



Report Library

Double Take offers more different types of reports with more information than any other merge/purge program for the PC. Plus it offers more flexibility in viewing and printing. This means Double Take gives a mailing house just exactly the documentation it needs when presenting results (and invoices) to customers. Double Take's reports make the mailing house look totally professional.

- **Processing Summary** This report gives you general processing statistics.
- **File Summary** This report contains a detailed listing of what happened with each file.
- **Dupe Matrix** The Dupe Matrix give intra- and inter-file record counts.
- **Quality Matrix** This report gives duplicate counts regardless of where an output record came from (ideal for obtaining "saturation counts").
- **Matchcode Matrix** This report lists the number of matches made by each matchcode combination.
- **Matchcode Quality** This report tells you how well each matchcode component was populated, giving you an idea as to the quality of your data.
- **Source Code List** Lists each source code detected during processing.
- **Source Code File Summary** This report is just like the File Summary, but is based on source codes rather than files.
- **Source Code Dupe Matrix** This report is just like the Dupe Matrix, but is based on source codes.
- **Source Code Quality Matrix** Just like the Quality Matrix, but based on source codes instead of files.

On most of the reports, context-sensitive charts are displayed. These give you a graphical picture of the process. The chart's data is dependent on the row or column that is currently highlighted.

Peoplesmith Software

Processing Summary

Setup File: d:\dt\getting started.dl
 Date Processed: 4/20/1999
 Elapsed Time: 0 d; 0:00:42
 Records per Hour: 3,608,000

Source Files (4):
 D:\DT\demo\Demo1.dbf
 Data Type: dBase III
 File Type: Regular
 D:\DT\demo\Demo2.dbf
 Data Type: ASCII - BCP
 File Type: Regular
 D:\DT\demo\Demo3.dbf
 Data Type: ASCII: Delimited
 File Type: Regular

Demol
 Data Type: ODBC: DSN+Access RT,SQL=O:\DT\DEMO.MD;DefaultDir=D:\DT;DriverId=25;FIL=MS Access;M
 File Type: Regular

Matchcode (Next) E:3+L2:5+PH:3+POB:10+SR:15+SH:4)

Component	Title	Size	Start	Fuzzy	Short	Swap	1	2
Zip5	All	5	Left	No	Both Empty	No	X	X
Last Name	All	5	Left	No	Both Empty	No	X	X
First Name	All	3	Left	No	Both Empty	No	X	X
PO Box	All	15	Left	No	No	No	X	
Street #	All	5	Left	No	Both Empty	No	X	
Street Name	All	4	Left	No	No	No	X	

1: E:3+L2:5+PH:3+POB:10
 2: E:3+L2:5+PH:3+SR:15+SH:4

Page: 1

File Summary

Source	Input	Processed	Filtered	Output	Dupes	Suppressed	Not Intract
Demol	500	500	0	393	107	0	0
Demol2	500	500	0	36	464	0	0
Demol3	500	500	0	393	107	0	0
Demol4	500	500	0	392	108	0	0
Total	2,000	2,000	0	1,214	776	0	0

Source Code List

Source	1a	2a	3a	4a	5a	6a	7a
Demol	354	39	0	0	0	0	0
Demol2	0	25	13	0	0	0	0
Demol3	339	45	0	0	0	0	0
Demol4	0	245	44	42	0	2	0
Total	692	364	56	42	0	2	0

Source Code File Summary

Source	9a	9b	10a	10b
Demol	0	0	0	0
Demol2	0	0	0	0
Demol3	0	0	0	0
Demol4	0	0	0	0
Total	0	0	0	0

Matchcode Quality

Matchcode	1a	2a	3a	4a	5a	6a	7a
Demol	0	0	0	0	0	0	0
Demol2	0	0	0	0	0	0	0
Demol3	0	0	0	0	0	0	0
Demol4	0	0	0	0	0	0	0
Total	0	0	0	0	0	0	0

Page: 1

A legible, full sized copy of these reports can be found at the back of the Report Library

In addition to these standard printed formats, RTF and various ASCII formats can be printed to file, for importing into Microsoft Word or Excel.

Processing Summary

The **Processing Summary** contains processing statistics and information about your setup.

