Unit objectives

- After completing this unit, you should be able to:
  - Describe the Standardize stage in the Data Re-engineering Methodology
  - Identify rule sets
  - Apply the Standardize stage
  - Interpret standardization results
  - Investigate unhandled data and patterns

Standardize

- Transformation
  - Parsing free form fields
  - Comparison threshold for classifying like words
  - Bucketing data tokens
- Standardization
  - Applying standard values and standard formats
- Phonetic Coding for use in Matching
  - NYSIIS
  - Soundex

- decompose free-form fields into single component fields
- assign data to its appropriate metadata fields

Standardize example

Input File:

Address Line 1	Address Line 2
1721 W ELFINDALE ST	UNIT 20
1721 W ELFINDALE ST # 20	
16200 VENTURA BOULEVARD	SUITE 201
C/O JOSEPH C REIFF	12 WESTERN AVE
1705 W St	PHILADELPHIA
1655 PONCE DE LEON AVENUE	15TH FLOOR

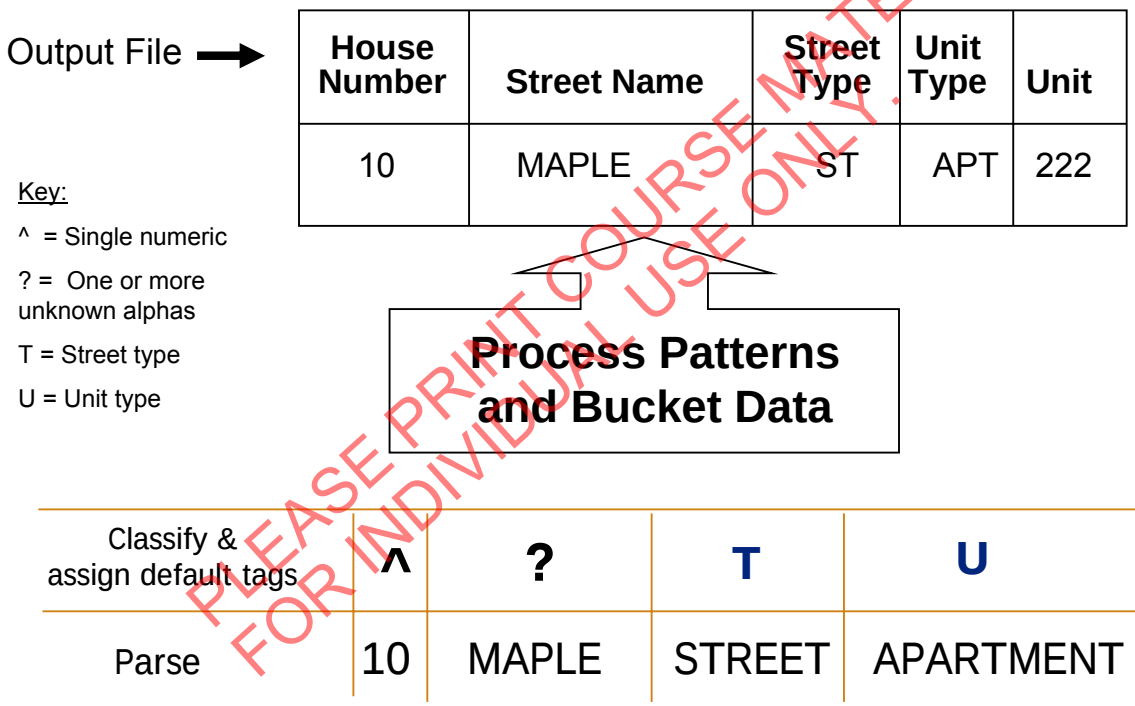
Result File:

House #	Dir	Str. Name	Type	Unit Type	Unit. Value	Floor Type	Floor Value	NYSIIS	Soundex	City
1721	W	ELFINDALE	AVE	UNIT	20					
1721	W	ELFINDALE	ST		20					
16200		VENTURA	BLVD							
12		WESTERN	AVE							
1705		W	ST							PHILADELPHIA
1655		PONCE DE LEON	AVE			FLOOR	15			

Notice:

- Unit 20 & # 20, both bucketed as unit type and value
- C/O Joseph C Reiff, moved to another field (can't see on screen)
- 12 Western Ave, recognized as an address

Standardize process



Notice the “standardization” of St & APT

1. First we parse the data
2. Classify known words (classification table)
3. Apply general (default tags) to unclassified tokens
4. Create new output fields (dictionary file)
5. Process the patterns (Pattern action File) to move data into correct field and apply standard values and formats.

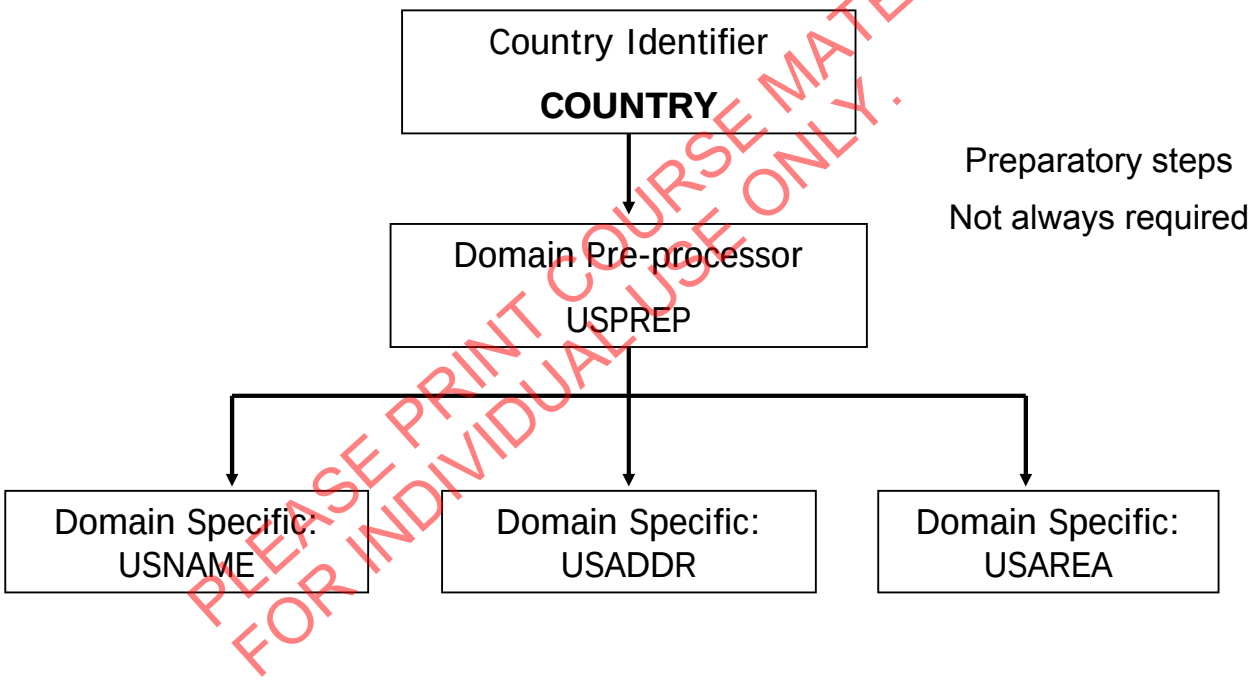
Standardize stage

- Standardize Stage
  - Uses Rule sets for:
    - Country processing
    - Pre-domain processing
      - USPREP
    - Domain processing
      - USADDR
      - USAREA
      - USNAME
    - Multi-national Address
    - WAVES

- Identify new fields based on underlying data
- Examples:
- set a gender flag
- Name type flag - identify individual address from an organization address. I=Individual, O=Organization
- Address type flag – S=Street address, B= box address, R=rural route address, O=other type of address
- Transformation rules are created both for matching and creating the load file (sometimes these rules are different)

WAVES (World-wide address verification) performs address validation against a database

Types of rule sets



Three levels of rule sets

- Country Identifier - Identify the country and append ISO country code based on address format
- Domain Pre-processing – separates mixed domain fields into specific domain fields
- Domain Specific

Example: country identifier

Input Record

100 SUMMER STREET 15TH FLOOR BOSTON, MA 02111
SITE 6 COMP 10 RR 8 STN MAIN MILLARVILLE AB T0L 1K0
28 GROSVENOR STREET LONDON W1X 9FE
123 MAIN STREET

Output Record

US Y 100 SUMMER STREET 15TH FLOOR BOSTON, MA 02111
CA Y SITE 6 COMP 10 RR 8 STN MAIN MILLARVILLE AB T0L 1K0
GB Y 28 GROSVENOR STREET LONDON W1X 9FE
US N 123 MAIN STREET

Note: US default is assumed using ZQUSZQ country code delimiter

ISO Country Code
Indicator Flag (Y or N, where Y = country code verified)

The format of the country code delimiter is:

- ZQ<ISO Country Code>ZQ

For example, the country code delimiter for the United States would be:

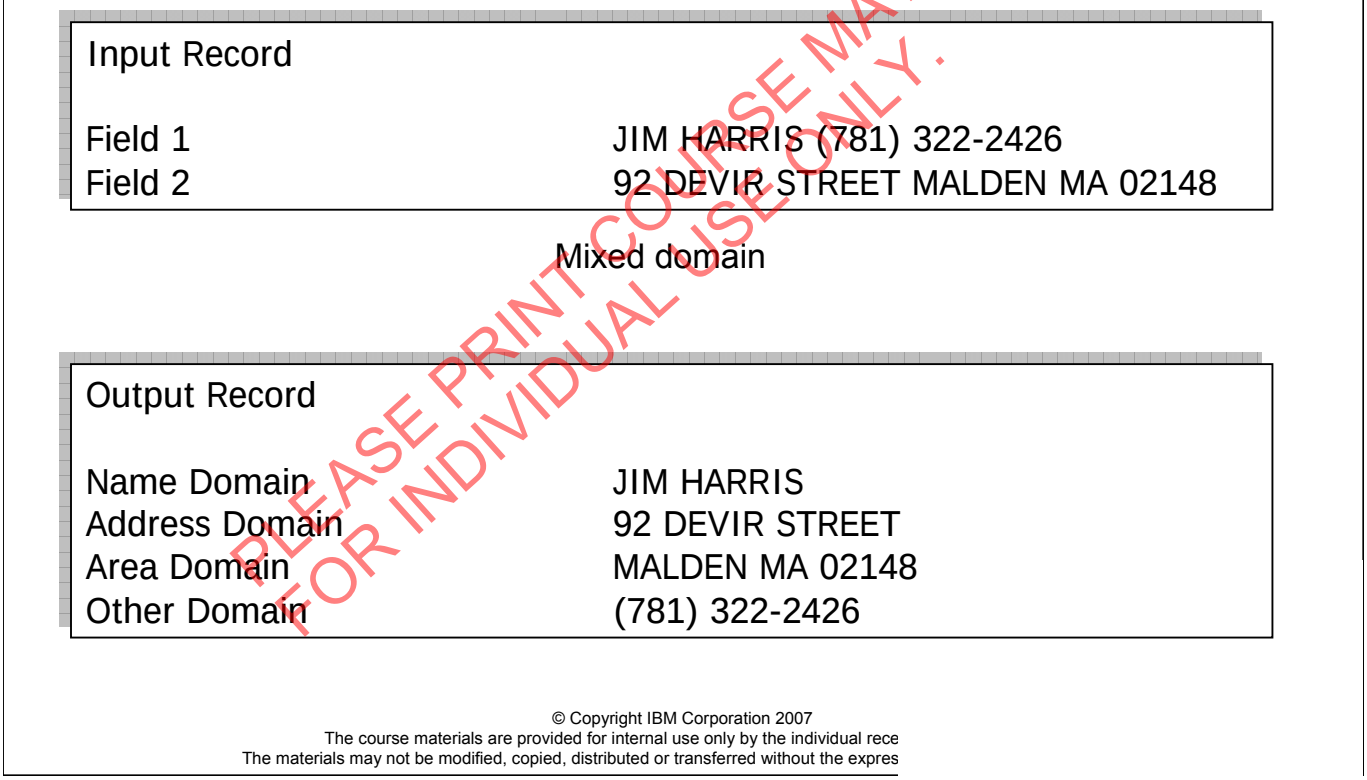
- ZQUSZQ

Allows easy processing of multi-national data

Assigns an ISO country code to each record

When the Country Identifier Rule Set can not determine the country code, the default value will be taken from the country code delimiter

Example: domain preprocessor



Categorize input data into one of the following domain-specific column sets:

- Name - individual and organization names, attention instructions, and secondary names
- Address - low level geography including street, rural, box, unit, and building addresses
- Area - high level geography including city name, postal code, and country code
- Other - non-name and non-geography data

The metadata delimiters indicate what kind of information you are expecting to find within the fields of your input data

If the pre-processor can not determine the domain of a token, it will be defaulted based on its metadata delimiter

The format of the metadata delimiter is:

- ZQ<Domain>ZQ

There are four accepted metadata delimiters:

- ZQNAMEZQ - Name delimiter
- ZQADDRZQ - Address delimiter
- ZQAREAZQ - Area delimiter
- ZQOTHRZQ - Other delimiter

Example: domain specific

Input Record

100 SUMMER STREET 15TH FLOOR

Output Record

House Number	100
Street Name	SUMMER
Street Suffix Type	ST
Floor Type	FL
Floor Value	15
Address Type	S
NYSIIS of Street Name	SANAR
Reverse Soundex of Street Name	R520
Input Pattern	^+T>U

Evaluate domain-specific input

Generate business intelligence fields:

