check for updates, but ask first.
- Periodically check for updates without asking.

Check for updates every: days

Last update was done on **04/27/2007**.

Auto-Update Select how often MatchUp should check for updates:

- **Don't ever check for updates** You can still download updates from our web site and install them manually.
- **Periodically check for updates, but ask first** Sometimes you really need to get the job done without waiting for an update.
- **Periodically check for updates without asking** Updates are pretty fast over broadband connections and don't take more than a minute.

7.5.2 Setup

Setup: Field Naming

Setup: Field Naming
These descriptions are used to determine what kind of data is contained in a field.

Edit descriptions for: Last Name

Add description: Add

Existing descriptions: LN
LastName
Last
LName
L_N*

Remove
Move Up
Move Down

This screen lets you customize how MatchUp determines what kind of data is contained in a field. By entering your field naming habits, you can set up a new job significantly faster. These descriptions are used in Merge, Purge, Merge/Purge, and Update setups. Similar descriptions are used for CASS setups.

Edit Descriptions for Select the type of field you want to add naming description(s) to. For example, if you always name Prefix fields "MR_MS", you'll want to select "Prefix" in this drop down box.

Add Description Enter a field name for this type of field. Wildcards are allowed. For example, if you always name Prefix fields "MR_MS", you'll want to put "MR*" or "MR_MS" in this box. When you have typed the information, click **Add** to put it on the **Existing Descriptions** list. If your database structures are fairly consistent, it is usually better to not use wildcards and specify the entire field name. For example, if you specified "CO*" to catch a field you usually call "COMPANY", MatchUp can mistake "CONTACT" as a Company field, as it fits the wildcard description.

Existing Descriptions Lists field names for the currently selected type of field.

Remove Removes the highlighted description from the list.

Move Up/Move Down The position of a field in the description list is important, as MatchUp evaluates the **Description List** from the top down. Therefore, you can improve how well MatchUp determines field names by putting the most common field descriptions at the top of the list.

Setup: Field Mapping

Setup: Automatic Field Mapping

- Automatically map fields on "Matchcode Mapping" tab
- Automatically map fields on "Output Field Mapping" tab
- Automatically map fields on "Update" tab

Automatically map fields on "Matchcode Mapping" tab Check this box if you want MatchUp to automatically determine field names on the Matchcode Mapping setup tab.

Automatically map fields on "Output Field Mapping" tab Check this box if you want MatchUp to automatically determine field names on the Output Field Mapping setup tab.

Automatically map fields on "Update" tab Check this box if you want MatchUp to automatically determine field names on the Update setup tab.

Setup: Default Output Table

Setup: Default Output Table

Default Table Type: ▼

When creating a Merge or Merge/Purge setup, the default output structure...

...is empty

...is inherited from the first added input table

...is the following:

Field Name	Field Type	Size	De...	Data Type
PRE	Fixed Character	6		Prefix
FN	Fixed Character	20		First Name
MI	Fixed Character	2		Middle Name
LN	Fixed Character	20		Last Name
SUFF	Fixed Character	10		General
ADDR1	Fixed Character	30		Address
ADDR2	Fixed Character	30		Address
ADDR3	Fixed Character	30		Address
CITY	Fixed Character	20		City
ST	Fixed Character	2		State

Default Table Type When you create a new Merge/Purge or Merge setup, certain restrictions are made on your Output Fields (naming, sizes, etc.). These restrictions are based on what database type you choose for your output tables. The catch is that MatchUp usually has no idea what database type you have in mind (as the Output Tables often have yet to be specified). In cases like these MatchUp will usually make a guess based on the database type(s) of your input tables. However, if you've specified a Default Table Type here, it will be used instead.

When creating a Merge or Merge/Purge setup, the default output structure:

- **is empty** No output structure will be entered. You will have to enter a structure for the output table.
- **is inherited from the first added input table** Inherit the structure from the first source file given. When you do a Merge or a Merge/Purge, the default output field structure will be the structure of the first input file you select.
- **is the following** The default output structure is specified in the table below. Use the **Add**, **Remove** and **Change** buttons to add, remove and change fields. The **Inherit from Table** button allows you to acquire a structure from an existing database that you specify (you can modify the structure afterwards if you need to).

Remember that no matter what output structure you decide to go with, you will have the option to change the output table structure while you are creating the setup.

Setup: CASS

CASS: Field Naming

This screen lets you customize how MatchUp determines what kind of data is contained in a field. By entering your field naming habits, you can set up a new job significantly faster. These descriptions are used in CASS setups. Similar descriptions are used for Merge, Purge, Merge/Purge and Update Setups.

Edit Descriptions for Select the type of field you want to add naming description(s) to. For example, if you always name Prefix fields "MR_MS", you'll want to select "Prefix" in this drop down box.

Add Description Enter a field name for this type of field. Wildcards are allowed. For example, if you always name Prefix fields "MR_MS", you'll want to put "MR*" or "MR_MS" in this box. When you have typed the information, click **Add** to put it on the **Existing Descriptions** list. If your database structures are fairly consistent, it is usually better to not use wildcards and specify the entire field name. For example, if you specified "CO*" to catch a field you usually call "COMPANY", MatchUp can mistake "CONTACT" as a Company field, as it fits the wildcard description.

Existing Descriptions Lists field names for the currently selected type of field.

Remove Removes the highlighted description from the list.

Move Up/Move Down The position of a field in the description list is important, as MatchUp evaluates the **Description List** from the top down. Therefore, you can improve how well MatchUp determines field names by putting the most common field descriptions at the top of the list.

CASS: Output fields

Setup: CASS: Output Fields

These field names and sizes will be used as default entries for output fields when creating new CASS setups.

Company:	CASSCompany	30	▲▼
Address Line 1:	CASSAdd1	30	▲▼
Address Line 2:	CASSAdd2	30	▲▼
Suite:	CASSAdd3	30	▲▼
City:	CASSCity	20	▲▼
State:	CASSState	2	▲▼
Zip:	CASSZip	10	▲▼
Plus 4:	CASSZip	10	▲▼
Urbanization:		0	▲▼
Carrier Route:	CASSCRRT	4	▲▼
Delivery Point Code:	CASSDP	3	▲▼
DP Check Digit:	CASSDP	3	▲▼
Error Code:	CASSError		
Status Code:	CASSStatus		
LACS ^{Link} Status:	CASSLACS		

When a CASS setup is created, these field names (and sizes if the fields don't exist in the table) will be automatically specified. If you have developed a field naming convention, entering it here will save you from ever entering it again in CASS setups.

CASS: Options

Setup: CASS: Options

These options will be used as default entries when creating new CASS setups.

If there's a coding error...

- Clear the Input Plus 4 field
- Clear the Input Carrier Route field

If joining City/State/Zip...

- Delimit City & State with a space.
- Delimit City & State with a comma.
- Delimit Zip & Plus 4 with a dash.
- Delimit Zip & Plus 4 with a space.
- Join Zip & Plus 4 with no delimiters.
- Perform Delivery Point Validation (DPV)
- Perform LACS^{Link} Processing
- Use CASSMate enhanced processing
- Process via Zip Index

Form 3553 Information:

Processor Name:	<input type="text" value="Melissa Data Corp."/>		
Mailing Company:	<input type="text" value="Melissa Data Corp."/>		
Address:	<input type="text" value="22382 Avenida Empresa"/>		
City:	<input type="text" value="Rancho Santa Margarita"/>		
State:	<input type="text" value="CA"/>	Zip:	<input type="text" value="92688-2112"/>

The following settings will automatically be selected whenever you start a new CASS setup.

If there's a coding error What should MatchUp do if it comes across a coding error (you can choose one or both):

- **Clear the Input Plus 4 field**
- **Clear the Input Carrier Route field**

City/State Delimiter If you elect to put city, state, and zip data into a single field, how would you like them delimited?

- **Delimit City & State with a space**
- **Delimit City & State with a comma**

Zip/Plus 4 Delimiter Same deal, but separating the Zip and Plus 4:

- **Delimit Zip & Plus 4 with a dash**
- **Delimit Zip & Plus 4 with a space**
- **Delimit Zip & Plus 4 with no delimiters**

Perform Delivery Point Validation (DPV) Check if you want to verify each address's delivery point during CASS verification.