Processing Statistics

Matchcode Quality | Source Code List | Source Code File Summary

Source Code Dupe Matrix | Source Code Quality Matrix

Processing Summary | File Summary | Dupe Matrix | Quality Matrix | Matchcode Matrix

Setup File: C:\dt\Demos\asdf.dt

Processing Statistics:

Date Processed: 5/13/1999 Average Records per Block: 52
Elapsed Time: 0:00:03 Minimum Records per Block: 2
Total Records Processed: 2,000 Maximum Records per Block: 123
Records per Hour: 2,400,000
Blocks Loaded: 38 Average Key Size per Block: 1
Block Spillovers: 0 Minimum Key Size per Block: 1
Block Overflows: 0 Maximum Key Size per Block: 5

Setup Information:

Ranking: Assign priority or field/Ascending

Output File: C:\dt\Demos\output.Dbf
Duplicate File: C:\dt\Demos\dupes.Dbf
Suppression File: C:\dt\Demos\suppre.Dbf

Matchcode: (HsHldr2)Z:5+POB:10+S#:5+SN:4

Component	Size	Start	Fuzzy	Short/Empty	1	2	3	4	5	6	7	8
Zip5	5	Left	No	Both Empty	X	X						
PO Box	10	Left	No	No	X							
Street #	5	Left	No	Both Empty		X						
Street Name	4	Left	No	No		X						

Close | Print... | Options... | Help

Blocks Loaded

Most of the headings are self-explanatory until we come to "Blocks". Nearly every deduper uses a blocking algorithm. Sometimes it's called "break grouping", "clustering", "partitioning", or "neighborhood sorting", but it's the same thing.

In an ideal deduping world, every record would be compared against every other record. After all, that's the only way you can be absolutely sure. Unfortunately, it's also incredibly impractical because the number of comparisons grows geometrically with the number of records. For a typical mailing list, the number of comparisons would take several lifetimes even on a Cray supercomputer!

The obvious solution is to take some common factor (typically the Zip/Postal Code with mailing lists) which will make the number of comparisons reasonable (and quick).

The older DOS program had a shortcoming called "Block Overflow". This was rarely a problem except when a database was extremely dense so that the computer's memory couldn't hold all the records of one block.

The new Windows version solves this problem with "Spillover Processor" which uses the hard disk in these rare cases. You will never see a "Block Overflow" with Double Take 2, so this number will always be zero.

Of course the hard disk is much slower, so you want to keep "Block Spillovers" to a minimum - otherwise you'll be processing at a snail's pace!

File Summary

The **File Summary** contains a detailed listing of what happened with each file. Counts for Output, Dupe, Suppress, etc. are given, as well as multiple counts.

Processing Statistics

Matchcode Quality | Source Code List | Source Code File Summary

Source Code Dupe Matrix | Source Code Quality Matrix

Processing Summary | **File Summary** | Dupe Matrix | Quality Matrix | Matchcode Matrix

File Summary

File	File Type	Data Type	Total	Processed	Filtered
Demo1	Regular	DBase III	500	500	0
Demo2	Regular	DBase III	500	500	0
Demo3	Regular	DBase III	500	500	0

Demo1:

- Filtered
- Output
- Duplicates
- Suppressions
- Not Intersected

Demo1:

- 1x (Unique)
- 2x
- 3x
- 4x
- 5x
- 6x
- 7x
- 8x
- 9x
- 10x
- 10+x

Multiples

File	1x	2x	3x	4x	5x	6x	7x	8x
Demo1	326	99	0	23	1	0	1	0
Demo2	275	50	49	0	0	0	0	0
Demo3	0	325	0	50	0	0	0	0

Close | Print... | Options... | Help

The leftmost pie chart relates to the highlighted file in the File Summary table. The rightmost pie chart relates to the highlighted file in the Multiples table.

File Summary:

- File** Lists each file that was processed.
- File Type** Indicates the type of file processed: Regular, Suppression, Intersection, No Purge/Single, or No Purge/Global.
- Data Type** Indicates the format of the file processed: DBase III, ASCII, ODBC, etc.
- Total** Total number of records in this file.
- Processed** Number of records processed from this file.
- Filtered** Number of records filtered from this file.
- Dupes** Number of records in this file 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 File.
- Suppress** Number of records in this file that matched against records in a suppression list. Suppression records themselves don't contribute to this count.

No Intersect Number of records in this file that didn't match against any record in any intersection list. Intersection records themselves don't contribute to this count.

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

Multiples:

File Lists each file that was processed.

1x Number of unique records (being unique, they ended up in the Output File).

2x Number of records that were selected as the output record (and ended up in the Output File) 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:

Processed = Output + Not Intersected + Suppressed + Dupes

Total = Processed + Filtered

Output = 1x + 2x + 3x + 4x + 5x + 6x + 7x + 8x + 9x + 10x + 10+x

Total Dupes = 2x + (3x * 2) + (4x * 3) + (5x * 4) + (6x * 5) + (7x * 6) + (8x * 7) + (9x * 8) + (10x * 9)

NOTE: If you have any 10+x counts, the formula can't be used, as we have no way of knowing its multiplier.