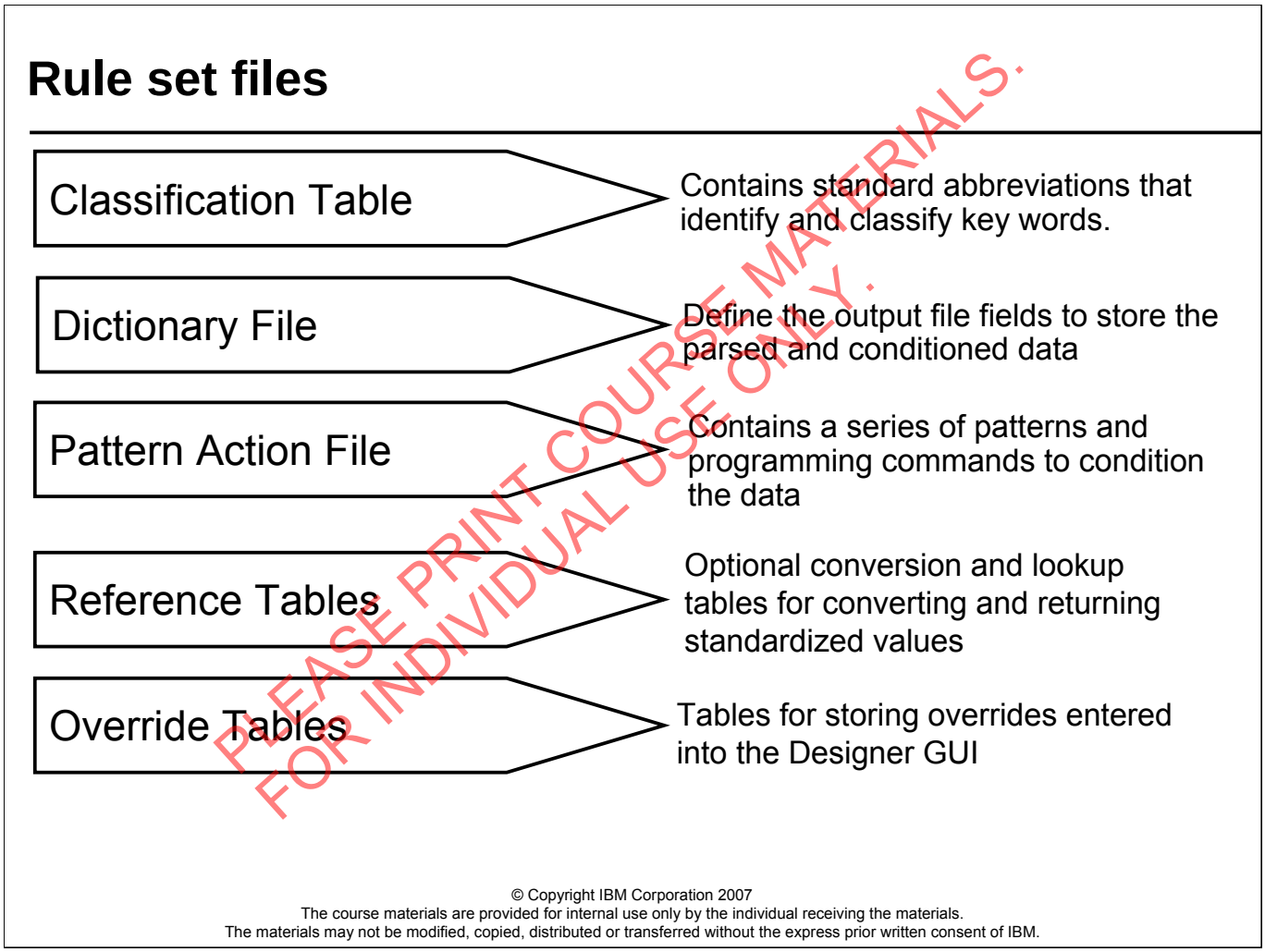
- Create all subordinate domain elements needed for data storage and presentation
- Apply consistent representations to data
- Incorporate applicable standards such as postal standards for addresses

Generate matching fields:

- Blocking keys
- Primary match keys

Rule sets

- Rule sets contain logic for:
  - Parsing
  - Classifying
  - Processing data by pattern and bucketing data
- Three required files
  - Classification Table
  - Dictionary File
  - Pattern Action File
- Optional files
  - Lookup tables
  - Override tables



The first three files are required for each rule set.

If the rule set name is USNAME, then the classification table is named USNAME.CLS, USNAME.PAT, USNAME.DCT

Rule sets can be copied.

Dictionary File: Defines two-character field abbreviations that are used in the output file for a particular rule set.. Each field is referenced in the PAT file to determine where individual tokens will be bucketed.

First Name Lookup table: Applies an enhanced first name, e.g. Barbara to Barb or Barbie, Kenneth to Ken, Kathleen to Kathy.

Example:

Classification table ST. is classified as a street type (T) with a standard value of (ST)

Classification table

- Contains the words for classification, standardized versions of words, and data class
- Data class (data tag) is assigned to each data token
- Default classes are the same across all rule sets
- User-defined classes are assigned in the classification table
  - Users may modify, add or delete these classes
  - User-defined classes are a single letter

Default classes

Class	Description
^	A single numeric
+	A single unclassified alpha (word)
?	One or more consecutive unclassified alphas
@	Complex mixed token, e.g., C3PO
>	Leading numeric, e.g., 6A
<	Trailing numeric, e.g. A6
Zero (0)	Null class

Default classes are the same across all rule sets.

If any of the characters in the class column actually occur in the data they are represented by a ~. A ~ also represents itself.

Null does not mean database null. It means the PAT file will delete the string.

User-defined classes

Class	Description
USNAME	
G	Generational, e.g., Senior, I, II
P	Prefix, e.g. Dr., Mr., Miss
USADDR	
T	Street Type
D	Directional
B	Box Type
USAREA	
S	State Abbreviation

User defined classes are specific to the rule set.

Classification table example

```
; USADDR Classification Table
;-------------------------------------------------------------------
  ----------
; Classification Legend
;-------------------------------------------------------------------
  ----------
; B - Box Types
; D - Directionals
; F - Floor Types
; H - Highway Modifiers
; R - Rural Route, Highway Contract, Star Route
; T - Street Types
; U - Unit Types
;-------------------------------------------------------------------
  ----------
; Table Sort Order: 51-51 Ascending, 26-50 Ascending, 1-25 Ascending
;-------------------------------------------------------------------
  ----------
DRAW                      "PO BOX"                  B
DRAWER                    "PO BOX"                  B
PO                        "PO BOX"                  B
POB                       "PO BOX"                  B
POBOX                     "PO BOX"                  B
POBX                      "PO BOX"                  B
PODRAWER                  "PO BOX"                  B
```

Token

Standard form

Classification

Comparison threshold

- May be used in the Classification table
- Used to efficiently make entries into the classification table
- Helps overcome spelling and data entry errors
- Not required
- Threshold uses a logical string comparator

Threshold level	
900	Exact match
850	Almost certainly the same
800	Most likely equivalent
750	Most likely not the same
700	Almost certainly not the same

Two passes are made through the classification table:

First pass looks for an exact match

Second pass looks for a fuzzy match based on the threshold level

Classification table example with comparison threshold

```
; USADDR Classification Table
;-------------------------------------------------------------------------
  ---------
; Classification Legend
;-------------------------------------------------------------------------
  ---------
; B - Box Types
; D - Directionals
; F - Floor Types
; H - Highway Modifiers
; R - Rural Route, Highway Contract, Star Route
; T - Street Types
; U - Unit Types
;-------------------------------------------------------------------------
  ---------
; Table Sort Order: 51-51 Ascending, 26-50 Ascending, 1-25 Ascending
;-------------------------------------------------------------------------
  ---------
DRAW                      "PO BOX"                  B
DRAWER                    "PO BOX"                  B
.....................................................................
NORTHEAST                 NE                        D   850
NORTHWEST                 NW                        D   850
NW                        NW                        D
S                         S                         D
SO                        S                         D
SOUTH                     S                         D
```


850 – almost certainly the same

Dictionary file

- Defines the field definitions for the output file
- When data is moved to these output fields it is called "bucketing" the data
- The order that the fields are listed in the dictionary file defines the order the fields appear in the output file
- Dictionary file entries are similar to field definitions

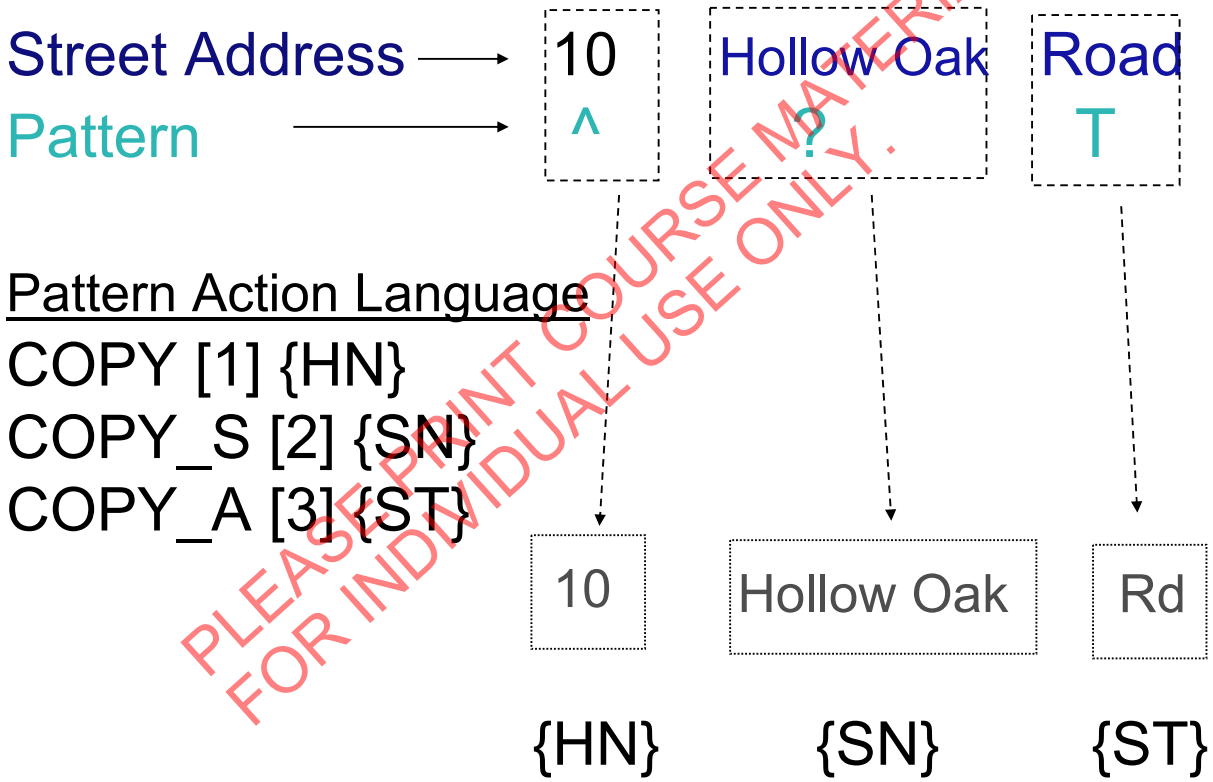
Dictionary file example

```
;;QualityStage v8.0
\FORMAT\ SORT=N
;-------------------------------------------------------------------------
; USADDR Dictionary File
;-------------------------------------------------------------------------
; Total Dictionary Length = 411
;-------------------------------------------------------------------------
; Business Intelligence Fields
;-------------------------------------------------------------------------
HouseNumber C 10 S HouseNumber ;0001-0010
HouseNumberSuffix C 10 S HouseNumberSuffix ;0011-0020
StreetPrefixDirectional C 3 S StreetPrefixDirectional ;0021-0023
StreetPrefixType C 20 S StreetPrefixType ;0024-0043
StreetName C 25 S StreetName ;0044-0068
StreetSuffixType C 5 S StreetSuffixType ;0069-0073
StreetSuffixQualifier C 5 S StreetSuffixQualifier ;0074-0078
```

Pattern-Action file

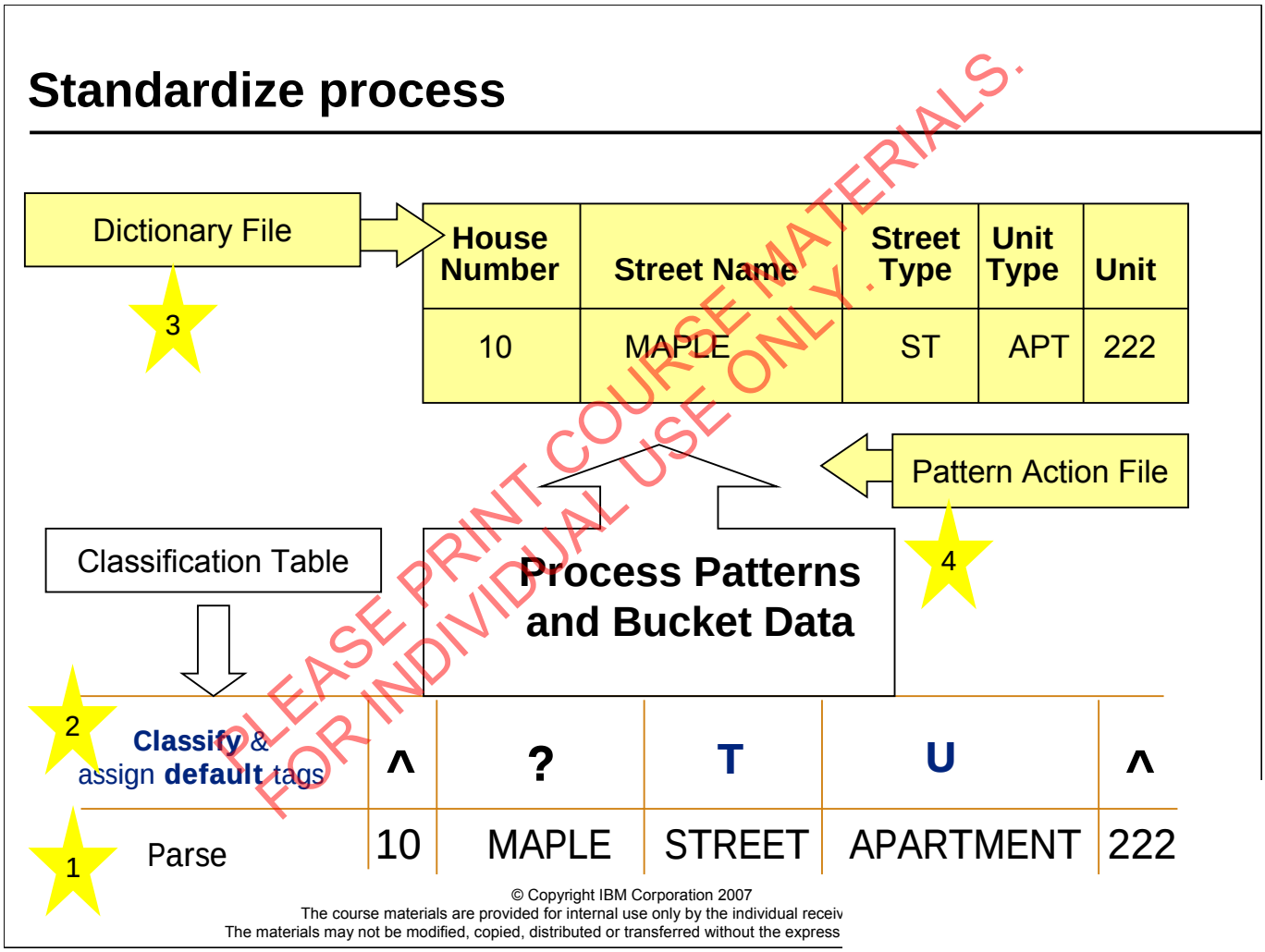
- Contains the rules for standardization; that is, the actions to execute with a given pattern of tokens
- Records are processed from the top down
- Written in Pattern Action Language (PAL)
- Complex parsing can be coded in this file

Pattern Action file process



Optional lookup tables

- Called from the Pattern Action File
- Rule sets may contain lookup tables such as:
  - Common First Names and Enhanced First Names
    - Barb & Barbara
    - Ted & Edward
  - Gender based on name
  - State abbreviations
  - Common city abbreviations
    - NYC = New York City
    - LA = Los Angeles