Use CASSmate enhanced processing Whether or not to use the enhanced power of CASSmate to get more CASS matches. Enhanced CASSmate processing is only used when normal CASSing attempts have yielded no results.

Process via Zip Index CASSmate generally processes faster in Zip order. But, often your table is not in Zip order (and you don't want it to be). With this option checked, MatchUp will create a temporary index that will cause your table to be CASSed in Zip order (but it's physical order will remain untouched). This option is disabled for all table types except dBASE (the speed gain is countered by random access slowdowns).

Form 3553 information This is the general information necessary to fill out the USPS Form 3553 for the CASS certification. Be sure to fill in this information or your Form 3553 will not print out complete.

Setup: Warnings

Setup: Warnings
Before processing, MatchUp checks for mistakes that you may have made in your setup. You can disable them here:

- No Rank was specified for <Table>.
- No Rank Field was specified for <Table>.
- <Field> will be added to <Table>.
- <Field> was also specified as <Field> for <Table>.
- No field has been mapped into <Component> from <T
- <Table> will be overwritten.
- <Table> is ASCII or Read-Only.
- Missing CASS 3553 Information.
- No Output, Duplicate, Suppression, or Non-Intersecti

Warning messages that are checked will be reported before processing. Messages that aren't checked will not.

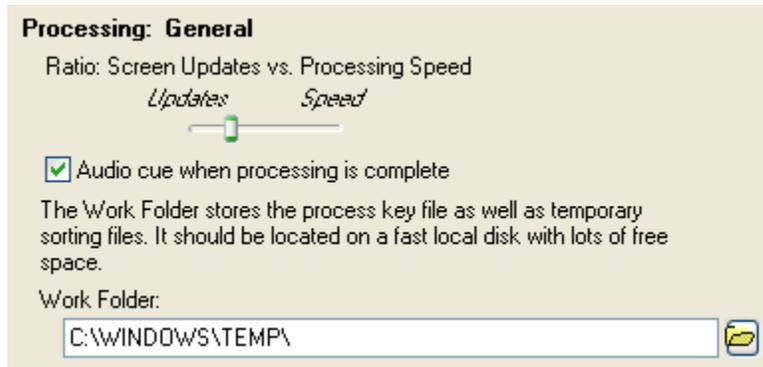
Before processing, MatchUp checks for possible error and warning situations. You can suppress any or all of the warnings by unchecking it here.

Note: Only *warnings* can be shut off. Errors (as opposed to warnings) are not shown here because you can not turn them off.

See also Setup Inspector.

7.5.3 Processing

Processing: General



Ratio: Screen Updates vs. Processing Speed:

Adjust the slider to your preference. When you set for more frequent screen updates, it appears that records are being processed faster, but it's an optical illusion. In reality, fewer records are being processed (because more time is being wasted updating the screen). Test it yourself!

Because Windows is a multitasking environment, it is quite possible that you may be running another Windows program while processing (which is completely legal). You will notice that faster slider settings will slow down other windows programs running, while slower slider settings will speed them up. You should choose the slider setting that works best for what's running - some programs demand more processor time than others.

The processing screen has a pause button, so if you need to have your computer's "undivided attention", you can pause MatchUp to do your things and then resume processing.

Work Folder: The work folder stores the Purge and Merge/Purge Key and Index files. If you are often getting "Out of disk space" warnings, you should confirm that the work folder is a Drive/folder with lots of disk space. Also, in network situations, it is desirable to use a fast local drive instead of a network drive as speed is greatly degraded by network traffic.

Processing: Translation Table

Processing: Translation Table

The Translation Table is used to translate foreign characters into a standardized equivalent (for example, 'Ç' into 'C').

The character set that is used during data entry is known as a "Code Page". When working with a foreign list, you should ask the list provider which Code Page was used during data entry, and select it below.

MatchUp does not convert your data. Translation Tables are only used to determine if two records match.

Translation Table:

If you've created a Translation Table in Dirty Harry's Character Assassin, you can convert it here:

[Import a Translation Table from Dirty Harry](#)

Translation Table These tables are used to exchange foreign characters for an English equivalent when building matchcodes:

- **CP 1252 Windows** Default
- **CP 437 English** Old DOS days
- **CP 850 Multilingual** Slightly less old DOS days
- **CP 852 Slavic**
- **CP 860 Portuguese**
- **CP 10000 Macintosh** They just have to be different, don't they?
- **CP 863 Canadian**
- **CP 865 Nordic**

For data that's been keyed in North America in the past 5 years or so, **CP 1252 Windows** is usually the Code Page that was used. If it was keyed overseas or a while ago, anything goes.

Import a Translation Table from Dirty Harry Allows you to import tables from our Dirty Harry's Character Assassin program. This also means that you can use Dirty Harry to build custom translation tables. Note that this doesn't change your data in any way, just how MatchUp compares data.

If you have created your own translation tables (the code pages we didn't cover), please share these with us so we can supply them to all our users.

Processing: Counting Method

Processing: Counting Methods

MatchUp can populate a Count Field in one of three ways (in the examples below, assume that you have a group of three duplicates):

- All records get a count of duplicates in the group

Record 1 (Output) Count=3

Record 2 (Dupe) Count=3

Record 3 (Dupe) Count=3

- Output record gets count, dupes get sequenced

Record 1 (Output) Count=3

Record 2 (Dupe) Count=1

Record 3 (Dupe) Count=2

- Output and dupe records get sequenced

Record 1 (Output) Count=1

Record 2 (Dupe) Count=2

Record 3 (Dupe) Count=3

Multi-Buyer Counting Method

MatchUp can populate a Multi-Buyer Count Field in one of three ways (in the examples below, assume that you have a group of five duplicates, and the output record is from **File1**. **Bold** records are counted).

- Count all Multi-Buyer hits

File1 File1 File1 **File2 File2 File3** Count=3

- Count only unique Multi-Buyer hits

File1 File1 File1 **File2 File2 File3** Count=2

- Count unique Output and Multi-Buyer hits

File1 File1 File1 **File2 File2 File3** Count=3

This determines how your Count and Multi-Buyer Count fields are populated. The best way to understand your options is to carefully examine the examples on the screen.

Processing: Gathering

Processing: Gathering

How should MatchUp gather information with respect to the Duplicate record table?



Gather information to both the Output and Duplicate tables.



Gather information to only the Output table.

Gather information to both the Output and Duplicate tables MatchUp will gather information from source tables and copy it to the output tables and any dupe tables.

Gather information to only the Output table MatchUp will copy the information only to the output tables.

Processing: Stacking

Processing: Stacking

The easiest explanation is with an example:

Input Records:

John	Smith	781-545-7300
Sally	Smith	
Bobby	Smith	800-777-2460

Align Stack Levels:

John	Smith	781-545-7300
Sally	Smith	
Bobby	Smith	800-777-2460

Top Justify Stack Levels:

John	Smith	781-545-7300
Sally	Smith	800-777-2460
Bobby	Smith	

This determines how your Stacking data is populated. The best way to understand your options is to carefully examine the examples.

Processing: Status Codes

Processing: Status Codes

How should MatchUp record status codes? If you're in doubt, choose ASCII.

- ASCII
(A'=Dupe, B'=Has Dupe, etc)
- Binary
(0x01=Dupe, 0x02=Has Dupe, etc)

MatchUp collects the matchcode combination(s) that caused a match. This information can be recorded in the Status Code field.

- Record Matchcode Hit Combinations

In Merge/Purging and Deduplicating, you are given the opportunity to specify an output status field. This field may be formatted as a Binary or as ASCII characters.

Binary saves space, but it is difficult to read. Basically, each status code (a number) is stored in a four byte binary representation. For example, the number "12345" is stored as "90".

Alternately, ASCII Characters can be used.

Purge & Merge/Purge Status Codes:

ASCII	Hexadecimal	Decimal	Description
A	0X20	32	Record is a Dupe.
B	0X10	16	Output record has a Dupe.
C	0X00	0	Output record has no Dupes (it is unique).
D	0X02	2	Record was suppressed.
E	0X08	8	Record was not intersected.
F	0X04	4	Intersection record was used to intersect.
G	0X01	1	Suppression record caused a suppression.

In addition to this code, MatchUp can tell you which matchcode was used in a match. Immediately following the status code will be the number(s) of the matchcode combination(s) that caused the hit. If the binary format is used, this will be a 16-bit mask of the OR'd values. If you don't know what a "16-bit mask" is, stick with ASCII.