Dupe Matrix

The **Dupe Matrix** gives record counts of inter- and intra-file dupes. The pie chart illustrates the highlighted row.

To understand this report, remember (1) it is reporting dupes only (output records are not counted), and (2) the dupes are counted with respect to which file the output record came from. This latter point is the key difference between this report and the Quality Matrix.

Processing Statistics

Matchcode Quality | Source Code List | Source Code File Summary

Source Code Dupe Matrix | Source Code Quality Matrix

Processing Summary | File Summary | Dupe Matrix | Quality Matrix | Matchcode Matrix

Dupe Matrix

File	Demo1	Demo2	Demo3
Demo1	39	15	20
Demo2	0	52	72
Demo3	0	0	45
Total:	39	67	137

Demo1:

Demo1	52.7 %
Demo2	20.3 %
Demo3	27.0 %

Close | Print... | Options... | ? Help

For all records output from Demo1, there were 39 dupes of these records in Demo1 (intra-file dupes), 15 dupes in Demo2 (inter-file dupes), and 20 in Demo3 (also inter-file dupes).

For all records output from Demo2, there were no dupes of these records in Demo1 (inter-file dupes), 52 dupes in Demo2 (intra-file dupes), and 72 in Demo3 (inter-file dupes).

For all records output from Demo3, there were no dupes in Demo1, none in Demo2, and 45 in Demo3 (intra-file dupes).

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 type indicates the output record. 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 Melissas 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 because it's the *Dupe Matrix*.

How the Numbers Add Up:

Each column of numbers add up to the Dupe count for that column's file 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.

Quality Matrix

The **Quality Matrix** gives duplicate counts regardless of where an output record came from. The pie chart illustrates the highlighted row.

Processing Statistics

Matchcode Quality | Source Code List | Source Code File Summary

Source Code Dupe Matrix | Source Code Quality Matrix

Processing Summary | File Summary | Dupe Matrix | **Quality Matrix** | Matchcode Matrix

Quality Matrix

File	Demo1	Demo2	Demo3
Demo1	39	69	22
Demo2	70	52	73
Demo3	22	72	46
Total:	131	193	141

Demo1:

<input type="checkbox"/> Demo1	30.0 %
<input checked="" type="checkbox"/> Demo2	53.1 %
<input type="checkbox"/> Demo3	16.9 %

Close | Print... | Options... | ? Help

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 files. 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 files. This count was "hidden" in the Dupe Matrix because the file ranking caused all of these matches to accumulate into the Demo1 or Demo2 rows (because Demo1 was ranked over Demo2 which was ranked 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 Melissas 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