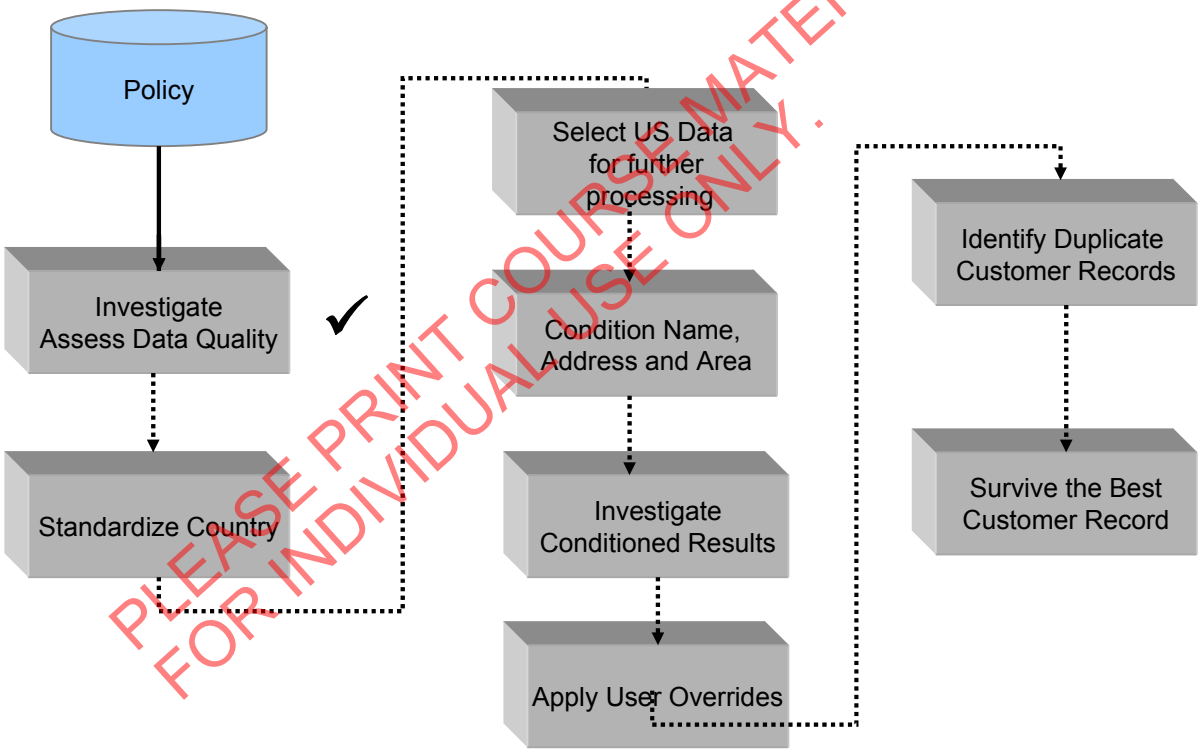
1. (Pattern Action file) parses the data
2. (classification table) Classifies known words
3. (dictionary file) Defines new output fields
4. (Pattern action File) Processes the patterns to move data into correct field and applies standard values and formats.

Standardizing international data

- Two methods
  - Method 1: Use country pre-processor, domain pre-processor, and domain-specific rules
    - Uses out-of-the-box, included functionality/rules
  - Method 2: Use Multinational Standardize and WAVES
    - Requires purchase of WAVES database

WAVES first uses the Multinational standardization, then validates the result against a database.

Course exercise project design

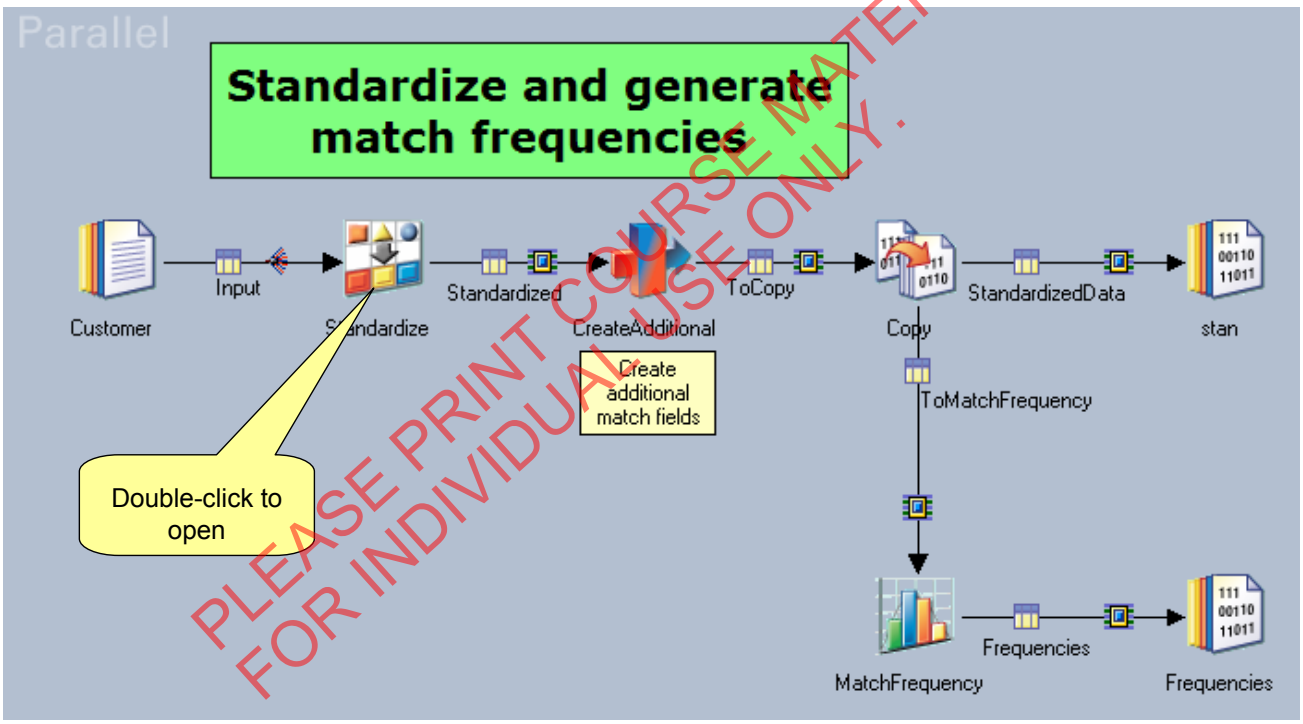


Country rule set

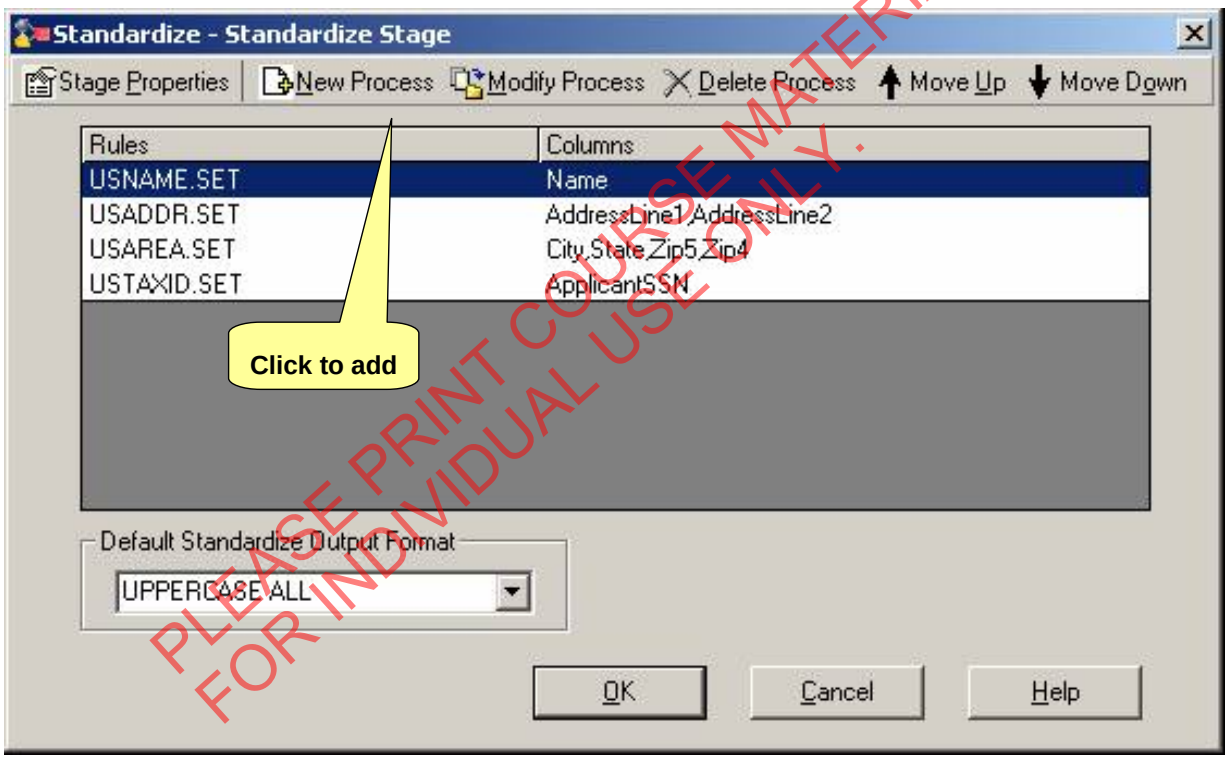
- Country Rule set appends the two byte ISO country code
- Input to the country rule set includes:
  - Street Address
  - City or locality
  - State
  - Zip or Postal code
  - Country field (if it exists)
- Output:
  - Two byte ISO country code
  - Flag identifying explicit or default decision