Processing: Multi-Threading

Multi-Threaded processing **is enabled**. If you are running a multi-processor machine, multi-threading can use additional processors for faster processing. Check the documentation for more information.

1: Maximum Threads when building keys (optimal = number of processors present):

 **0** Threads

2: Maximum Threads when deduping (optimal = number of processors present):

 **0** Threads

If you have purchased the multi-processor add on, this option will be enabled. This gives you control over how many simultaneous threads can be used during the two dedupe passes.

This option requires a bit of trial and error as the best settings may not necessarily be the number of processors. Other factors include disk access speed, available memory, and other running process.

Multi-threading works in this way:

Pass 1 - Building Keys:

A thread is launched for the first source table. If the maximum thread count is more than one, a second thread is launched for the second source table. This continues until the maximum thread count is reached. Once this happens, new threads (for additional source files) are only launched when one becomes available (ie a table's keys have all been built).

Obviously, for a single input table, multi-threading this pass has no advantage.

Threads are assigned to files in the order they are listed in the Input Tables tab. It is usually best to put your larger tables at the top of the list. For example, say you have 3 input tables, 2 medium-sized, and 1 large, and are processing with 2 threads:

If they are ordered 1, 2, 3 in the setup, thread allocation would look like this:



If they are ordered 3, 1, 2 in the setup, thread allocation is more like this:



The second scenario is quite a bit faster. MatchUp is not able to figure out which table is going to take the longest (it is not necessarily the table with the most rows), so you need to determine this yourself and perform the appropriate action.

In most cases, the optimal maximum thread setting is equal to the number of processors in your machine.

Pass 2 - Deduping:

A thread is launched for each cluster. MatchUp typically processes several thousand clusters in a single run, so one would think multi-threading would really speed this process up. Unfortunately, this isn't usually the case, as each one of these threads are vying for the same resources, primarily the key file, source tables, and output table(s). When one thread is accessing any a resource, any other threads must wait until the first thread is done with the resource before they can access it (known as 'resource locking'). Although you may find otherwise (experimentation with your data is the key), usually a setting of 0 or 1 work best here.

0 Threads versus 1 Thread:

MatchUp without the multi-threading option uses a setting of 0 threads. What this means is that it never launches threads while processing. For example, during pass 2 (0 threads), the calling program (aka, the 'thread master') actually pauses while a cluster is processed and waits until it is complete before figuring out the makeup of the next cluster. If the maximum thread setting is 1 or more, the thread master keeps itself busy by looking for the next cluster (ie, it builds up a 'thread queue'). The thread master can keep itself pretty busy with this work, to the point that it steals processing cycles from the launched threads. For this reason, usually a pass 2 setting of 1 is better than 2 on a 2 processor machine.

7.5.4 Analyzer

Analyzer

Display Font: Size:

Printer Font: Size:

Default Show/Hide settings:

Records from Regular, Self-Purge, and No-Purge List(s):

Unique Records Suppressed Records
 Records having Duplicates Non-Intersected Records
 Duplicate Records

Records from Suppression List(s):

None
 Only Hits
 All Records

Records from Intersection List(s):

None
 Only Hits
 All Records

These settings are used as a default when you analyze a setup for the first time as well as when you reset the Analyzer display.

Display Font/Size The font used for the analyzer display.

Printer Font/Size The default font used when printing.

Records from Regular, Self-Purge and No-Purge List(s):

- **Unique Records** Show records with no matches.
- **Records having Duplicates** Show all records that have matched other records.
- **Duplicate Records** Show duplicate records (this doesn't include suppressed or non-intersected records).
- **Suppressed Records** Show records that were matched against a suppression list(s).
- **Non-Intersected Records** Show records that didn't match against any intersection list(s).

Records from Suppression List(s):

- **None** Don't show any records from the suppression list(s).
- **Only Hits** Only show records that were actually used to suppress other records.
- **All Records** Show all records from the suppression list(s).

Records from Intersection List(s):

- **None** Don't show any records from the intersection list(s).
- **Only Hits** Only show records that were actually used to intersect other records.
- **All Records** Show all records from the intersection list(s).

7.5.5 Reporting

Reporting

Display Font: Size:

Printer Font: Size:

Template Reports are presented with a bitmap at the top left, usually used for a corporate logo. The default logo is the **Melissa Data** logo. You may have a more appropriate one in mind!

The logo file must be a **Windows Bitmap** file (.bmp). For best results it should be about 200 pixels by 200 pixels

Logo Bitmap: 

Multi-Buyer Statistic

MatchUp can report the Multi-Buyer counts seen in the File Summary in several ways (in the examples below, assume that you have a group of five duplicates, and the output record is from **File1**. **Bold** records are

- Number of multi-buyer dupe groups
File1 File1 File1 File2 File3 Count=1
- Count of all Multi-Buyer hits
File1 File1 File1 **File2 File2 File3** Count=3
- Count only unique Multi-Buyer hits
File1 File1 File1 **File2 File2 File3** Count=2
- Count unique Output and Multi-Buyer hits
File1 File1 File1 File2 File2 File3 Count=3

Allows you to change the default display and printer fonts for Reporting.

Display Font/Size The font used for the analyzer display.

Printer Font/Size The default font used when printing.

Logo Bitmap Template Reports usually print a bitmap at the top of the first page. By default, this is the Melissa Data Logo. You probably have a better one in mind. To use it, create a Windows bitmap, sized approximately 200 x 200 pixels and specify it here.

Multi-Buyer Statistic The Multi-Buyer count that appears on the File Summary and Source Code File Summary can be reported in a variety of ways, as shown in the examples on the screen.

8 Additional Tools

8.1 View File

Allows you to view the an open table.



FIRST	LAST	STREET
Sunny	Nyman	1634 Queens Rd.
Alex	Short	7797 Bear Creek Rd.
M	McAleer	3706 Wood Duck Rd.
Austin	Stewart	1141 Port Tack
Randolph	Abbot	5149 Live Oak Rd.
Rita	Payne	1771 Fiddlers Cove
Woody	Reeves	2128 Pinto Ct.
Hector	Hesse	4624 Headlands Dr.
Allison	Bevis	8382 Scarborough Head Rd.
T	Pitts	4072 Squire Pope Rd.
Wallace	Fennell	7087 River St.
Shelly	Tanner	2910 Windflower Ct.
Earl	Farmer	7331 Spindle Ln
Cardio	Trix	7336 Captains Cove
Brendaline	Newton	1664 Queens Grant
Karla	Lane	308 Stones Throw Dr.

Right-clicking gives the options:

- **Auto-Size Column** Change the width of a column to fit the longest string.
- **Auto-Size All Columns** Change the width of each column to fit its longest string.
- **All Columns This Size** Change the width of each column to the size of a column.
- **Reset Columns** Reset all column widths to their original sizes.

8.2 Field List

Lists the fields in each source table.

Fields shown here with a cyan background have already been used in the setup, ones with a white background have not.

When you place the mouse over a field, general information about that field is displayed at the bottom of the window:



inhouse	national	suppress	househol...
FIRST	NAME	ADDR1	NAME
LAST	STREET	ADDR2	STREET
STREET	CITYSTZIP	CSZ	CITY
CITY	LASTCALL		STATE
STATE	AGE		ZIP
ZIP	SALARY		LASTCALL
LASTCALL			SALARY
AGE			AGE
SALARY			

FIRST Fixed Character Size: 20

The Field List isn't just for show, however. You can click and drag fields from the field list into input and output fields in your setup:

The screenshot shows the 'Purge Setup' application with the 'Matchcode Mapping' tab selected. The main window displays a table for field mapping and configuration options for a matchcode component and an input field.

Field Mapping for Matchcode: Address, Company OR Last Name

Component	inhouse	national	suppress	household
Zip5	ZIP	CITYSTZIP	CSZ	ZIP
Last Name	LAST	NAME		NAME
Company				STREET
Address Line 1	STREET		1	STREET
Address Line 2			ADDR2	
Address Line 3				

Matchcode Component:
 Data Type: Zip5
 Label:
 Size: 5
 Start: Left
 Match Strategy: Exact
 Short/Empty: Both Fields
 Swap: None

Input Field:
 Name:
 Field Type:
 Size:
 Data Type:
 Conversion:
 Decimals:
 Change Data Type...