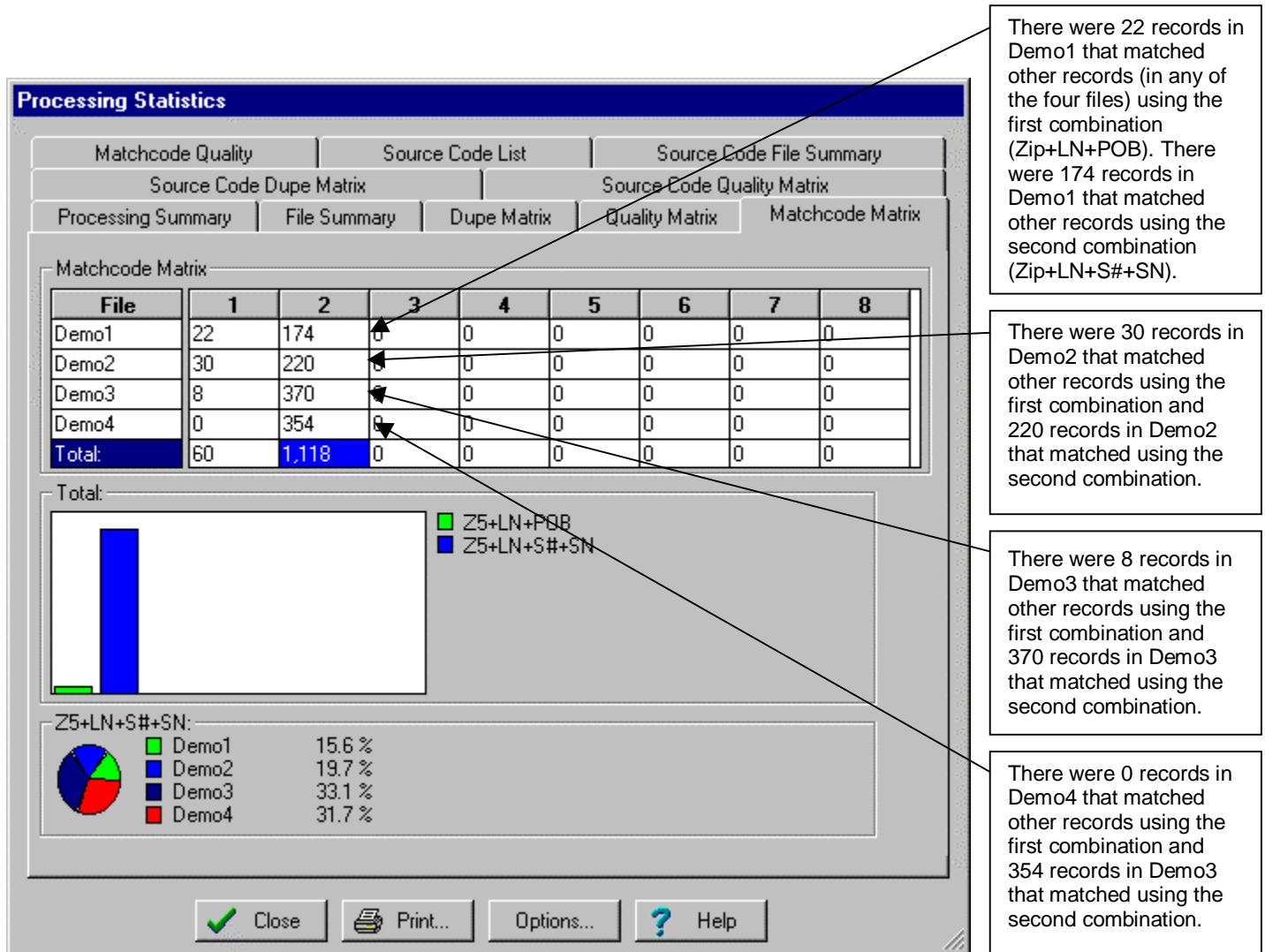
Basically, you look at each unique record in a file and dedupe it against everything else. The unique record does not get figured into the counts (just duplicates of it). Using Brand X software, you would have to run 3 dedupe runs, assigning different priorities each time: first to Demo1, then to Demo2, 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. Please don't call us and ask us if we've "fixed" this report, or made some sort of scientific breakthrough - the numbers don't add up because of what they are; there's nothing wrong with your TV.

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.