Standardization implementation

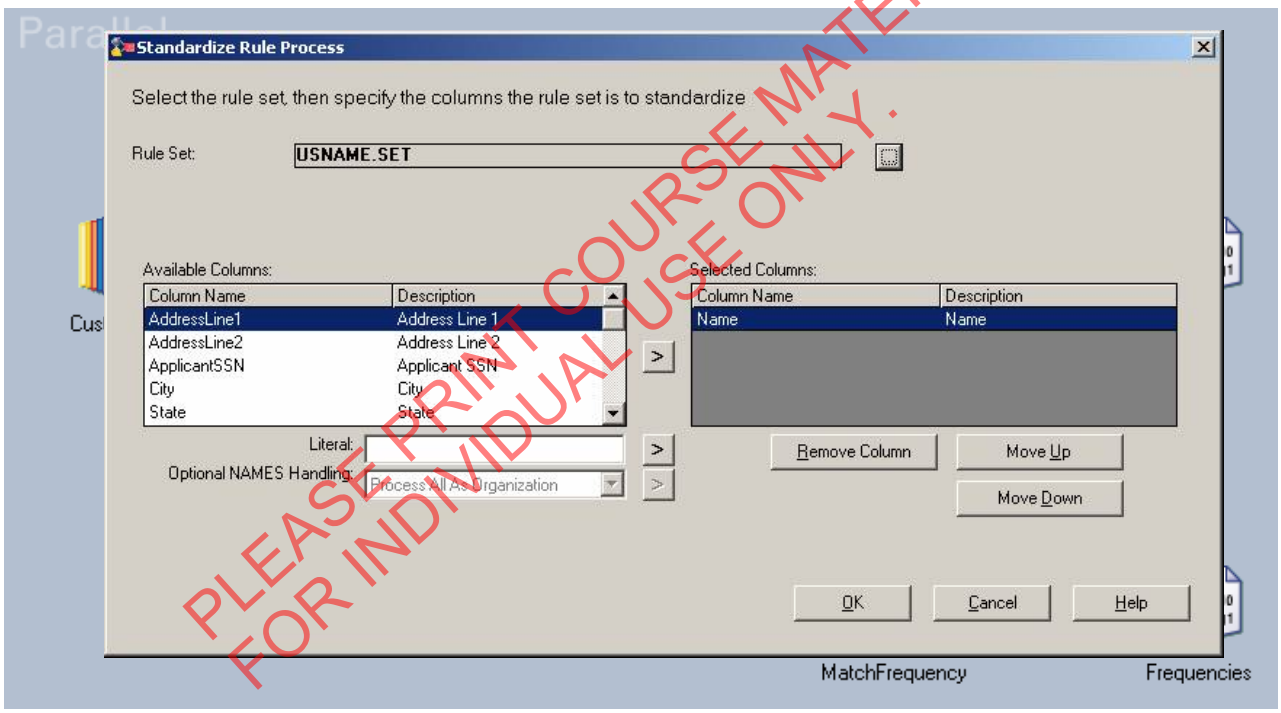
Standardization



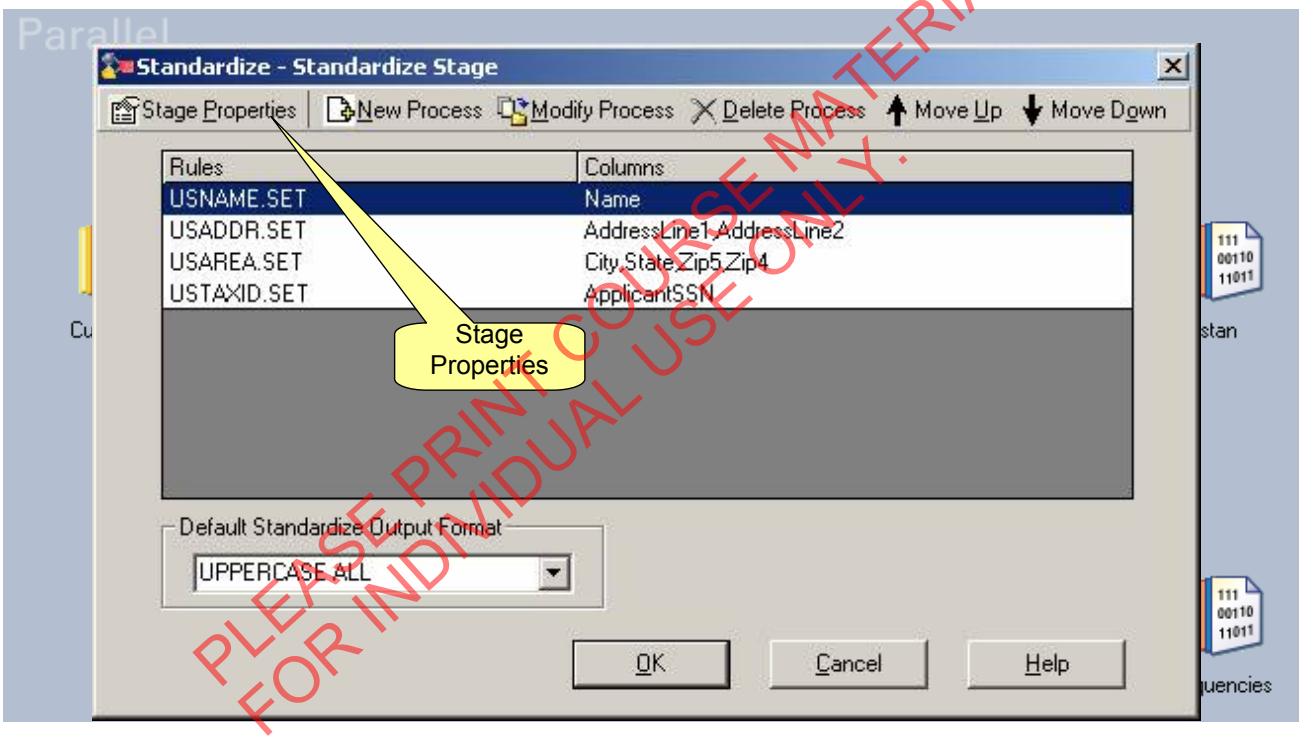
Standardization



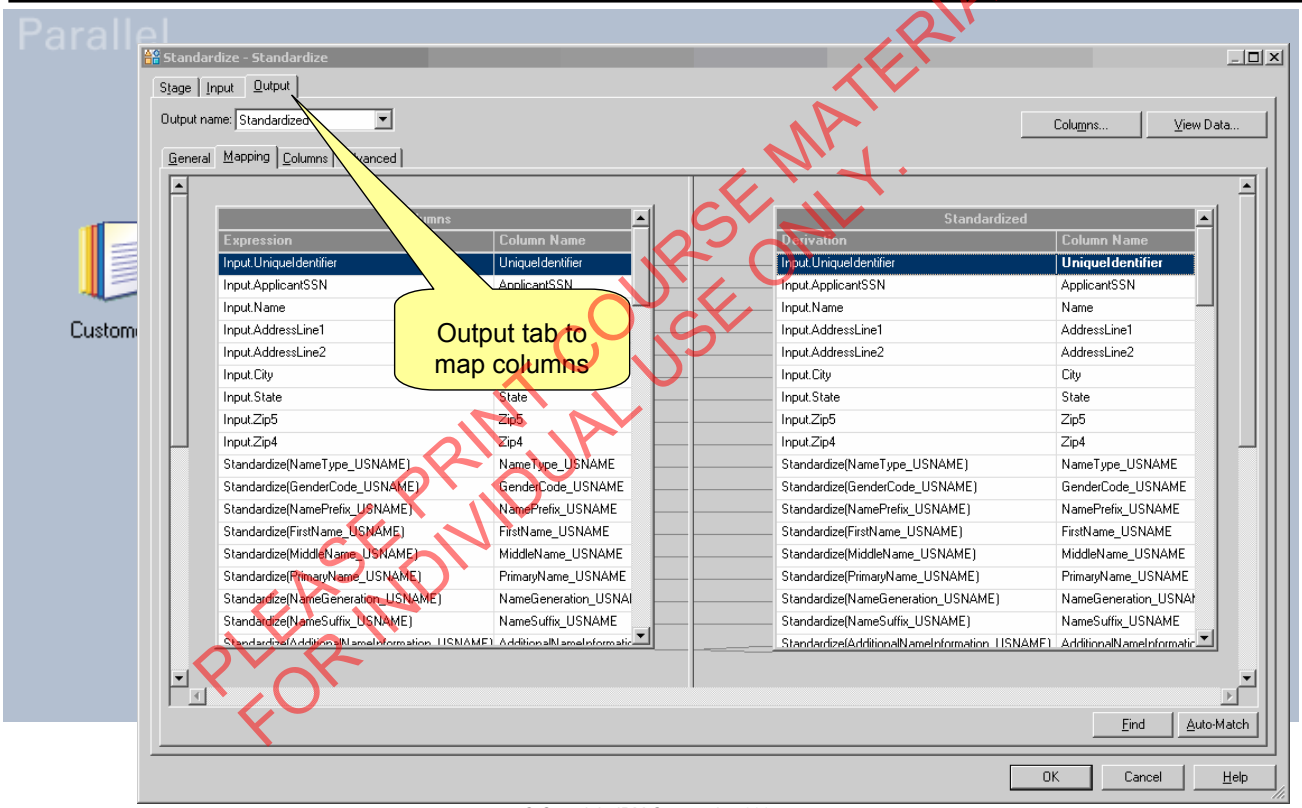
Standardization - Word



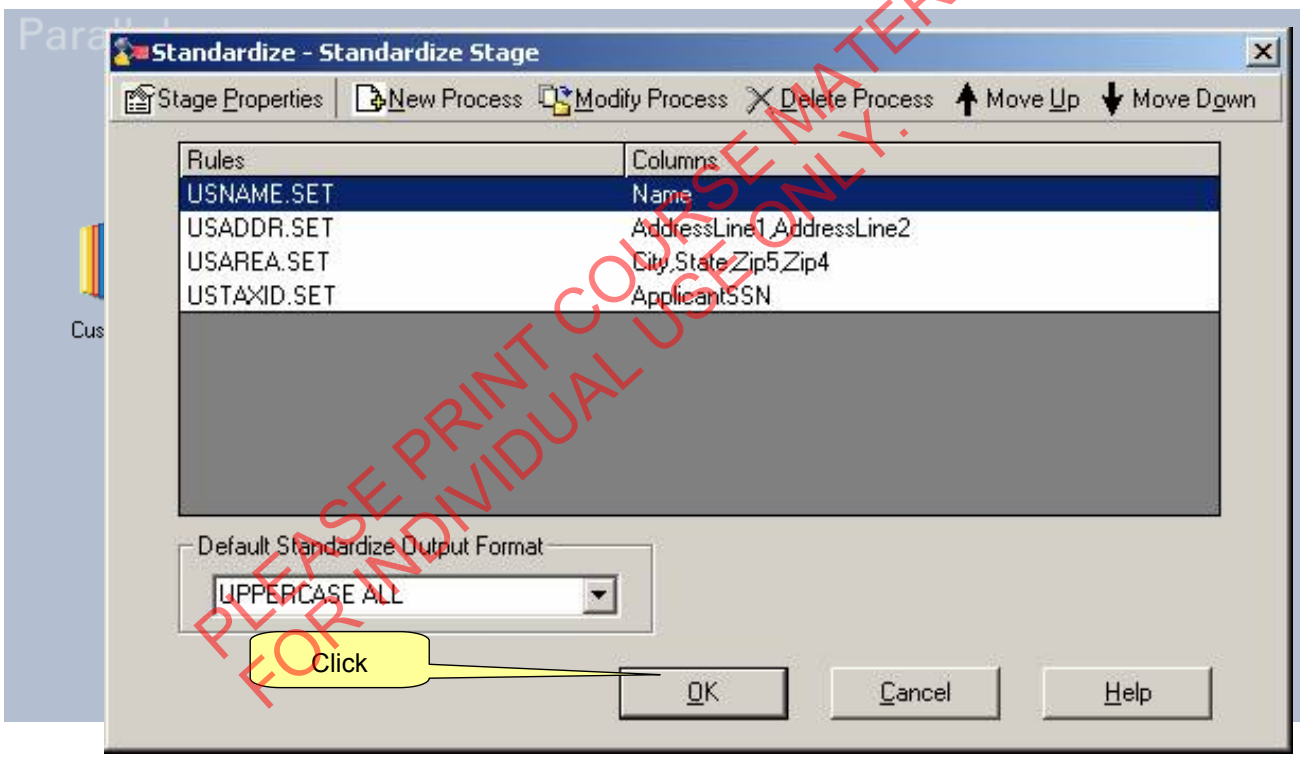
Standardization



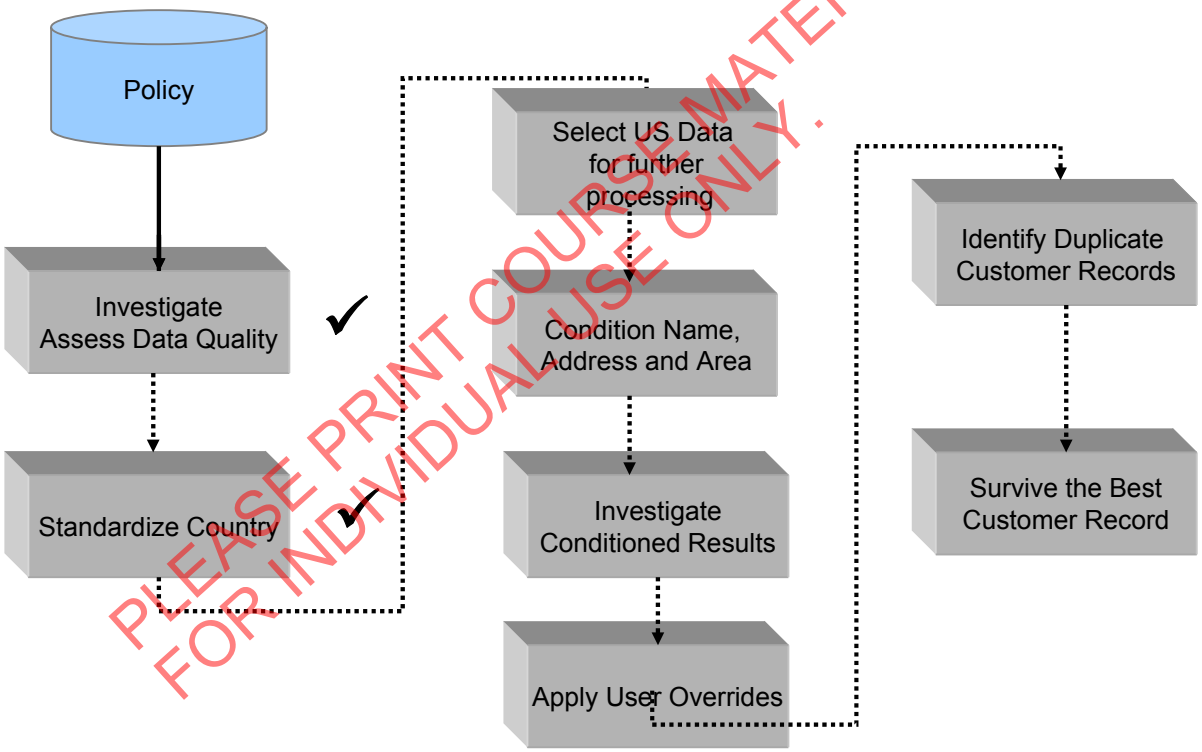
Standardization



Standardization

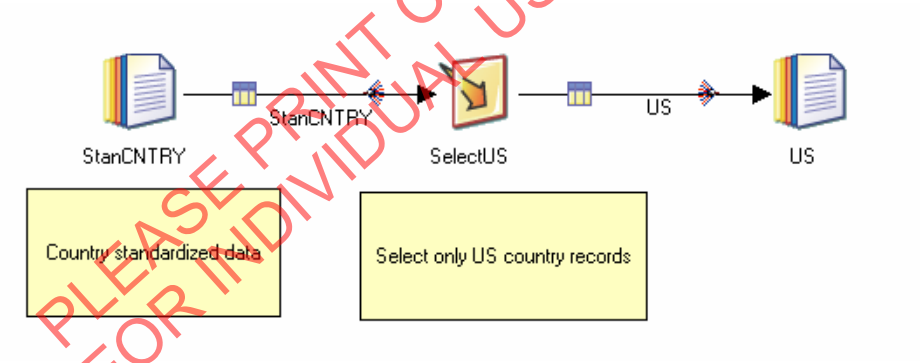


Course exercise project design

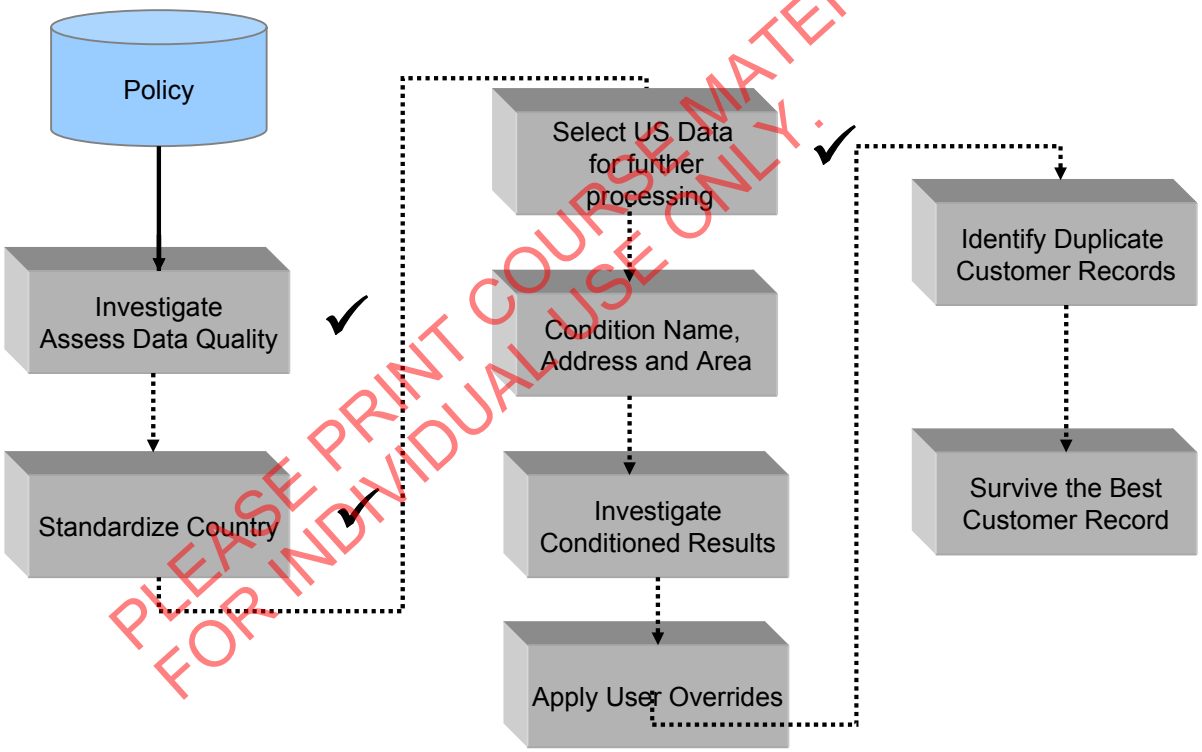


Selecting US data

- The DataStage Filter Stage provides the capability of selecting and/or rejecting records based on a set of values for a field
- Selecting or splitting data requiring compound or complex logic may require Transformer stage



Course exercise project design



Domain pre-processor rule sets

- Pre-processor rule sets are designed to filter name, street address and area (city, state, zip) data
  - For example, if the city, state and zip is found in ADDRESS LINE 2, the pre-processor rule set will attempt to recognize this data and move it into the area domain
- The pre-processor rule set prepares the data for processing by domain specific rule sets

Domain rule sets

- Domain rule sets expect only data for that domain as the input
- Domain rule sets that come with QualityStage are:
  - Name
  - Street address
  - Area (city, state and zip)

USNAME rule set

- The USNAME rule set works on both personal names and organization names for US data
- Data is parsed into name components
- Phonetic coding of the First Name and Primary Name are created for matching

See USNAME.DCT file for all parsed names (PUT COPY OF USNAME.DCT, CLS KEY, PAT PARSING REQ. IN SG)

A “Name Type” Flag is applied to help identify individual names vs. organization names which may require different match strategies

Data is parsed into individual name fields

USADDR rule set

- This rule set is applied to street address fields
- The “Address Type” flag identifies different types of addresses
  - ‘S’ Street address
  - ‘B’ Box address
  - ‘R’ Rural route address
- Phonetic coding of the Street Name is created for matching

Note: (PUT COPY OF USADDR.DCT & CLS KEY & PARSING REQ FROM PAT FILE IN SG)

Address Type Flag: S=street address (street name is populated), B=Box address (no street name but a box type is populated), R=Rural Route (no street name, no box type, butr the rr type is populated).