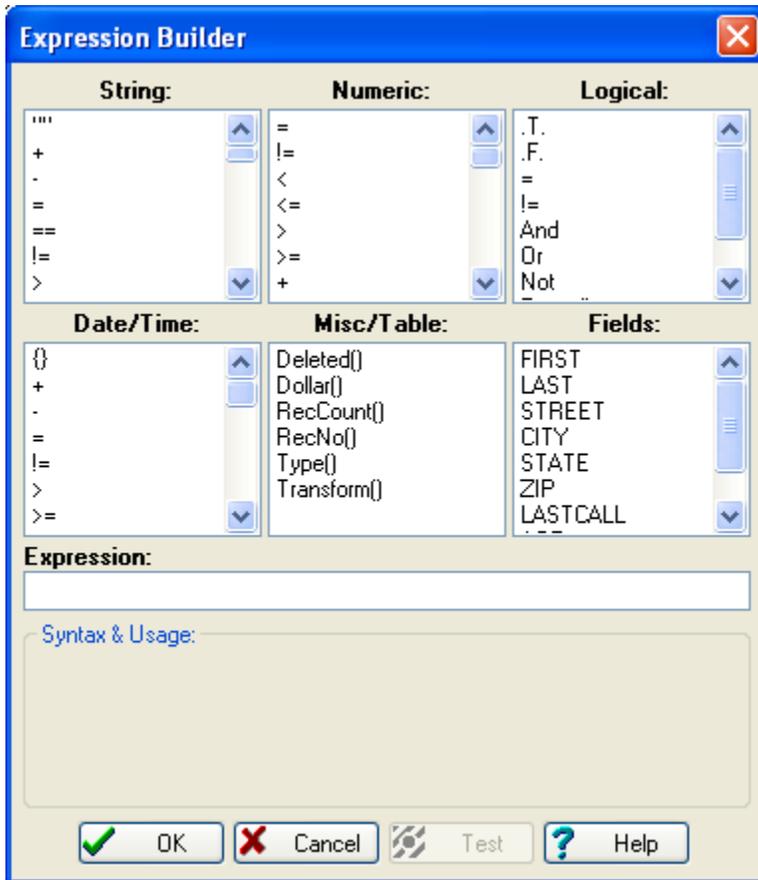
Drag and Drop to Map Fields

inhouse	national	suppress	househol...
FIRST	NAME	ADDR1	NAME
LAST	STREET	ADDR2	STREET
STREET	CITYSTZIP	CSZ	CITY
CITY	LASTCALL		STATE
STATE	AGE		ZIP
ZIP	SALARY		LASTCALL
LASTCALL			SALARY
AGE			AGE
SALARY			

STREET Fixed Character Size: 30

Navigation buttons: Previous, Next, OK, Cancel, Help

8.3 Expression Builder



The Expression Builder is a tool to aid you in creating dBASE expressions. Basically, you build your expression step by step. See dBASE syntax for more information on what makes up a dBASE expression. There is some overlap between some of the different data types as some functions are useful with different types.

String Available string operators and functions. Double-click a selection to insert it into **Expression**.

Numeric Available numeric operators and functions..

Logical Available logical operators and functions.

Date/Time Available date operators and functions.

Misc/Table Functions that are specific to table operations and/or dont really fit into any other category.

Fields This list allows you to select the fields in the current database.

Expression This area shows the expression as you build it.

Test Click this button to ensure that the dBASE expression shown is valid in syntax. This won't verify whether the expression will do what you want - only you can tell.

8.4 Check Mapping

Check Mappings for national2.dbf:

Mapped Fields: These fields have already been mapped:	Doubly Mapped: These fields have been mapped at least twice (maybe by mistake?):	Unmapped Fields: These fields have not been mapped, but Output Fields exist with the same name:	Missing Fields: These fields have not been mapped and no Output Fields exist with the same name:	
<p>NAME STREET CITYSTZIP LASTCALL AGE SALARY</p>	<p>NAME CITYSTZIP</p>	<p></p>	<p></p>	
<p>No action will be performed with these fields.</p>	<p>No action will be performed with these fields.</p>	<p>Check the fields that you would like to map to same-named Output Fields.</p>	<p>Check the fields that you would like to add as Output Fields (they will be mapped as well).</p>	
<p>← Previous</p>	<p>Next →</p>	<p>✓ OK</p>	<p>✗ Cancel</p>	<p>? Help</p>

When merging files, it is important that all input fields line up and that no data is lost. MatchUp makes this chore a lot easier by telling you which fields have not been used. You can use Check Mapping on each file used in the merge or have been used more than once.

Mapped Fields These fields are currently being used in the merge.

Doubly Mapped These fields have been mapped more than once, possibly by mistake.

Unmapped Fields These fields are not being used in the merge, but there is an output field that has the same name. When auto mapping is on, you won't see fields in this box unless you have unmapped the field yourself. With the auto-mapping off, you could end up with several fields in this box. If you would like to map the field to the output file's structure, check the box next to that field.

Missing Fields These fields are not being used in the merge and there isn't an output field having the same name. If you would like to add a field to the output file's structure, check the box next to that field.

8.5 Check Truncations

Check Truncations for Demo2.dbf:

These fields below have been mapped to Output Fields. Fields with a disabled check box (grayed) are not in danger of being truncated. Fields with an enabled check box could possibly be truncated during processing.

Note that we say *could be truncated*. This is because only field sizes are used to determine truncations, *not content*. So if you have a 200 character state field that is mapped into a 2 character state field, it will be reported here.

<input type="checkbox"/>	In: NAME (30) Out: FIRST (20)		<input type="button" value="All"/>
<input type="checkbox"/>	In: NAME (30) Out: LAST (20)		<input type="button" value="None"/>
<input type="checkbox"/>	In: STREET (30) Out: STREET (30)		
<input type="checkbox"/>	In: CITYSTZIP (30) Out: CITY (30)		
<input type="checkbox"/>	In: CITYSTZIP (30) Out: STATE (2)		
<input type="checkbox"/>	In: CITYSTZIP (30) Out: ZIP (10)		
<input type="checkbox"/>	In: LASTCALL (8) Out: LASTCALL (8)		

If you would like to adjust an Output Field's size to accommodate an Input Field, check the box of that field.

Note:

There is absolutely no easy way that MatchUp can find when truncations have actually occurred. This option tells you when the *potential* for truncations exists, not if they do exist. All output fields are listed, but the ones that don't have a potential for truncation are greyed and can not be selected.

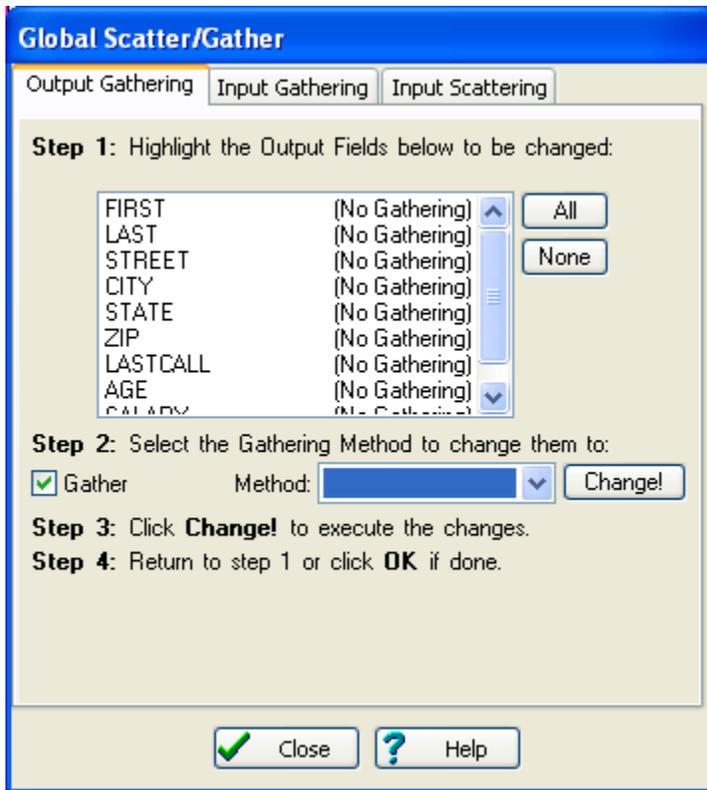
Say you have a City/State/Zip field of 30 characters, and you elect to split the field into a City field of 20 characters. MatchUp has no way of knowing if you have any cities that are longer than 20 characters. To be on the safe side, MatchUp warns you about the possibility of truncation here.