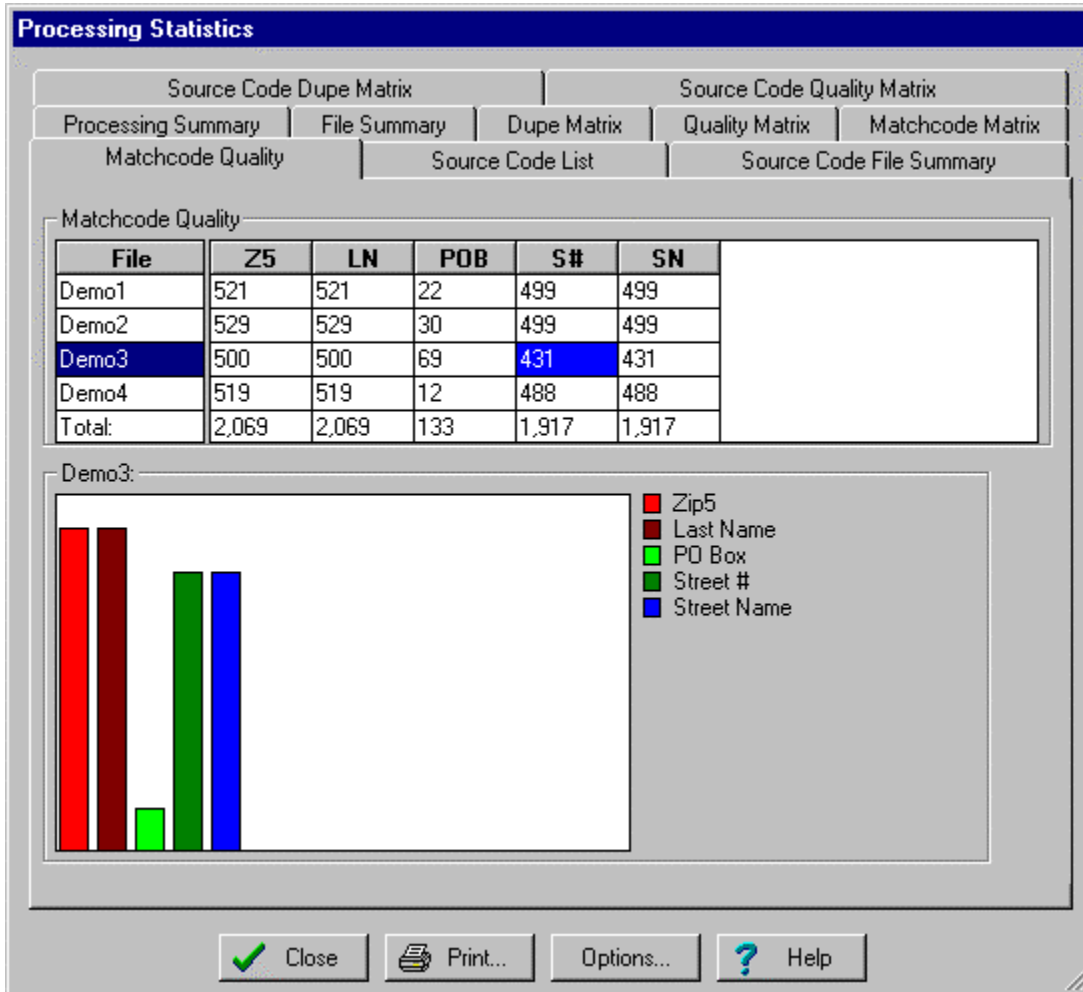


Note that the output records are not used in these calculations. Also, if matches are made using both combinations, both columns' counts will reflect that match.

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 fewer hits than the Street Address combination - after all, most folks prefer mail delivered to their door!

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 file, 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.

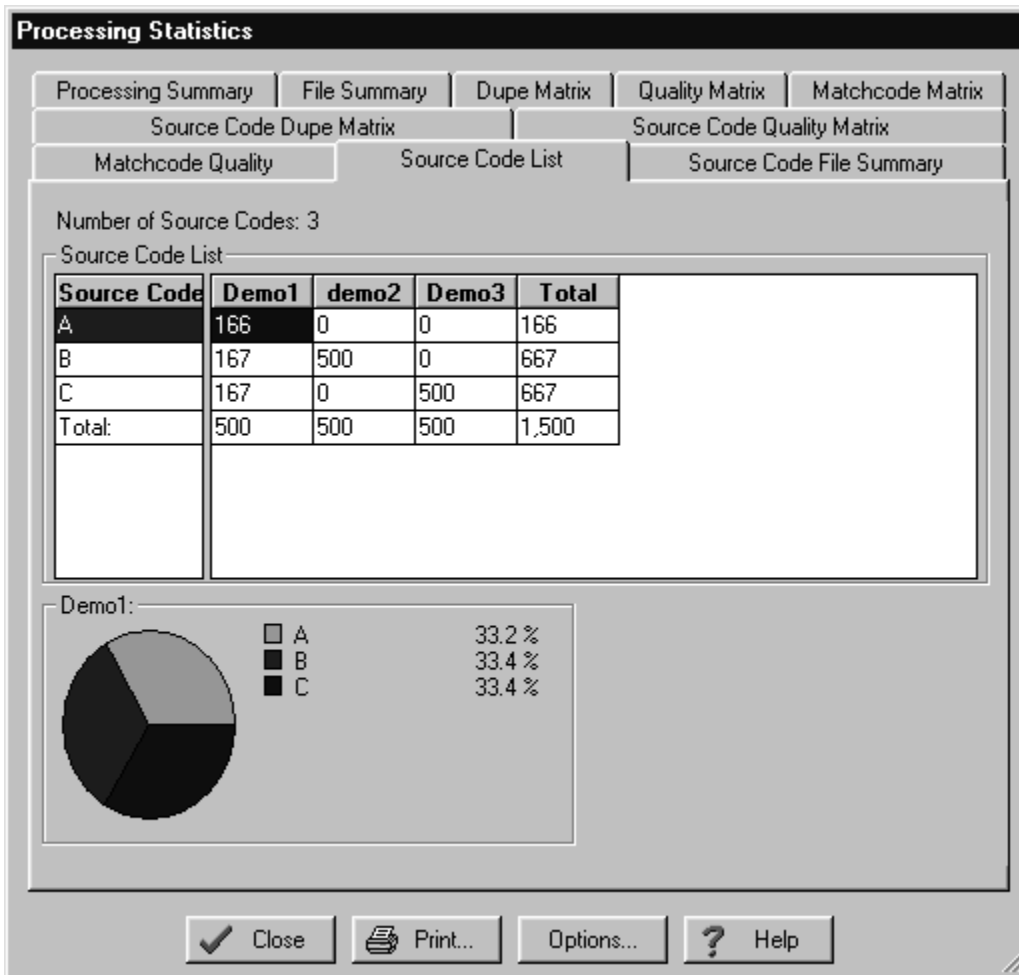
How the Numbers Add Up:

You would love to see each count match the number of records processed for that file. 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*.