Different types of addresses may require different match strategies.

Data is parsed into individual street address fields

USAREA rule set

- This rule set is applied to city, state and postal code fields
- Data is parsed into city name, state abbreviation, zip code and zip plus four
- Phonetic coding of the city name is created for matching

Standardize results

- Business Intelligence fields
  - Parsed from the original data, they may be used in matching and generally they are moved to the target system
- Matching Fields
  - Generally these fields are created to help during the match process and are dropped after successful matching
- Reporting fields
  - Specifically created to help review results of Standardize and recognized handled and unhandled data

Field types are documented in the dictionary file of the rule set.

Business intelligence fields

- Intelligent data parsed and bucketed from the input free-form field

USNAME Examples	USADDR Examples	USAREA Examples
Title	HouseNumber	City
First Name	Directional	State
Middle Name	Street Name	Zip5
Primary Name	Unit Types	Zip4
Generational	Box Types	
	Unit Values	
	Building Names	

Fields are parsed from the source data.

Standardize matching fields

- Phonetic coding
  - NYSIIS
  - Reverse NYSIIS
  - Soundex
  - Reverse Soundex
- Hash keys
  - First 2 characters of the first five words
- Packed Keys
  - Data concatenated, or packed

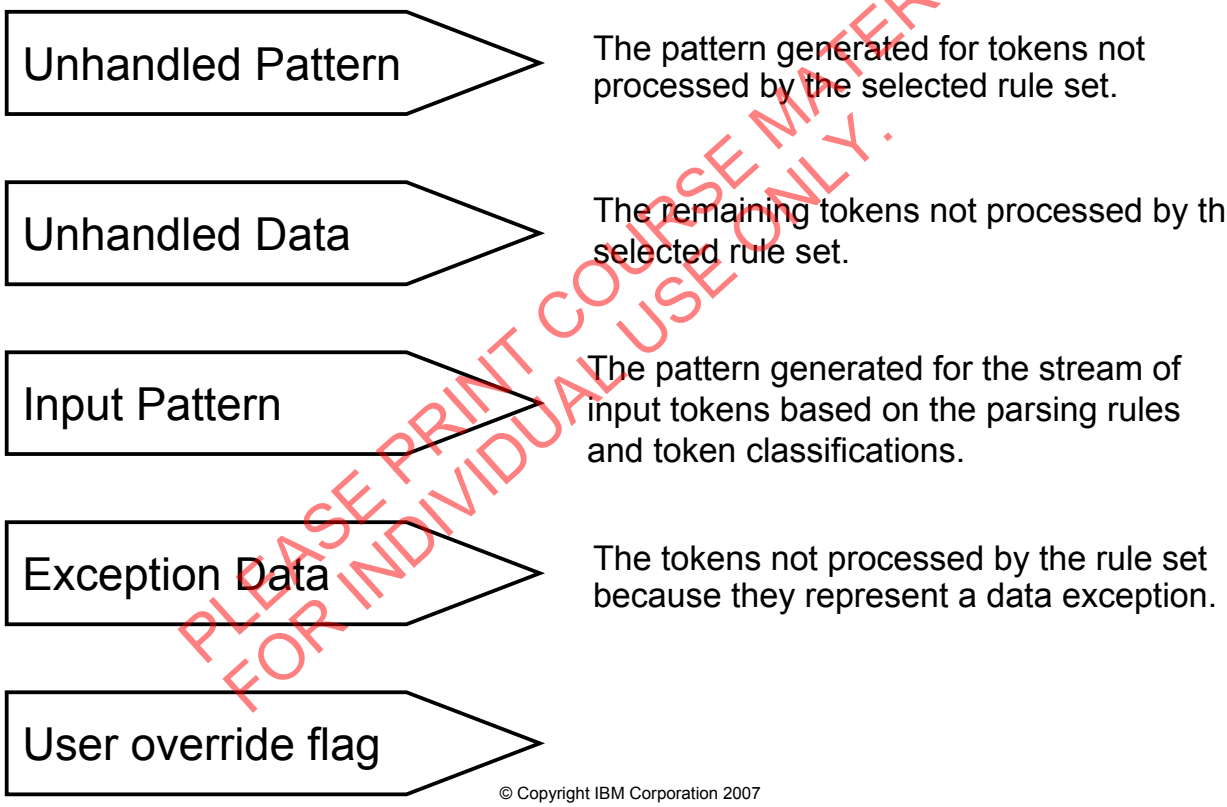
Fields used to facilitate matching.

Soundex field somewhat more stable for reverse phonetics.

Phonetics usually used for name fields.

Hash keys and packed keys are name-specific.

Standardize reporting fields



These components are present in every domain-specific rule set.

Investigate NAME unhandled patterns and data

- Identify the unhandled patterns for the NAME field. In the report include the unhandled data, input pattern, original data and the record key.
  1. Build a Character Concatenate Investigation using the following fields
  2. Increase the number of samples to 5

Field Name	Field Description	Type
UPUSNAM	Unhandled Pattern	C
UDUSNAM	Unhandled Data	X
IPUSNAM	Input Pattern	X
NAUSPRE	Name domain data	X

Standard practice: investigate handled and unhandled data

- Review the business intelligence fields to ensure accurate bucketing of the data
- Build a Character Discrete Investigation for each field and review the contents and the format
- Build Investigation to review:
  - Unhandled Patterns
  - Unhandled Data
  - Input Pattern
  - Input Fields

Example: Ensure that the House Number field (HNUSADD) contains only numeric data

Is the House Number field always blank?

Directions fields contain: N, S, E, W, or NW, NE, SW, SE

You may complete some quick visual inspections, the Investigation reports allow you to quantify the changes and improvements

Customizing rule sets

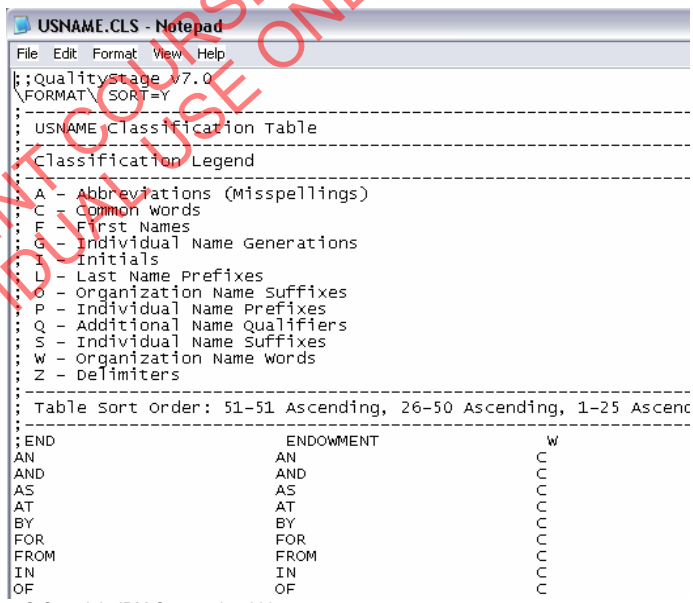
- A rule set may require modification if some input data is:
  - Not processed
  - Incorrectly processed
- QualityStage provides functions to:
  - Modify classification table
  - Apply user Overrides
  - Test strings for classifications using the Rules Analyzer

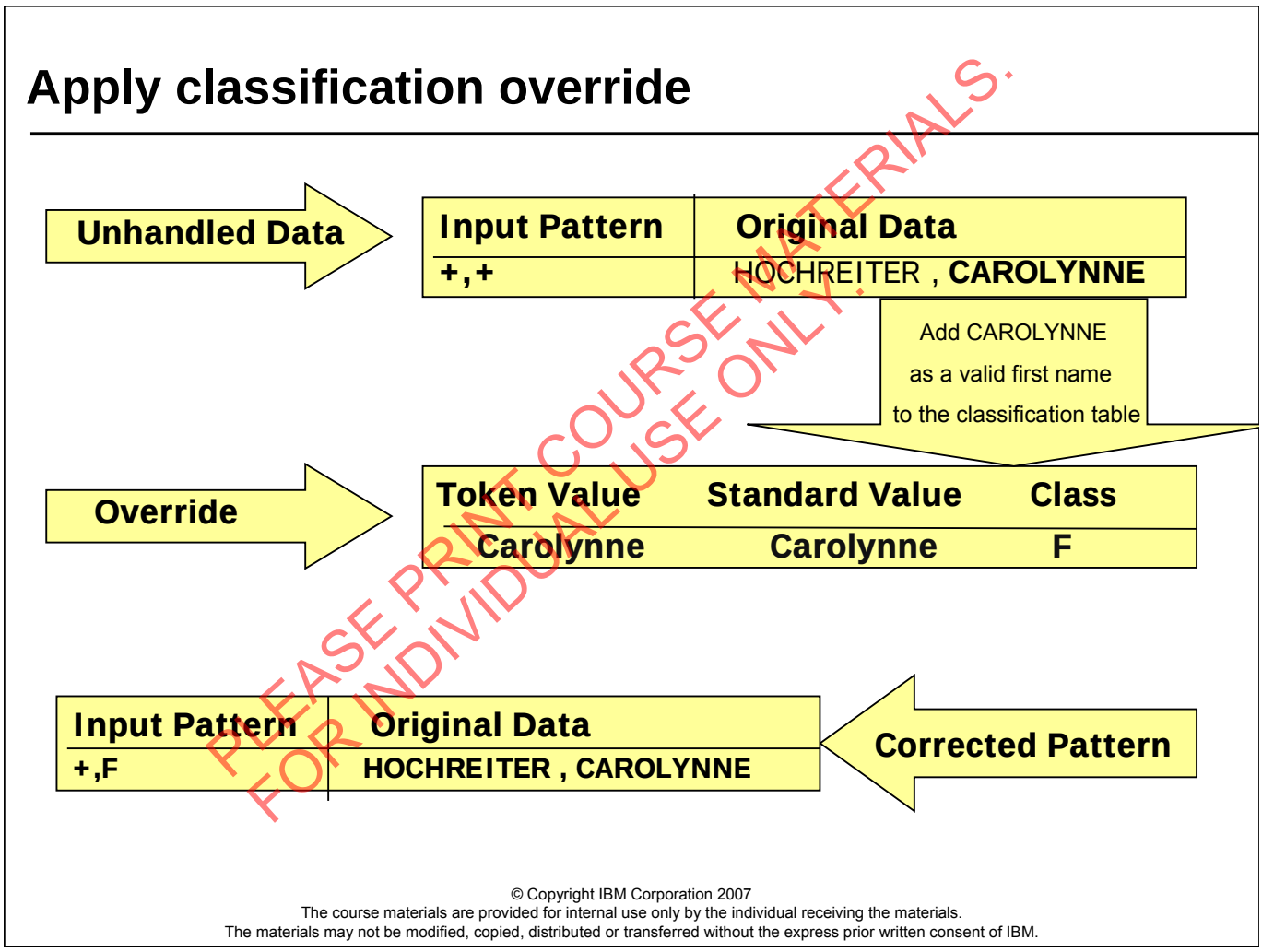
Modify classification table

- In repository
  - Copy the rule set
  - Modify the copy

To modify:

USNAME.CLS





The word (alpha) Carolynne is not recognized as a First Name. Review the Name Word INV frequency report and note that Carolynne appears 5 times in teh data. This frequency influences the decision to add Carolynne as a Classification Override so that it is recognized as a first name.

The input pattern before the override is: +,+ ➔ unknown alpha comma unknown alpha

After the override the pattern is +, F ➔ unknown word, comma first name

The second pattern is recognized and processed by the pattern action file.

User overrides

- Provides the user with the ability to modify rule sets
- The following types of rule sets can be modified using User Overrides
  - Domain Pre-processor rule sets
  - Domain rule sets
- There are five types of user overrides relating to: classifications, patterns, and text strings
- User overrides are
  - GUI Driven
  - Stored in separate lookup tables
- Rule set should be provisioned after modifications applied

Provides the user the ability to specify their own Standardize rules

User overrides are GUI-driven

The user does not need to know pattern-action language syntax

The user does not need to edit the classification table or the pattern-action file

Overrides require the following information:

- Dictionary field name to move the token to
- Original value or standard value of token
- Leading space or no leading space for multiple tokens moved to the same dictionary field

User classification override

- Recognized as a keyword and classified
  - Additional words
    - New abbreviation, variation
    - Misspelling of a word
- User Classifications may override or add:
  - Original values (Token values)
  - Standard value
  - Class

Example:

Carolynne is not recognized as a first name. Since the name carolynne occurs 5 times in the data (review word inv report on names, word frequency count). We might want to add this name to the classification table so that it is recognized.

Use the Word Investigation Word Frequency reports to check the frequency that a word, abbreviation or misspelling occurs.