8.6 Global Scatter/Gather

When you need to specify gathering and/or scattering for several fields or files, it can get quite tedious to setup each file. To make this easier, use the Global Gather/Scatter dialog:

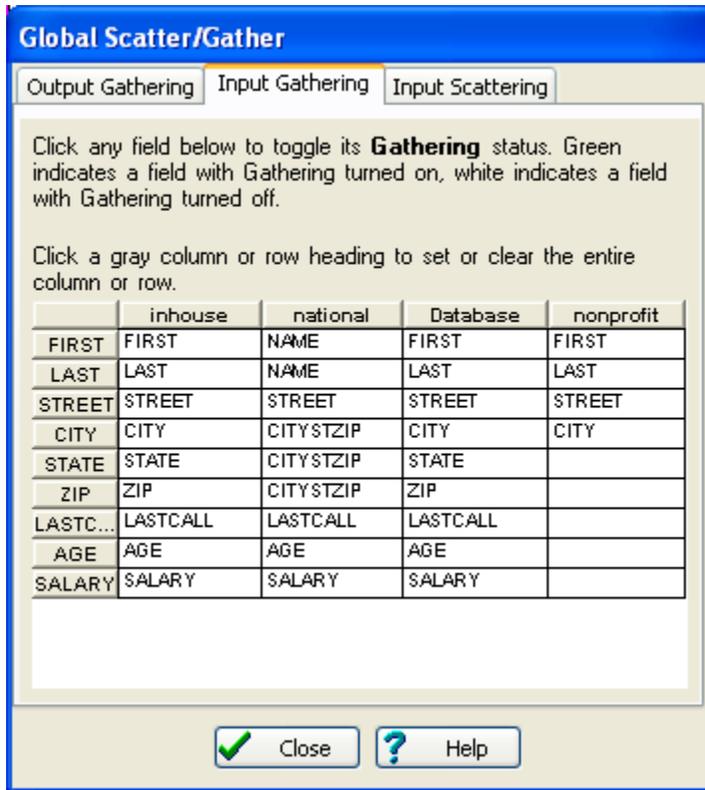
- **Output Gathering**
- **Input Gathering**
- **Input Scattering**

8.6.1 Output Gathering



Once you select the field(s) you want to gather to, check **Gather** and then select the **Method** you want to use. To execute your selections, click **Change!**

8.6.2 Input Gathering



Select the source fields you want to use while Gathering. Left-click the fields that you want to change. Fields shown in green have their gather/scatter setting on, fields in white have their setting off. Left-clicking a row or column heading will toggle the entire row or column.

Note:

You are able to select any field here, even if Gathering has not been enabled for its Output Field. The **Gather** status on the main screen will be grayed until Gathering has been enabled for its output field.

8.6.3 Input Scattering

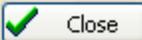
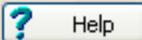
Global Scatter/Gather

Output Gathering Input Gathering **Input Scattering**

Click any field below to toggle its **Scattering** status. Green indicates a field with Scattering turned on, white indicates a field with Scattering turned off.

Click a gray column or row heading to set or clear the entire column or row.

	inhouse	national	Database	nonprofit
FIRST	FIRST	NAME	FIRST	FIRST
LAST	LAST	NAME	LAST	LAST
STREET	STREET	STREET	STREET	STREET
CITY	CITY	CITYSTZIP	CITY	CITY
STATE	STATE	CITYSTZIP	STATE	
ZIP	ZIP	CITYSTZIP	ZIP	
LASTC...	LASTCALL	LASTCALL	LASTCALL	
AGE	AGE	AGE	AGE	
SALARY	SALARY	SALARY	SALARY	

 Close  Help

Select the source fields you want to use while Scattering. Left-click the fields that you want to change. Fields shown in green have their gather/scatter setting on, fields in white have their setting off. Left-clicking a row or column heading will toggle the entire row or column.

Unlike Gathering, the Output Field does not need to have Gathering turned on.

9 Reference

9.1 Command Line

Several of MatchUp's options are available only from the command line. Most of these options are used in "hands off" processing (what we call batch processing), while others can save you the time of selecting the same files every time you run MatchUp. The syntax for MatchUp's command line is:

dt3 [setupfile] [options]

[setupfile] The name of the setup file to use in processing. If the setup file is not in the working directory, you must specify the file's path.

[options] Processing options:

- **/T** Used in Batch Processing to test a batch run.
- **/R** Used in Batch Processing to execute a batch run.

Note that with no options, specifying [setupfile] will only open this file, but will not begin processing until you choose to.

The switches are not case-sensitive and can occur in any order. "-" can be used instead of "/". Each switch must be preceded by a space, however.

You can specify command line options in one of two ways:

- Creating a new shortcut: Right click on the Desktop, select **New | Shortcut**. For the **command line**, enter the desired command line. It is usually a good idea to specify the full path of the MatchUp executable, as well as the database and setup paths. For example,

```
c:\dt\dt3.exe c:\job285\fallmail.dt
```

would make MatchUp automatically open the fallmail.dt setup. Since the setup file contains the specifications for both source and output tables, it is unnecessary to specify them.

- Running MatchUp from a DOS session. From the **Start Menu**, select **Command Prompt**. At the DOS prompt, change to the desired working directory and simply type the desired command line. MatchUp will be opened in a new window.

Long file names with spaces?

For example, your setup file is called *Merge Purge.Dt*. Just surround it with double quotes. When using long file names having spaces, applications can get confused as to where one parameter stops and the next one begins. Surrounding a parameter with quotes solves this problem. This works on any 32-bit Windows program. Note that you do this for each parameter, not the whole command line (a common mistake). For example:

```
c:\dt\dt3.exe "c:\Job Folder\Merge Purge.dt" /r
```

Or another, more complicated example (notice the careful use of double quotes here):

```
"c:\Program Files\Dt3\Dt3.exe" "c:\Job Folder\Merge Purge.dt" /r
```


9.2 Batch Processing

You can run MatchUp completely uninterrupted from a shortcut or DOS command line. To let MatchUp know that you want this special capability, add one of these switches onto the end of the command line:

- **/T** To test that the setup file will run when the time comes.
- **/R** To run the setup file.

As outlined in the command line section, there are one or two ways to use these switches. You can exploit the DOS session method even further by using a batch file (just like in regular old DOS!). The only caveat to this method is that you should check the **Always Suspend** option in the shortcut's property dialog box. Why? Take the following sample batch:

```
c:\dt\dt3 c:\work\file.dt /R
c:\sl\sl32 c:\work\file.dbf c:\work\file.sl /R
```

With **Always Suspend** checked (nowadays more often than not), the second line (the StyleList line) will not be executed until the first line is complete. Without "Always Suspend" checked, the second line executes moments after the first line is *started*. Not a desirable situation when the input of the second line depends entirely on the successful completion of the first line!

On some systems, the "Always Suspend" setting is not enough. There are many script processing utilities that have addressed this problem and provide a solution. Also, the DOS START command will often do the trick if you use the /WAIT switch. For example:

```
start /wait c:\dt\dt3 c:\work\file.dt /R
c:\sl\sl32 c:\work\file.dbf c:\work\file.sl /R
```

For programmers, there's an alternate solution: When MatchUp is done processing in batch mode, it creates a file called *dt.flg*. You can use the presence of this file as an indication that it is okay to continue. For example, in FoxPro:

```
erase d:\dt\dt.flg                && erase the flag file
run d:\dt\dt3.exe "d:\dt\merge purge.dt" /R  && launch MatchUp

&& Loop until the flag file appears:
do while !file("d:\dt\dt.flg")
    =inkey(1000)                    && sleep for a second
enddo
```

Note the use of double quotes around the setup file. When using long file names having spaces, applications can get confused as to where one parameter stops and the next one begins. Surrounding a parameter with quotes solves this problem.