Source Code List

This report is generated whenever you have specified at least one Source Code Field. For each source code detected in the process, a count is accumulated for each file. The pie chart illustrates the highlighted column.



But, you ask, what about the times when I've only specified a Source Code Field for 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).

How the Numbers Add Up:

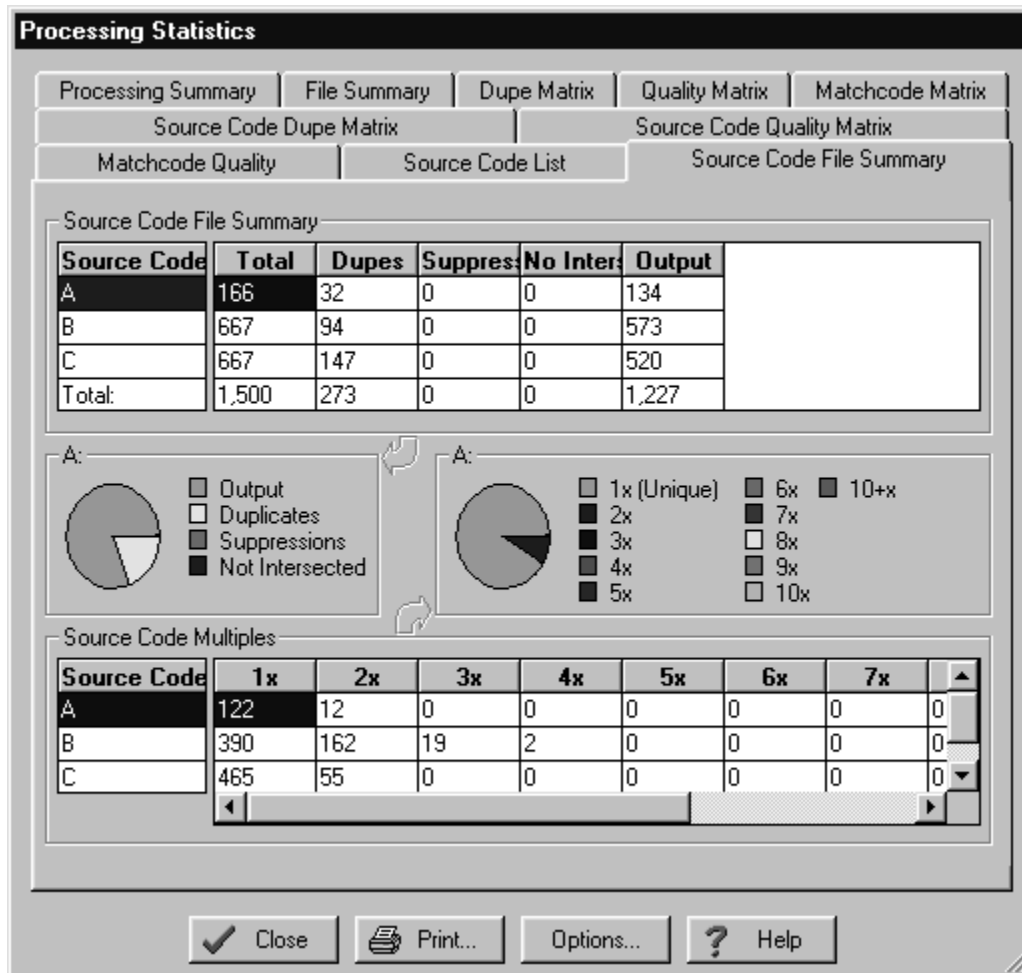
Total Source Codes for a File = Records Processed for that File

$$e=mc^2$$

The second formula will only work in the currently highlighted universe though (and may not apply to some suppression runs).

Source Code File Summary

The **Source Code File Summary** consists of a detailed listing of what happened during the process, broken down 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.



The leftmost pie chart relates to the highlighted file in the File Summary table. The rightmost pie chart relates to the highlighted file in the Multiples table.

File Summary:

- Source Code** Lists each source code that was processed.
- Total** Total number of records with this source code.
- Dupes** Number of records with this source code 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 File.
- Suppress** Number of records with this source code that matched against records in a suppression list. Suppression records themselves don't contribute to this count.
- No Intersect** 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.
- Output** Number of records with this source code that were unique or selected as the output record in a group of duplicates.