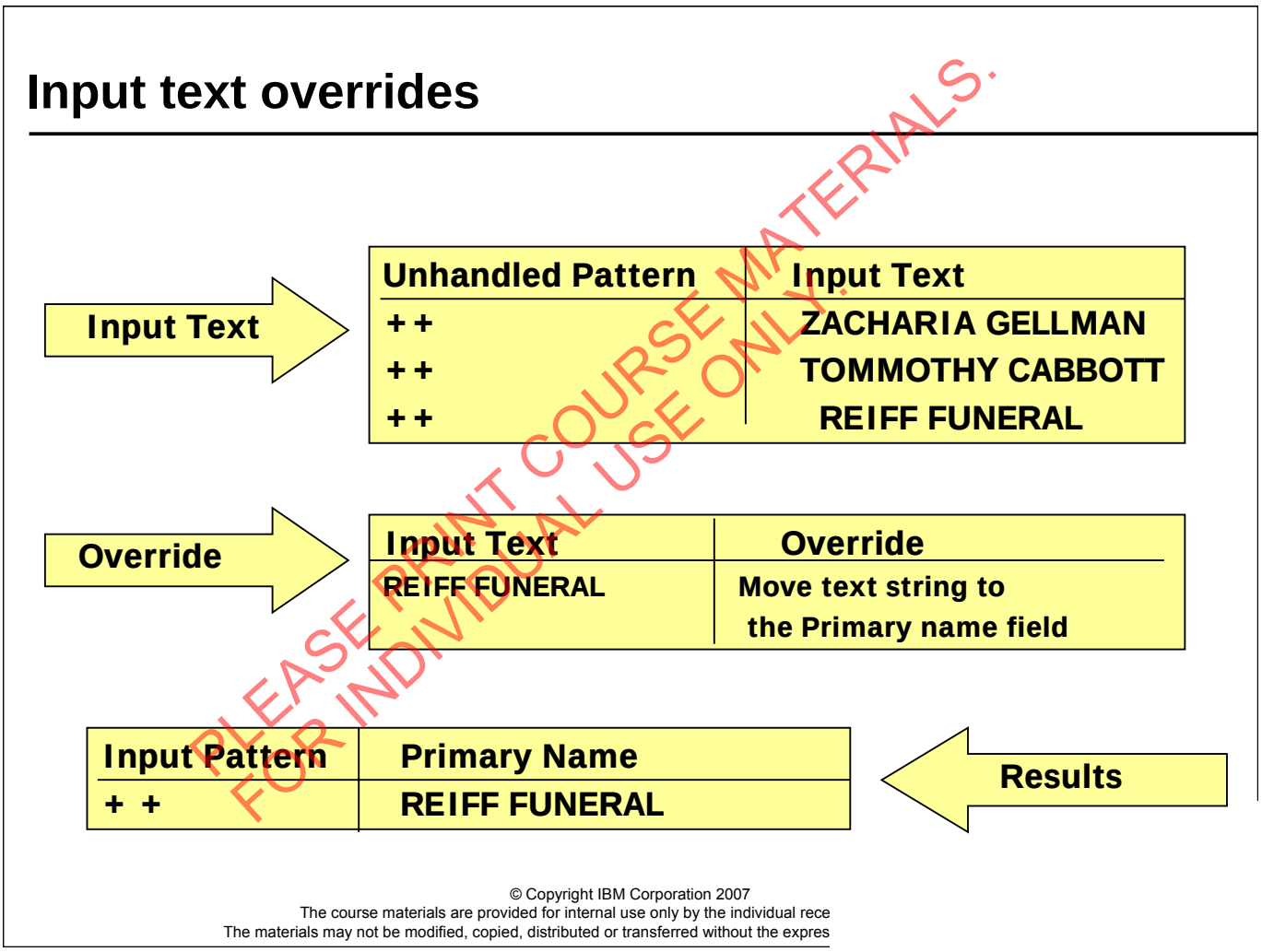
Classification overrides take precedence over the classification table

Classification overrides are available in both domain pre-processor rule sets and domain-specific rule sets

Text overrides

- Allow the user to specify overrides based on an entire text string
- Use this override for special cases and specific handling of a string of text
- Input Text Overrides
  - Applied to the original text string
- Unhandled Text Overrides
  - Applied to the Unhandled Data field

No partial string matching, only complete string matching



The example REIFF Funeral is a “Special case” as it needs to be handled different than the rest of the data with the Unhandled pattern of + +

The remaining Unhandled Patterns of + + may handled the same way. The best type of override to

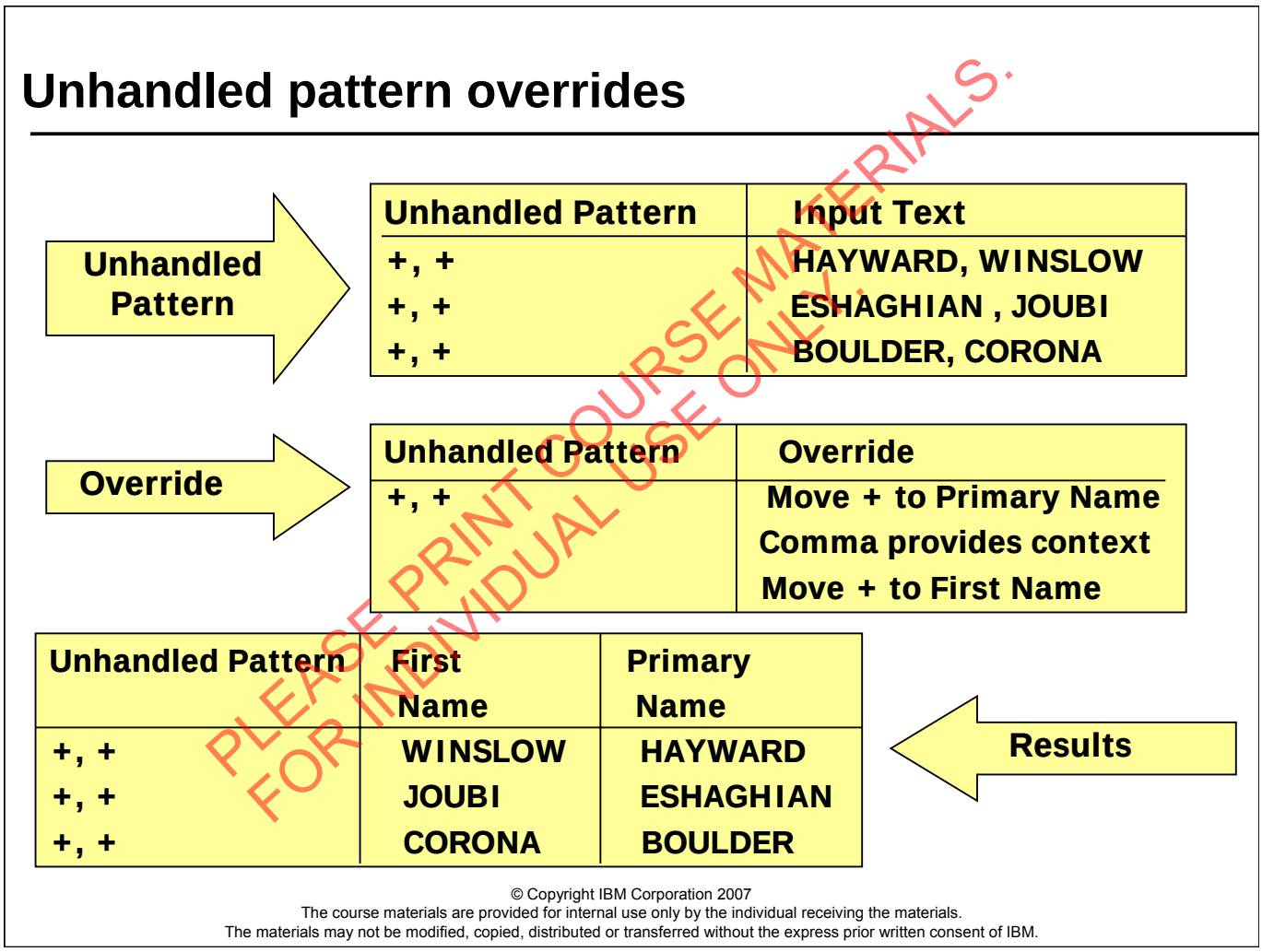
Pattern overrides

- Allow the user to specify overrides based on an entire pattern
- Use this override when most or all records should be processed with identical logic
- Input Pattern Overrides
  - Applied to the original text string
- Unhandled Pattern Overrides
  - Applied to the Unhandled Data field

Again no partial matching, only complete pattern matching

Pattern Overrides are the most general.

Whenever possible use a pattern override as it is more general and will be applied to many records one override improves the data quality on many records vs. a text override which is very specific to a string of text

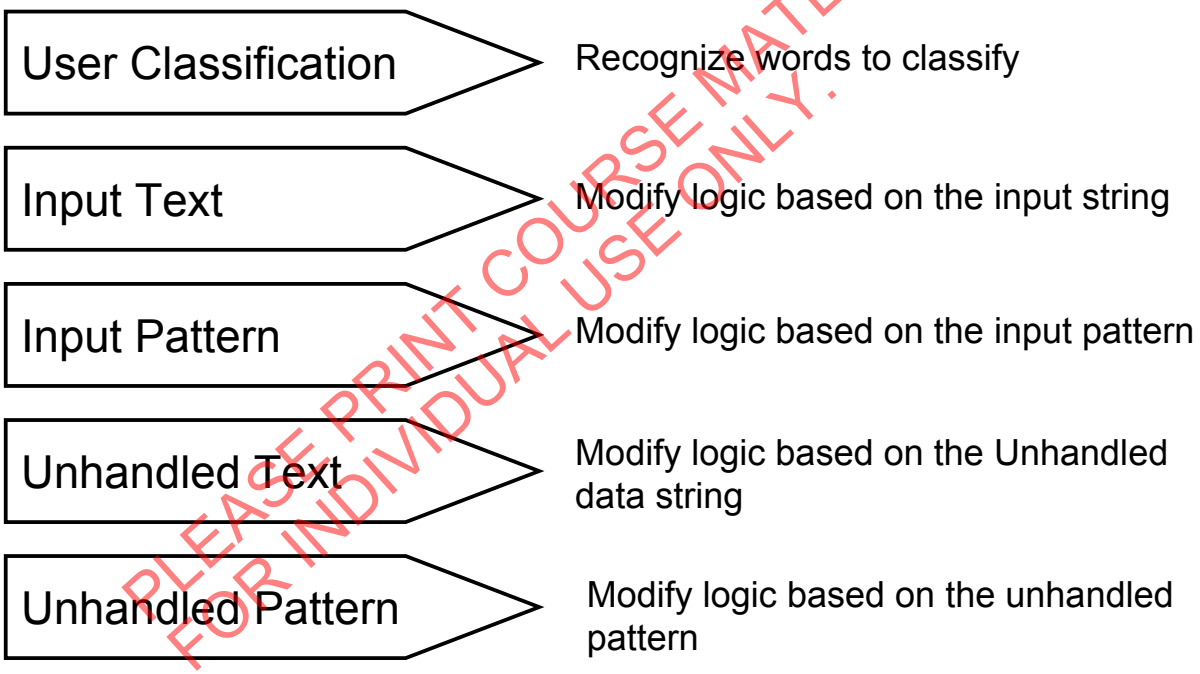


Why an Unhandled Pattern and not Input Pattern?

Below is an example of a record that has a different input pattern then other records in this category, however it has the same unhandled pattern. All records with this unhandled pattern are to be “handled” (processed) the same way, it is more efficient to use one unhandled pattern override rather than having to apply multiple input pattern overrides

+,+ SANCHEZ-CIFUENTES , RYLMA +-+,+

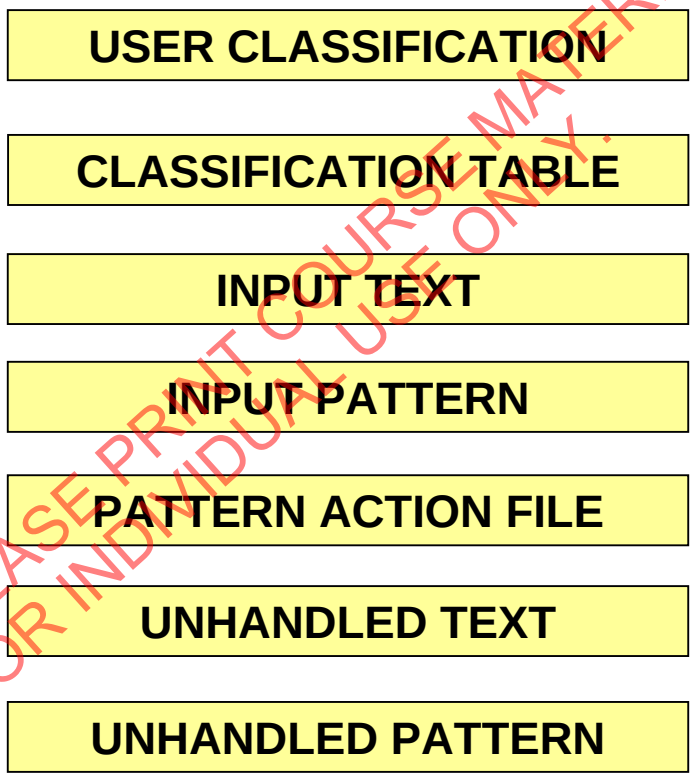
User override precedence



Text overrides take precedence over pattern overrides because they are more specific

Input overrides take precedence over all other patterns in the pattern-action file

Rule set customization precedence



Text overrides take precedence over pattern overrides because they are more specific

Input overrides take precedence over all other patterns in the pattern-action file

Order of what I to look for:

1. Words to classify
2. Input Pattern Overrides
3. Unhandled Pattern Overrides
4. Input Text Overrides
5. Unhandled Text Overrides

Investigate address and area unhandled patterns

- Identify the unhandled patterns for the Address and AREA fields. In the report include the unhandled data, input pattern, original data and the record key.
  1. Build a Character Concatenate Investigation using the following fields
  2. Increase the number of samples to 5

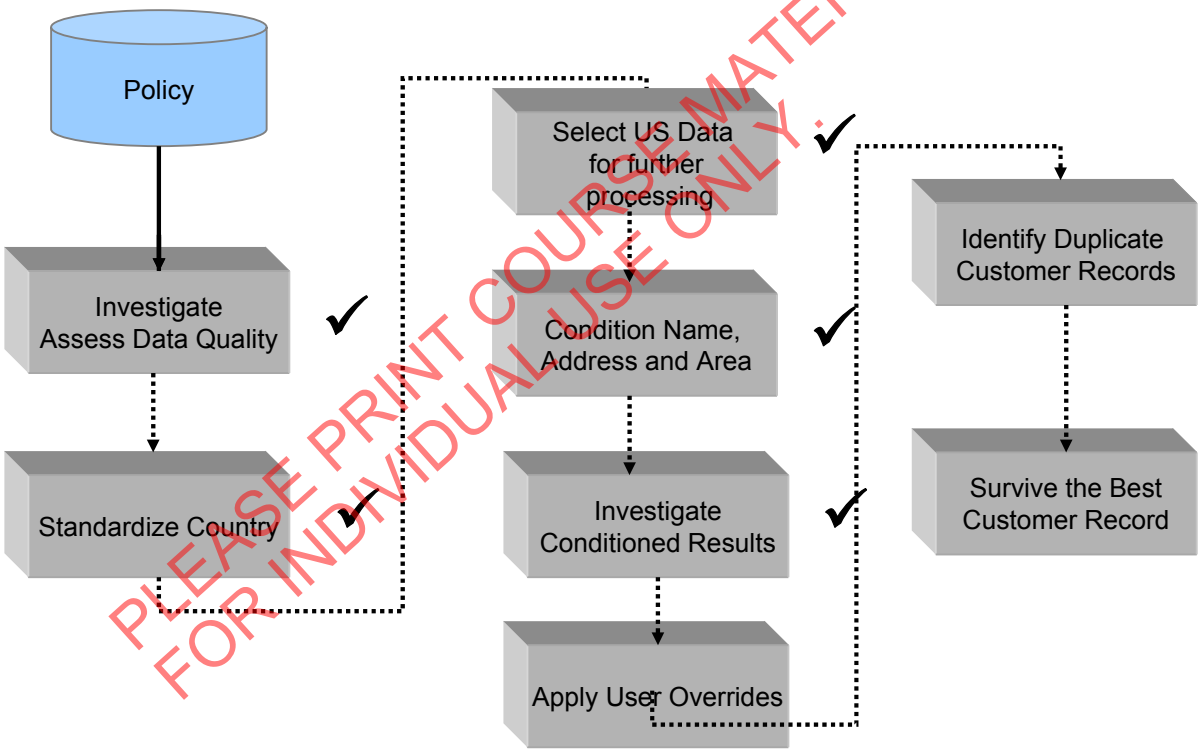
Field Name	Field Description	Type
UPUSADD	Unhandled Pattern	C
UDUSADD	Unhandled Data	X
IPUSADD	Input Pattern	X
ADUSPRE	Address Domain	X