9.3 Errors

General File Errors:

General file errors are situations that may occur while attempting to access a database, setup, or ASCII file:

- **Open Database fail** The specified database couldn't be opened, or perhaps the specified folder doesn't exist.
- **Open Table fail** The specified table couldn't be opened. It may not exist or you may not have the necessary permission to open it.
- **Create Table fail** The specified table couldn't be created. You may not have the necessary permission needed to create it.

Setup File Errors:

Setup file errors can occur while attempting to open a setup file:

- **File doesn't appear to be a MatchUp Setup file** You've specified a file that is not a MatchUp Setup File or the Setup File has become corrupt.

DBMS File Errors:

DBMS file errors can occur while processing, but shouldn't:

- **Read error** An unexpected error occurred while reading the specified file. Try the operation again. If the situation still persists, call Melissa Data Technical Support.
- **Write error** An unexpected error occurred while writing to the specified file. Try the operation again. If the situation still persists, call Melissa Data Technical Support.

dBASE Expression Errors:

dBASE errors occur when you are entering a dBASE expression as a While, For, Filter or Sort expression.

- **Syntax error** The expression is syntactically incorrect. Check for a misspelled function or field name.
- **Mismatched Parenthesis** The expression is missing a right or left parenthesis. Remember that you should have a matching number of left and right parenthesis.
- **Mismatched Delimiters** A literal character string was specified with non-matching (or missing) delimiters.
- **Expression does not evaluate to data type** The expression does not result in the specified data type. For example, "LTrim(First)" (a character expression) is not an appropriate expression for a Filter condition, which requires a logical expression.
- **Field does not exist** Specified field does not exist. Check for a misspelling.
- **Function does not exist** Function is not valid. Check for a misspelling.
- **Empty Expression** No expression was specified, although one is required.
- **Invalid number of arguments** Too many or too few parameters were given for the specified function. This is usually the result of a misplaced or missing parenthesis, comma, or misspelling.

- **Type error** One of the parameters given for the specified function is of the wrong type. For example, "SubStr(RecNo(),12,3)" will result in this error because SubStr requires a Character expression and RecNo() is Numeric.

Other Errors:

These errors may occur anytime while running MatchUp and, although you may find this hard to believe, may indicate a bug in the program:

- **Unknown error**
- **Exception Error**
- **GPF**

9.4 dBASE

9.4.1 dBASE Syntax

While it is beyond the scope of this documentation to explain the full use of dBASE syntax (you should see a book or manual on dBASE, Clipper, or FoxPro for that), some rudiments are explained below.

dBASE syntax involves using quasi-English expressions to tell the program what you want to do. These expressions use Operators, Functions and Literals in conjunction with the Field Names in your database.

Definitions:

- **FOR Condition** A True/False expression determining which records will be affected during a database operation. Each record in the table is evaluated "for" this condition.
- **WHILE Condition** An operation that will continue only "while" the expression is True, then it will stop.
- **SORT Expression** An expression that will be evaluated for each record to be sorted. It can result in a Character, Number, Date, or logical value.
- **FILTER Condition** Similar to a "for" condition, except in its context. "For" and "while" expressions are always performed in conjunction with another operation, whereas **filter** conditions occur all by themselves and affect any subsequent operations like a "for" would.

Types of Expressions:

There are four types of expressions: Character, Date, Numeric, and Logical. In this document, they look like this:

- **<Cexp>** Character expression.
- **<Dexp>** Date expression.
- **<Nexp>** Numeric expression.
- **<Lexp>** Logical expression.
- **<Exp>** Any type of expression

Expressions can consist of...

a literal	<code>"Melissa Data"</code>
an expression	<code>SubStr("Melissa Data", 5, 3)</code>
a field	<code>Name</code>

Specifying Literals:

Literals must be specified in a special way, so the program can distinguish between character, date, numeric, or logical literals.

- **Character Literals** Delimit with double quotes (") or single quotes ('). For example, "Melissa Data" and 'Melissa Data' specify the same character literal. You cannot embed the delimiters. For example 'MacDonald's' is not valid (so use one of the other delimiters, like "MacDonald's"). Note that the old square bracket delimiters are no longer allowed. They are now used to delimit field names.
- **Date Literals** Delimit with curly braces ({ and }), and use the format {mm/dd/yyyy}. For example, {3/10/1968} represents March 10, 1968. In the old days, there was no such thing as a date literal, and people usually specified them by converting a character literal (for example CToD("3/10/1968")). If you omit the century (i.e. {03/10/68}) the year will be assumed to be between 1950 and 2050.
- **Numeric Literals** There are no delimiters, just specify the number. For example, 3.14 is a numeric literal.
- **Logical Literals** There's only two, so they're special: .T. for true, and .F. for false. Yes, those periods are required.

Specifying Functions:

Functions are fairly simple. Type the function's name, a left parenthesis, any parameters (separated by commas), and a right parenthesis. If you nest functions, be sure that you consistently match parenthesis. Following the dBASE tradition, casing is not significant and only the first four letters of a function's name are needed, so...

```
ALLT(SUBS(NAME,5,4))
```

is the same as

```
AllTrim(SubStr(Name,5,4))
```

Specifying Fields:

In most cases, simply specifying the field name will work. However, if the field contains spaces, you must delimit the field with square brackets, as in [First Name]. dBASE tables don't allow embedded spaces, but many other DBMS's do.

See dBASE Functions (Alphabetic listing), dBASE Functions (Categorical listing) in the Help Files for more information.

9.5 About Melissa Data Corporation

Thank you for using our problem-solving software for mailing lists. Our programs are designed to help people to communicate more personally (and more effectively) in this increasingly computerized (and impersonal) age:

We offer these solutions to today's mailing list problems:

- **MatchUp™** combines many powerful and convenient features like letting you process files in different formats (Xbase/ASCII) and different field structures directly, saving you the time to make them uniform. Matchcodes can use any parts of any fields, and you can process up to 16 matchcodes simultaneously. Does suppression and intersection runs, parses street addresses, and offers complete, editable reports.
- **StyleList™** makes your labels and envelopes more attractive, so they stand out and get opened more often. It converts All-upper to Upper/Lower case (or vice versa), expands user-definable abbreviations (or creates them), corrects punctuation, and controls field lengths.
- **Personator™** increases response rates and profit from direct mail by (1) splitting Full or Inverse name fields into Prefix, FN, MN, LN, and Suffix/Title, (2) adding correct prefix (genderizing), and (3) making perfect salutations. It simplifies merge/sort/deduplication. Pro Series splits addresses.
- **GenderBase 100™** is an optional Personator add-on with over 100,000 single word, un-hyphenated first names (the genderizing table supplied with Personator has the 8,500 most common). When used with Personator pro series it can correct common abbreviations and typos.
- **Right Fielder™** "re-fields" badly fielded lists: distinguishes names from companies, primary from secondary address, domestic from foreign, etc. Imports "print image" and "label format" files into true databases
- **Dirty Harry's Character Assassin™** is a power user tool that strips or replaces any ASCII character with any other at a blazing speed.

For programmers, we offer several products as programming APIs:

- **Personator API™** Name splitting, genderizing, salutation building, and street splitting.
- **Right Fielder API™** "re-fielding" data.
- **StyleList API™** Casing, abbreviating, abbreviation expansion, e-mail, phone, and zip/postal code reformatting.
- **MatchUp API™** Matches database records.

APIs are provided as 32-bit Windows DLLs. These are callable from C/C++, Visual Basic, Foxpro, and other languages that support DLL function calls.

Typical uses are:

- Real-time data entry validation.
- Enhancement of databases with custom file structures (i.e., files that cannot be exported into a format that our Windows products can readily use).
- Providing functionality for a custom written application seamlessly (i.e., without shelling to another application and with the desired look and feel).

9.6 Copyright & License Agreement

Copyright 1995-2007 Melissa Data, Inc.

This Melissa Data Corporation program ("the Software") is protected by both United States Copyright Law and International Treaty provisions.

By installing this software you accept all of the terms contained in this license agreement, so read this license agreement carefully before installing this Software.

You may either (a) make one copy of the Software for backup or archival purposes only, or (b) transfer the Software to a single hard disk provided you keep the original disks for backup or archival purposes only.

You may use the Software (a) on any single-user computer provided you have physical possession of the CD at all times during use of the Software, or (b) on a network provided each person using the Software has purchased a separate copy of the Software and each such person has physical possession of the CD provided with it. You may not temporarily transfer possession of the right to use the Software to another individual.