Why no File Type and Data Type columns? Because source codes aren't file based - they can be scattered across many files. 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.

Multiples:

- Source Code** Lists each source code that was processed.
- 1x** Number of unique records (being unique, they ended up in the Output File).
- 2x** Number of records that were selected as the output record (and ended up in the Output File) 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:

$$\text{Total} = \text{Output} + \text{No Intersect} + \text{Suppress} + \text{Dupe}$$

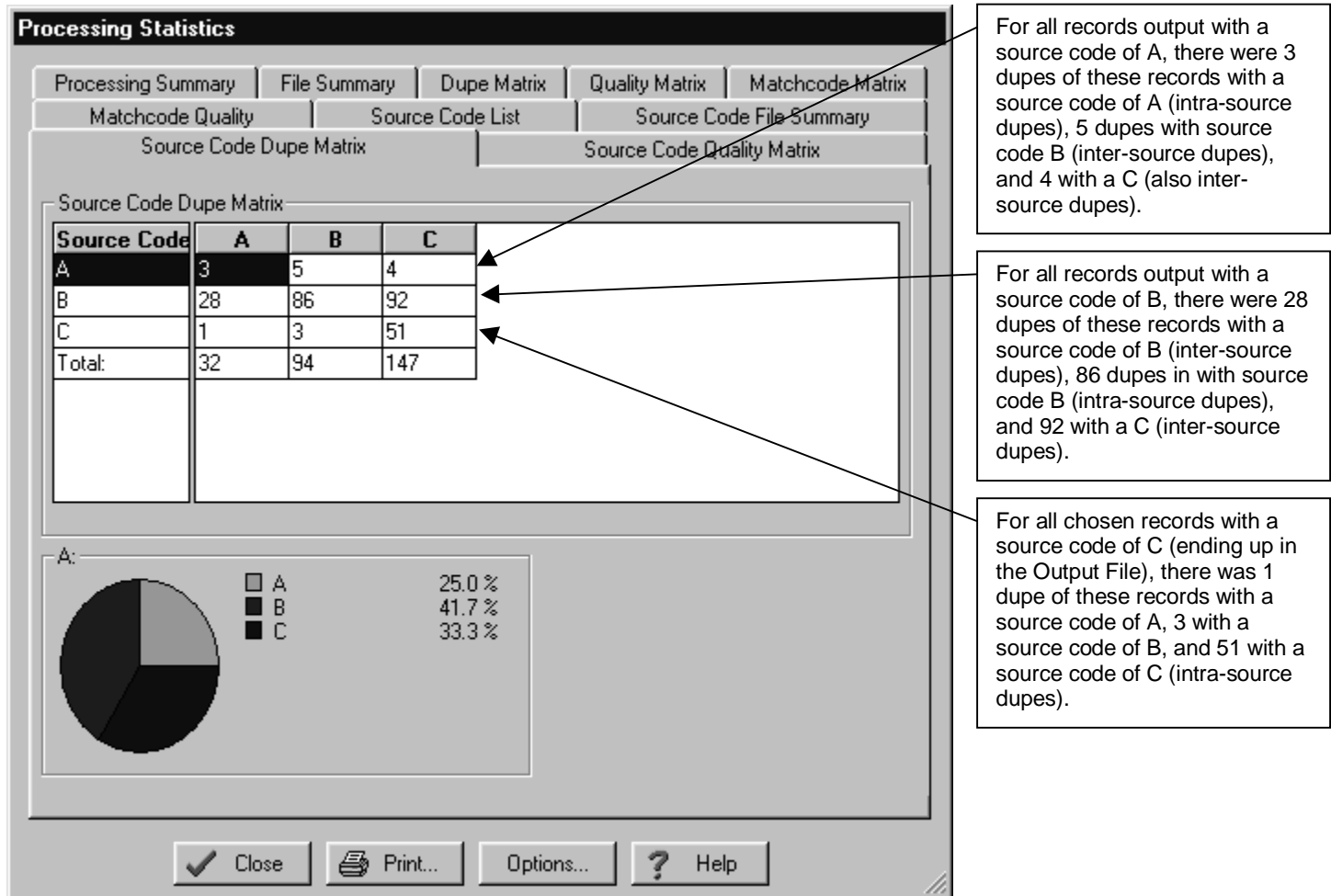
$$\text{Output} = 1x + 2x + 3x + 4x + 5x + 6x + 7x + 8x + 9x + 10x + 10+x$$

$$\text{Total Dupes} = 2x + (3x * 2) + (4x * 3) + (5x * 4) + (6x * 5) + (7x * 6) + (8x * 7) + (9x * 8) + (10x * 9)$$

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 Code 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 files.