Overrides

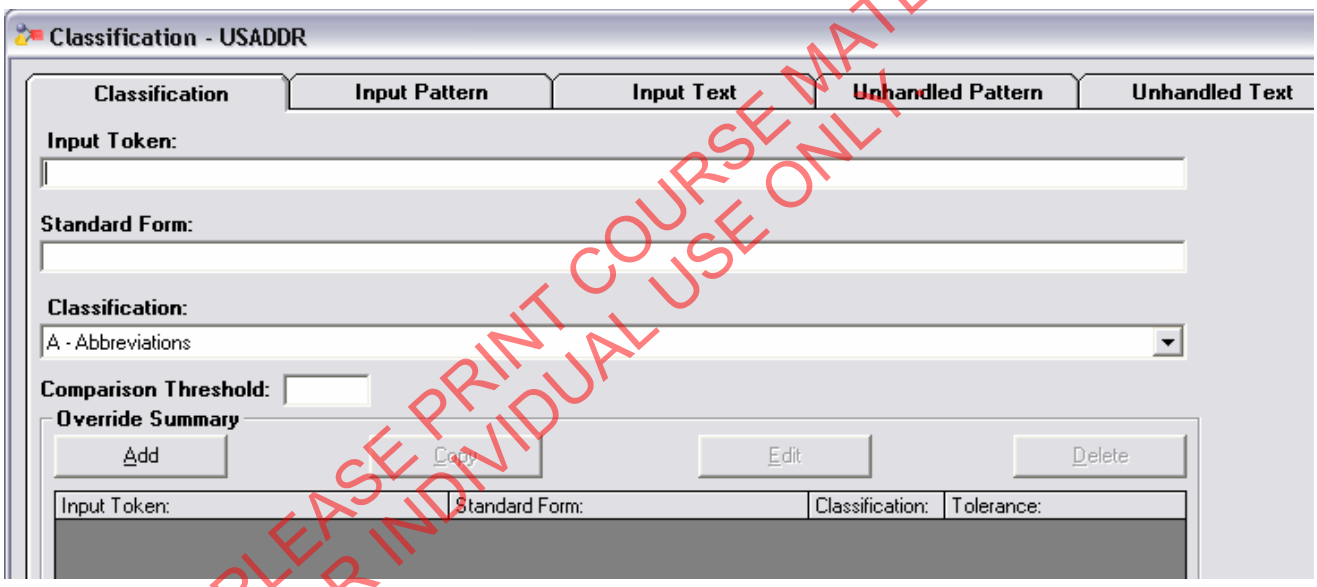
- Purpose
  - Correct problems found during standardization
- Rule set may require overrides because you have data
  - Not processed
  - Incorrectly processed
- Override types
  - Classification
  - Input pattern
  - Input text
  - Unhandled pattern
  - Unhandled text
- Can be tested with rules analyzer

Overrides are used to customized rule sets without applying changes to the Pattern Action File.

Course exercise project design



Overrides screen



Checkpoint

1. (T/F) WAVES can standardize name fields.
2. (T/F) Rule sets are used in standardization processing.
3. Name the components of rule sets.

Checkpoint solutions

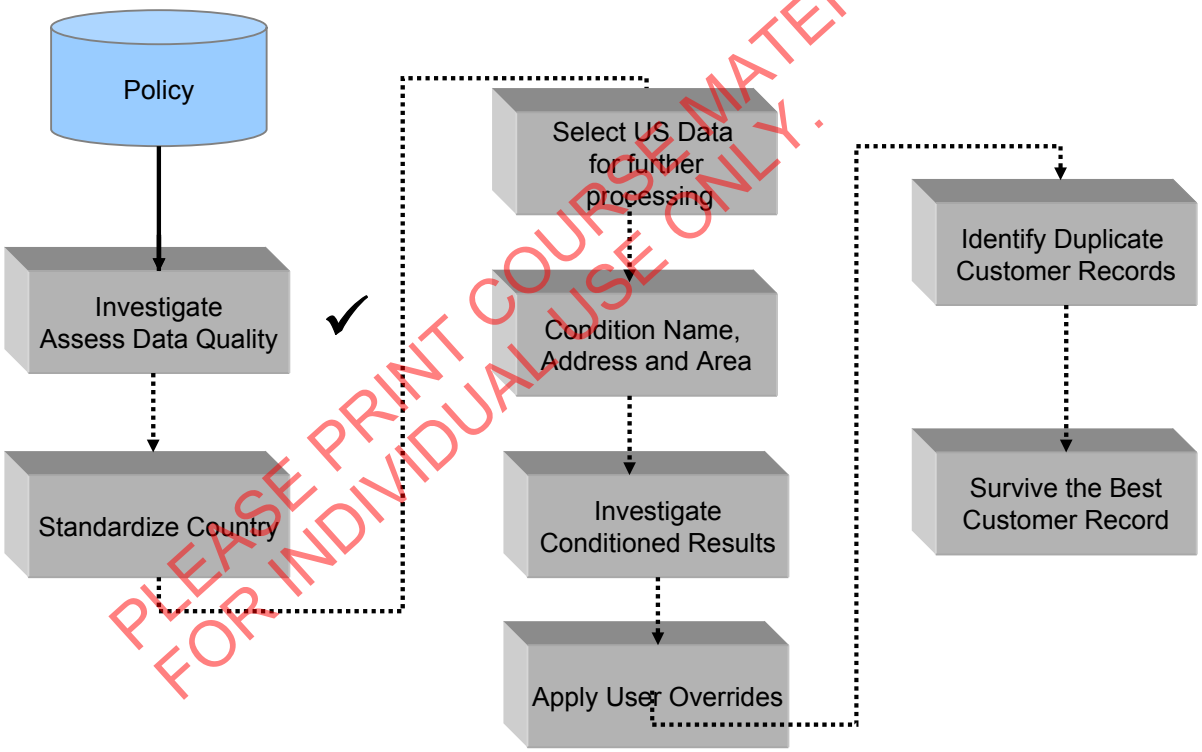
1. (T/F) (T/F) WAVES can standardize name fields.
   Answer: False
2. (T/F) Rule sets are used in standardization processing.
   Answer: True
3. Name the components of rule sets.
   Answer: Classification table, dictionary, pattern action file, lookup tables

Unit summary

Having completed this unit, you should be able to:

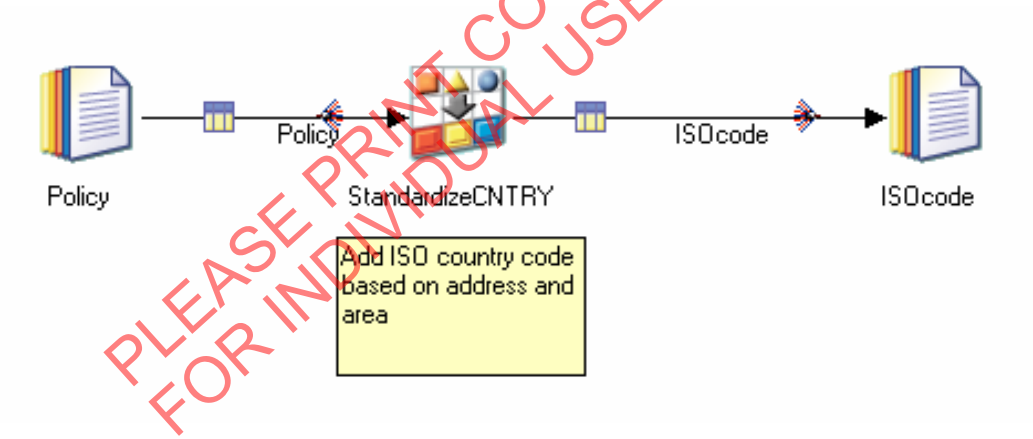
- Describe the Standardize stage in the Data Re-engineering Methodology
- Identify rule sets
- Apply the Standardize stage
- Interpret standardization results
- Investigate unhandled data and patterns

Course exercise project design



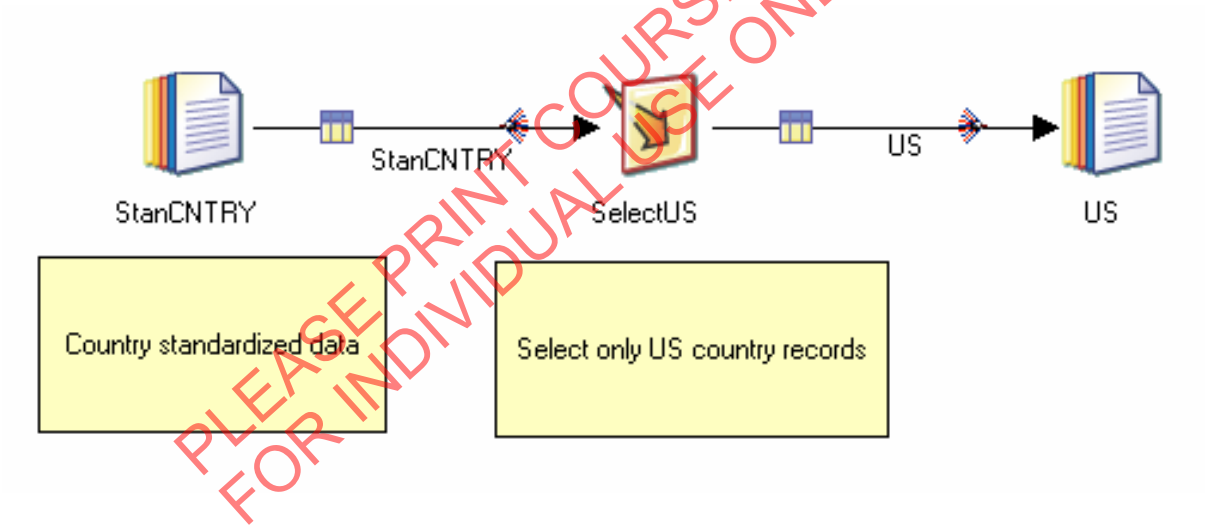
Exercise 10: Standardize country

- Word investigation
  - Uses COUNTRY rule set
- Rule set found in Other folder
- Adds ISO country code to records



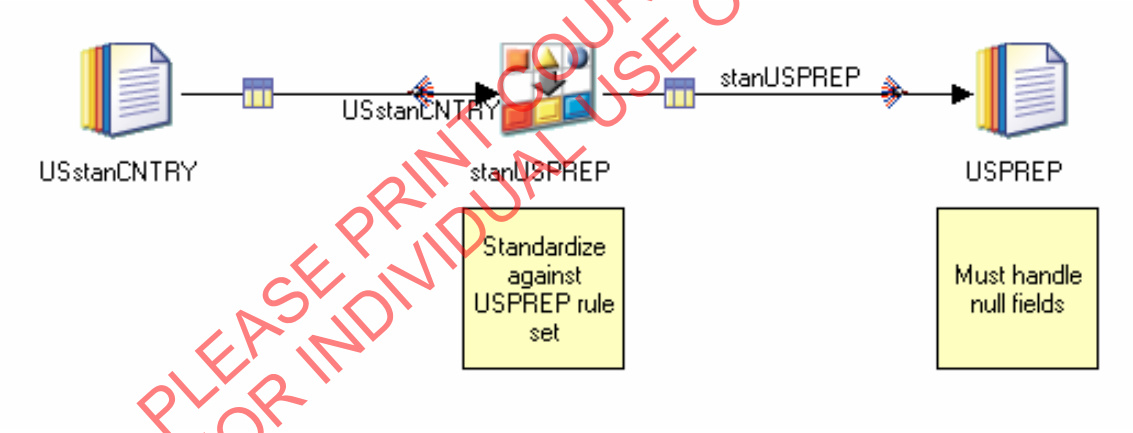
Exercise 11: Select US records

- Uses Select stage to separate records with US ISO code
- Could also use Transformer stage



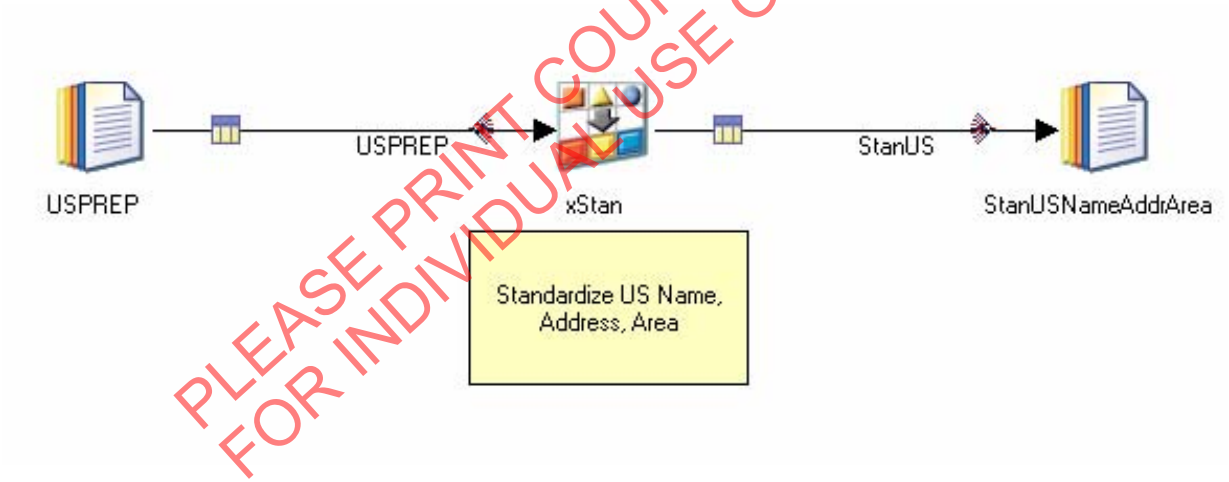
Exercise 12: Standardize USPREP

- Word investigation
  - Uses rule sets, must be provisioned
- Rule set found in US folder