To permanently transfer the Software to another party, you must at the same time either transfer all copies of the Software (regardless of form) to the same party or destroy any copies not transferred. You may terminate this license at any time by destroying all copies of the Software (in any form). Your license terminates automatically if you fail to comply with any terms or conditions of the Melissa Data Corporation License.

You may not copy, reproduce, or distribute the Software nor alter, modify, reverse engineer, decompile, disassemble, or otherwise attempt to render source code from the Software. You may not copy or otherwise reproduce the Manual or Help Files, in part or in whole, without the prior written consent of Melissa Data, Inc.

MatchUp, MatchUp API, Street Smart, CASSmate, StyleList, StyleList API, Personator, Personator API, GenderBase 100, Right Fielder, Right Fielder API, Dirty Harry's Character Assassin, and Melissa Data are registered trademarks of Melissa Data, Inc.

Borland C++ is a registered trademark of Borland International, Incorporated.

Microsoft Visual Basic, Microsoft Visual C++, Microsoft FoxPro, Microsoft Visual FoxPro, Microsoft Access, Microsoft Excel, Microsoft Word, SQL Server 7, Microsoft SQL Server 2000, Microsoft Windows 95/98/ME, and Microsoft Windows NT/2000/XP are registered trademarks of Microsoft Corporation.

Oracle is a registered trademark of Oracle Corporation.

DB/2 is a registered trademark of IBM Corporation.

AddressObject and Melissa Data are registered trademarks of Melissa Data Corporation.

ZIP Code, ZIP + 4, and CASS are registered trademarks of the United States Postal Service (USPS).

All other brands and products are trademarks of their respective holder(s).

9.7 Limited Warranty

Melissa Data warrants that the Software will perform substantially in accordance with Melissa Data's current published specifications, documentation, and authorized advertising; that the Help File(s) and Manual (if provided) contains the necessary information to use the Software; and that the media (if provided) on which the Software is furnished will be free from defects in materials and workmanship for a period of ninety (90) days from the date of purchase. The remedy for breach of this warranty is limited to replacement or refund at your discretion and shall not encompass any other damages, including but not limited to loss of profit, special, incidental, consequential, or other similar claims.

EXCEPT FOR THE LIMITED WARRANTY ABOVE, THE SOFTWARE, MANUAL, AND HELP FILE(S) ARE PROVIDED "AS IS". Melissa Data MAKES NO OTHER WARRANTY, EXPRESS OR IMPLIED WITH RESPECT TO THE SOFTWARE AND/OR HELP FILE(S) AND SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF THE MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. PEOPLESMTIH DOES NOT WARRANT THAT THE SOFTWARE AND/OR HELP FILE(S) WILL MEET YOUR REQUIREMENTS OR EXPECTATIONS OR THAT THE OPERATION OF THE SOFTWARE WILL BE UNINTERRUPTED AND/OR ERROR FREE. YOU ARE SOLELY RESPONSIBLE FOR THE SELECTION OF THE SOFTWARE TO ACHIEVE YOUR INTENDED RESULTS AND FOR THE RESULTS ACTUALLY OBTAINED. IN NO EVENT SHALL PEOPLESMTIH BE LIABLE FOR ANY LOSS OF PROFIT OR ANY OTHER COMMERCIAL DAMAGE, INCLUDING BUT NOT LIMITED TO SPECIAL, INCIDENTAL, CONSEQUENTIAL OR OTHER DAMAGES.

This Agreement shall be construed, interpreted, and governed by the laws of the State of California.

See also: Copyright & License Agreement

Index

- < -

<n>th Select 264
Split File 264

- A -

Accurate Near 30, 295
Add Column 193
 Analyze 193
 Browse 270
Add New Output Field 74
 Merge 74
Add Output Field 94
 Merge/Purge 94
Add Values 54
 Gathering 54
Additional Output Fields CASS 106
Address Field Mapping 40
Advanced Combination Settings 299
Advanced File Type 124
 Purge 124
Advanced Rank 88
 Merge/Purge 88
 Purge 125
Advanced Tab 114
 Merge/Purge 114
 Purge 132
 Merge/Purge 86
 Purge 124
Alphas Only 30
Analyze 189
 Print 189
 Vertical Display 191
Analyzer fonts 332

AND Values 54
 Gathering 54
Append Records 244
ASCII Export 285
ASCII file 13
ASCII Import 278
 Manually Enter 283
Audio 324
Auto-Update 314

- B -

Background 312
 Main Window 312
Batch Processing 353
Blank Field Matching 36
Both Blank Fields 30, 36
Break Grouping 23
Browse 238
 Add Column 270
 Change Column 271
Browse Font 272
Browser options 272

- C -

Canadian Postal Code Component 26
Canadian Users 59
CASS 319
 Field Naming 319
 Merge/Purge 104
 Output Fields 320
 Setup 152
CASS Input Fields 104, 152
CASS Options 154
CASS Output 110, 159
CASS Processing 177
CASS Reports 230
CASS Subscription 12

- CASS Updates 12
- CASSmate on CD 10
- Change Column 195
 - Analyze 195
 - Browse 271
- Change Structure 275
- Changing Mappings 40
- Check Mapping 343
- Check Truncations 344
- City Component 26
- Clustering 23
- Combinations 25, 33
 - Advanced 299
 - Matchcode 298
- Command Line 350
- Company Acronym Component 26
- Company Component 26
- Component Mapping 40
- Component Properties 30
- Component Sequence 44
 - Optimization 44
- Components 26
 - Matchcode Editor 290
- Concatenate Fields 246
- Concepts 23
- Concurrent Matching 25
- Consonants Only 30
- Containment 30
- Containment (matching) 295
- Contents of Field 268
 - Split File 268
- Copy Records 248
- Copyright 364
- Count Records 248
- Counting Method 326
- Country Component 26
- Credit Card Component 26

- D -

- Database vs. Table 13
- DB2 21
 - File Requirements 21
- dBASE file 13
- DBASE Syntax 358
- Deduping 23
- Default Output Table 317
 - User Settings 317
- Delete Records 250
- Department/Title Component 26
- Double Take Errors 355
- Dupe Group 57
- Dupe Matrix 206
 - Reports 206
 - Source Code Reports 225

- E -

- Earliest Date 54
 - Gathering 54
- Edit Results 182
- E-Mail Component 26
- Enter Key 310
- Enter Key behavior 272
- Errors 355
- EULA 364
- Exit 67
- Export ASCII 285
- Expression Builder 340

- F -

- Fast Near 30, 295
- Features 6
 - new 6

- Field Info 282
 - Interactive Import 282
- Field List 338
- Field Mapping 40
 - Merge/Purge 90
 - User Settings 316
 - File Update 143
 - Purge 128
- Field Matching 70
 - Merge 70
- Field Naming 315
 - User Settings 315
- Field Naming CASS 319
- Fields 36
 - Both Blank 36
- File Info 281
 - Interactive Import 281
- File location default 313
- File Requirements 13
 - DB2 21
 - ODBC 15
 - Oracle 20
 - SQL Server 19
- File Summary 203
 - Reports 203
 - Source Code Reports 222
- File Update 143
 - Field Mapping for Matchcode 143
 - General Tab 137
 - Information About 142
 - Input Field 146
 - Input Sources 141
 - Input Tables 141
 - Matchcode 138
 - Matchcode Component 145
 - Matchcode Mapping 143
 - Ranking 139
 - Setup 136
- File Update Processing 174
- Find Record 241

- First Component 33
- First Component Requirements 33
- First Data 54
 - Gathering 54
- First Name Component 26
- First/Nickname Component 26
- Fonts 332
 - Analyzer 332
 - Reporting 334
- Foreign Characters 30, 325
- Foreign Charactes 30
- FoxPro 13
- Frequency 30
- Frequency (Matching) 295
- Full Address Component 26
- Fuzzy 30
 - Optimization 44
- Fuzzy Matching 30
 - Accurate Near 295
 - Containment (matching) 295
 - Fast Near 295
 - Frequency (Matching) 295
 - Phonetex 294
 - Soundex 294

- G -

- Gathering 53
- Gathering Methods 54
- Gathering settings 327
- Gender Component 26
- General Component 26
- General Tab 137
 - File Update 137
 - Merge/Purge 79
 - Purge 118
- Global Scatter/Gather 346
 - Merge/Purge 346
- Global Update 150