This report is generated whenever you have specified at least one Source Code.

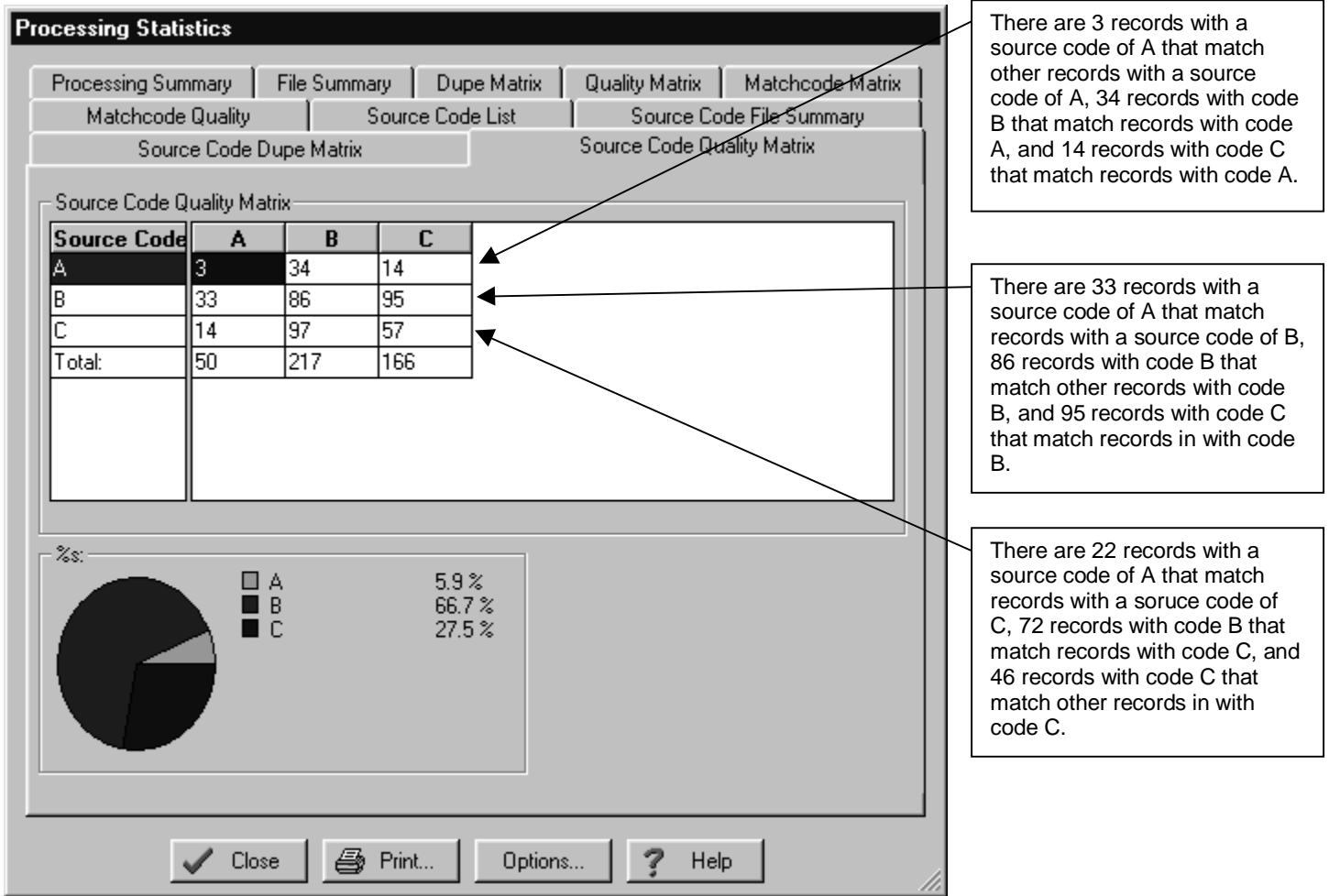
The important thing to remember about this report is that all dupes are counted with respect to which record was output. This is the key difference between this report and the Quality Matrix. The cells where a file intersects itself (the diagonal running from top left to bottom right) are commonly known as the intra-list counts, whereas the others are known as inter-list counts ("intra" meaning "within" and "inter" meaning "between")

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.

Source Code 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 files.

How the Numbers Add Up:

Like the Quality Matrix, the Source Code Quality Matrix does not produce counts that will "jive" with any other numbers. They're just not supposed to.