- H -

Hide Rows 186
 Analyze 186
How to Dedupe 23

- I -

Import ASCII 278
Index 243
 Browse 243
Inferred Matching 25
Information About 142
 File Update 142
 Merge Setup 69
 Merge/Purge 84
 Purge 122
Initial Only 30
Input Field 146, 148
 File Update 146, 148
 Merge 73
 Merge/Purge 92, 97
 Purge 131
 CASS 104, 152
Input Sources 141
 File Update 141
 Merge/Purge 83
 Purge 122
Input Tables 141
 File Update 141
 Merge Setup 68
 Merge/Purge 83
 Purge 121
Installation 10
Installing CASSmate 10
Interactive Import 279
 Field Info 282
 File Info 281

Interface 310
 User Settings 310
International records 62
Introduction 5

- J -

Join 54
 Gathering 54
 Gathering 54

- L -

Label 30
Last Name Component 26
Latest Date 54
 Gathering 54
License 364
Limited Warranty 367
Locate Again 185
Locate Record 241
Locate Row 185
Longest 54
 Gathering 54

- M -

Main Window 312
 User Settings 312
Manually Enter 283
 ASCII Import 283
Mapping 90, 128
 Merge/Purge 90, 128
Mapping Components 40
Mapping Fields 40
Mapping issues 343
Matchcode 138
 File Update 138

- Matchcode 138
 - Print 307
 - Purge 119
- Matchcode Combinations 298
- Matchcode Component 291
 - File Update 145
 - Merge/Purge 91
 - Purge 130
- Matchcode Components 26, 30
- Matchcode Editor 288
- Matchcode Mapping 40
 - File Update 143
 - Merge/Purge 90, 128
- Matchcode Matrix 216
 - Reports 216
- Matchcode Quality 219
 - Reports 219
- Matchcode Rules 80
 - Merge/Purge 80
- Matchcodes 25
 - Matchcode Editor 289
- Matching 295
 - Accurate Near 295
 - Containment (matching) 295
 - Fast Near 295
 - Frequency (Matching) 295
 - Phonetex 294
 - Soundex 294
- Matching Strategies 293
- Maximum Value 54
 - Gathering 54
- Maximum Words 30
- mceditor.exe 288
- Merge 74
 - Add New Output Field 74
 - Input Field 73
 - Output Field 72
 - Output Field Matching 70
 - Output Table 75
 - Reports 229

- Merge Processing 165
- Merge Setup 68
 - Information about 69
 - Input Tables 68
- Merge/Purge 83
 - Input Sources 83
 - Input Tables 83
 - Matchcode Mapping 90, 128
 - Output Fields 93
 - Setup 79
- Merge/Purge CASS 104
- Merge/Purge Processing 168
- Methods 54
 - Gathering 54
- Middle Name Component 26
- Middle/Nickname Component 26
- Minimum Value 54
 - Gathering 54
- Modify Structure 275
- Multi Threading 330
- Multi-Buyer Dupe Matrix 208
 - Reports 208
- Multi-Buyer Quality Matrix 213
 - Reports 213
- MultiBuyer Source Code 102

- N -

- Neighborhood Sorting 23
- New Features 6
- New Setup 63
- Numerics Only 30

- O -

- ODBC 15
 - File Requirements 15
- One Blank Field 30
- Open Setup 64

- Optimization 44
 - Component Sequence 44
 - Fuzzy Algorithms 44
 - Unnecessary Combinations 44
 - Unnecessary Components 44
- Optimize Matchcode 308
- Options 196
 - Analyze 196
 - CASS 154
- Options CASS 321
- OR Values 54
 - Gathering 54
- Oracle 20
 - File Requirements 20
- Other Software 361
- Other Uses for Swap Matching 50
- Output 110, 159
 - CASS 110, 159
 - Analyze 192
- Output Field 149
 - File Update 149
 - Merge 72, 74
 - Merge/Purge 96
 - Merge/Purge 93
- Output Field Matching 70
 - Merge 70
- Output Fields 93
 - Merge/Purge 93
- Output Fields CASS 320
- Output Table 75
 - Merge 75
 - Merge/Purge 99

- P -

- Partitioning 23
- Peoplesmith Software 361
- Phone Component 26
- Phonetex 30, 294

- PO Box Component 26
- Prefix Component 26
- Print Matchcode 307
- Print Records 251
- Print Rows 189
 - Analyze 189
- Processing 177
 - CASS 177
 - File Update 174
 - Merge 165
 - Merge/Purge 168
 - Purge 171
- Processing Mode 273
- Processing Speed 324
- Processing Summary 201
 - Reports 201
- Products 361
- Properties 36
 - Both Blank Fields 36
 - Component 30
- Purge 124
 - Advanced File Type 124
 - Advanced Rank 125
 - Advanced Tab 132
 - Advanced Table Type 124
 - Field Mapping for Matchcode 128
 - General Tab 118
 - Information About 122
 - Input Field 131
 - Input Sources 122
 - Input Tables 121
 - Matchcode 119
 - Matchcode Component 130
 - Ranking 120
 - Setup 118
- Purge Processing 171

- Q -

Quality Matrix 211
 Reports 211
 Source Code Reports 227

- R -

Random Select 266
 Split File 266
Ranking 139
 File Update 139
 Merge/Purge 81
 Purge 120
Recall Records 253
Replace Records 254
Report Settings 233
Reporting Fonts 334
Reports 230
 CASS 230
 Dupe Matrix 206
 File Summary 203
 Matchcode Matrix 216
 Matchcode Quality 219
 Merge Matrix 229
 Multi-Buyer Dupe Matrix 208
 Multi-Buyer Quality Matrix 213
 Processing Summary 201
 Quality Matrix 211
 Source Code 220
 View 198
Reports: Templates 235
Requirements 13
 File 13
 System 9

- S -

Save Setup 65
Save Setup As 66
Scatter/Gather 346
 Merge/Purge 346
Scattering 56
Screen Savers 310
Screen Updates 324
Search & Replace 256
Set Filter 242
Set Index 243
Settings 233
 Report 233
 Short/Empty 297
Setup 152
 CASS 152
 File Update 136
 Merge 68
 Merge/Purge 79
 Purge 118
Short/Empty Match 30
Short/Empty Settings 297
Shortest 54
 Gathering 54
Show/Hide Rows 186
 Analyze 186
Size 30
Software License 364
Sort Records 259
Sort Rows 188
 Analyze 188
Soundex 30, 294
Source Code Format 101
 Merge/Purge 101
Source Code MultiBuyer 102
Source Code Reports 220
 Dupe Matrix 225

- Source Code Reports 220
 - File Summary 222
 - Quality Matrix 227
- Specifying Matchcode Combinations 33
- Split File 261
 - <n>th Select 264
 - by Records/Bytes/Files 262
 - Contents of Field 268
 - Random Select 266
- SQL Server 19
 - File Requirements 19
- Stack Group 54
- Stacking 54
- Stacking settings 328
- Start 30
- State/Province Component 26
- Status Codes 329
- Street Name Component 26
- Street Number Component 26
- Street Post-Directional Component 26
- Street Pre-Directional Component 26
- Street Secondary Component 26
- Street Suffix Component 26
- Subscription 12
 - CASS 12
- Suffix Component 26
- Swap Match 30, 306
- Swap Matching 50
- System Requirements 9

- T -

- Table vs. Database 13
- Templates 235
- Tip of the Day 310
- Tool Tips 310
- Toolbar size 272, 312
- Translation Table 325
- Trim 30

- Truncations issues 344

- U -

- Unnecessary Combinations 44
 - Optimization 44
- Unnecessary Components 44
 - Optimization 44
- Update Methods 54
- Update Tab 147
- Updates 314
 - CASS 12
- Uppercasing 30
- User Settings 310
 - Default Output Table 317
 - Field Mapping 316
 - Field Naming 315
 - File Location 313
 - Interface 310
- Using CASS CD 10

- V -

- Vertical Display 191
 - Analyzer 191
- View File 336
- View Reports 198
- Visual FoxPro 13
- Vowels Only 30

- W -

- Warnings 323
 - User Settings 323
- Warranty 367
- What's New 6

- Z -

Zip4 Component	26
Zip5 Component	26
Zip9 Component	26