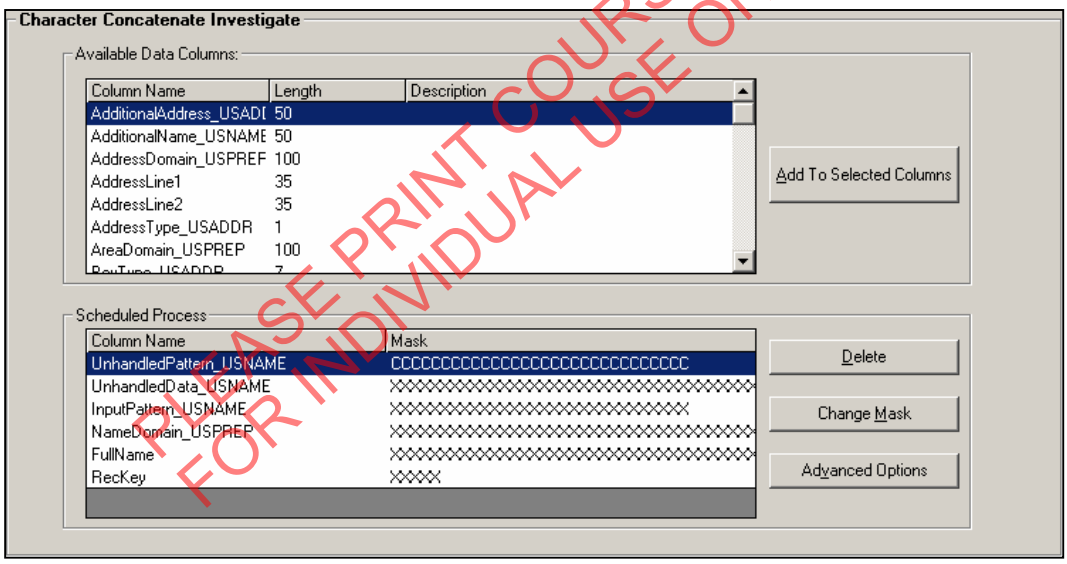
Exercise 13: Standardize USNAME, USADDR, USAREA

- Word investigation
  - Uses rule sets, must be provisioned
- Rule set found in US folder

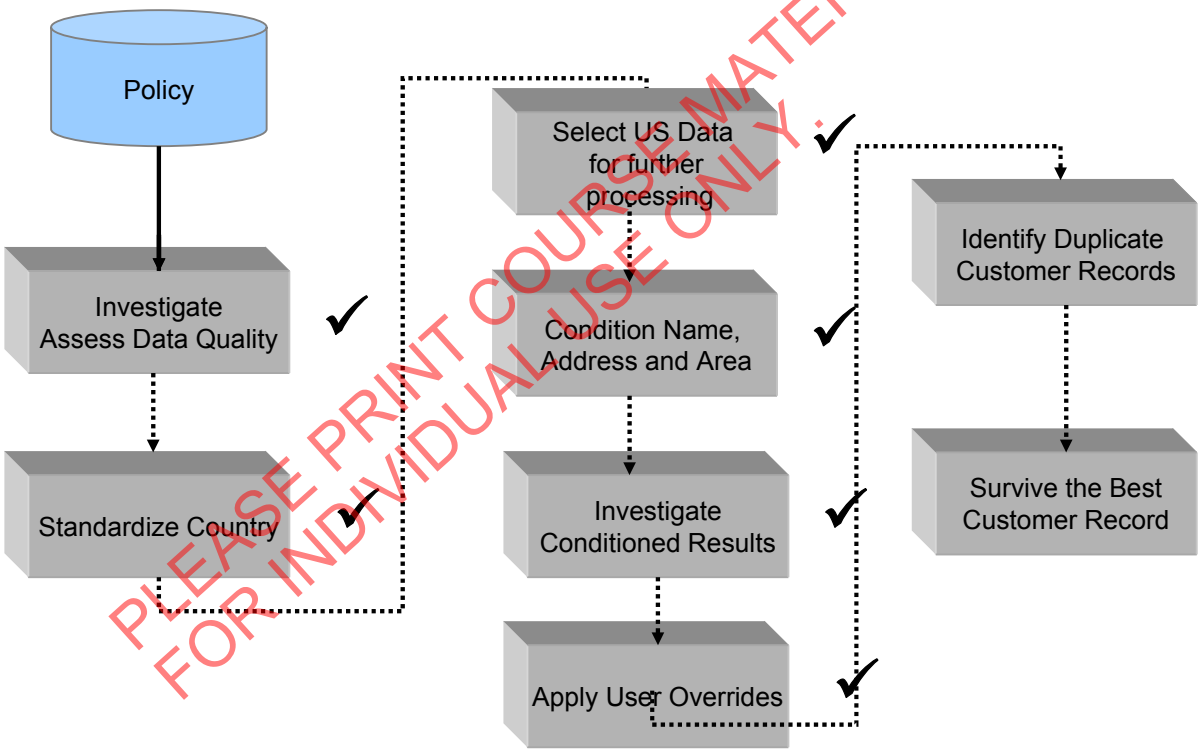


Exercise 14: Investigate standardization results

- Character concatenate investigation
- C mask used to produce histogram
- X mask used to display other fields of interest

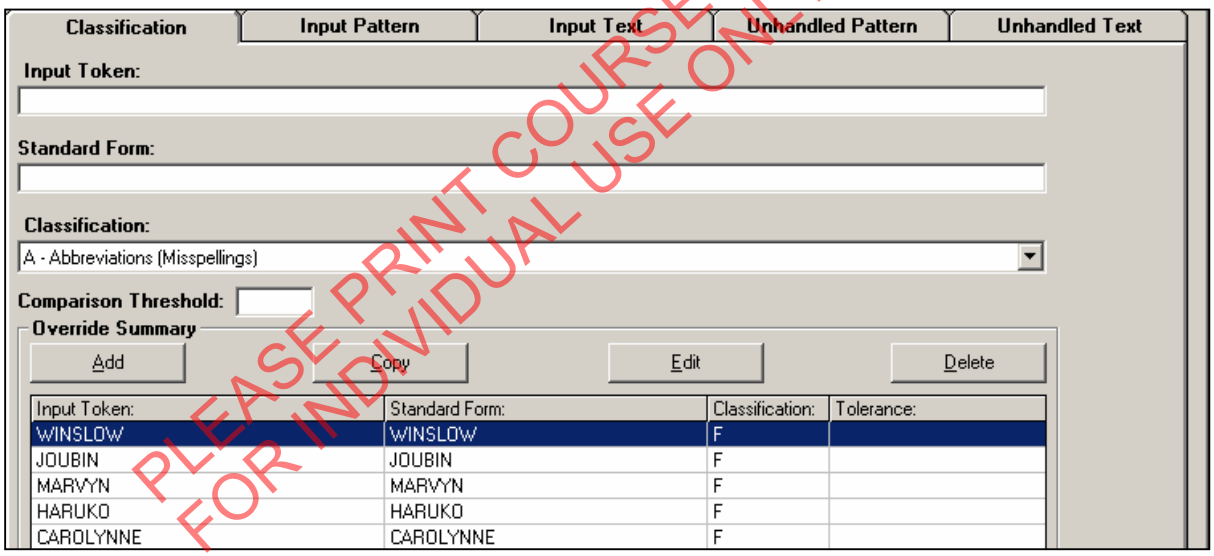


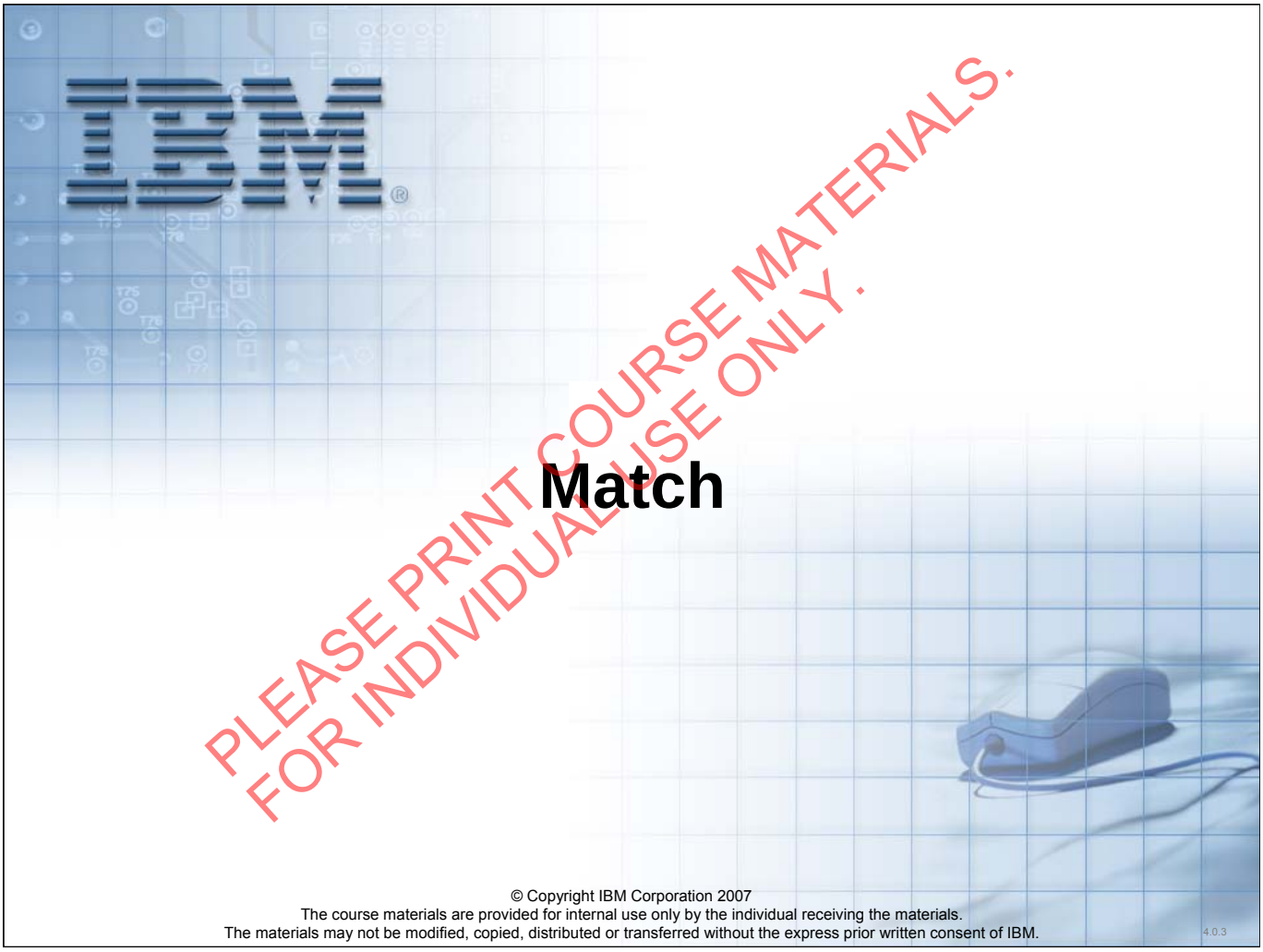
Course exercise project design



Exercise 15: Apply user overrides

- Classification
- Input pattern
- Unhandled pattern




Match
PLEASE PRINT COURSE MATERIALS.
FOR INDIVIDUAL USE ONLY.
© Copyright IBM Corporation 2007
The course materials are provided for internal use only by the individual receiving the materials.
The materials may not be modified, copied, distributed or transferred without the express prior written consent of IBM.
4.0.3

Unit objectives

- After completing this unit, you should be able to:
  - Describe QualityStage Match concepts
  - Define the type of matching algorithms
  - Describe the importance of blocking
  - Apply multiple match passes to increase efficiency/efficacy
  - Interpret and improve match results

Match stage

- Statistically-based method for determining matches
- Over 24 match comparison algorithms providing a full spectrum of fuzzy matching functions
- Ability to measure informational content of data
- Identify duplicate entities within one or more files
- Match Designer
- Critical field settings

Probabilistic record linkage theory is a subset of comparison of data.

What constitutes a good match?

Which of the following record pairs is a match? And how do you know?

```
W  HOLDEN  12 MAIN  ST
W  HOLDEN  12 MAINE ST

  W HOLDEN  128 MAIN  PL  02111 12/8/62
  W HOLDEN  128 MAINE PL  02110 12/8/62

    WM   HOLDEN  128A MAIN  SQ  02111 12/8/62  338-0824
    WILL HOLDEN  128A MAINE SQ  02110 12/8/62  338-0824
```

- Do you compare all the shared or common fields?
- Do you give partial credit?
- Are some fields (or some values) more important to you than others? Why?
- Do more fields increase your confidence?
- By how much? What is enough?

Before we discuss the technology of matching, think about the human process that you would apply in making a decision about these record pairs.

Do you fee ‘comfortable’ about the first pair? Is there really enough information to suggest that these records should be linked? The two locations could be anywhere in the world and the first name initial doesn’t offer much supporting information either.

What about the second pair? Now we know that we’re dealing with the same geographical area and the same birthdate. Has this additional information given you greater confidence? Do you find yourself assigning more-or-less importance to some of the fields or some of the values? For instance, does the abbreviation of PLACE (PL) carry a little more weight in your mind than the abbreviation of STREET (ST) even though they are both just 2 characters? Does the 3 digit building number and the matching PLACE words give you sufficient confidence that these two versions of MAIN are likely the same even though there is a one-digit conflict in the Zip Code? Is the date-of-birth sufficient to say that these are the same person, or is there still some risk of them being twins?

By the time you get to the 3rd pair your confidence should be very high. We now have phone number to further support the location data, and enough first name information to eliminate the risk of twins.

These are the issues that automated matching must consider as well. Being accurate, consistent and justifiable are essential; being able to navigate the “gray-areas” of missing and conflicting values is what separates the simplistic from the industrial strength methods. Now lets look at the methods.

The value of information content

- Information content measures the significance of one field over another (Discriminating Value)
  - A Gender Code contributes less information than a Tax-Id Number
- Information content also measures the significance of one value in a field over another (Frequency)
  - In a First-Name Field, JOHN contributes less information than DWEZEL
- Significance is determined by a value's reliability and its ability to discriminate, both can be calculated from your data

So what does INFORMATION CONTENT mean, and why should I care?

It's the phrase that describes the scientific process of measuring the amount of emphasis, meaning, significance, usefulness, or decide-ability that a piece of data contributes to a process -- in this case the process of determining a match.

Its actually a rigorous and mathematically defined concept based on INFORMATION THEORY. And QualityStage is the premier commercial implementation of that theory. QualityStage investigates your actual data, as a step in the matching process, and dynamically adjusts field and value-level scoring based on the characteristics of the data.

You care because it automates, with far greater precision, the human intuitions that cause you to give more or less emphasis to certain values even within the same field.

It results in greater accuracy and because it gives your matching process a legitimacy and justification not possible through other techniques. And that's often essential to enterprise and mission critical projects whose success is measured by the confidence and trustworthiness of the resulting information.

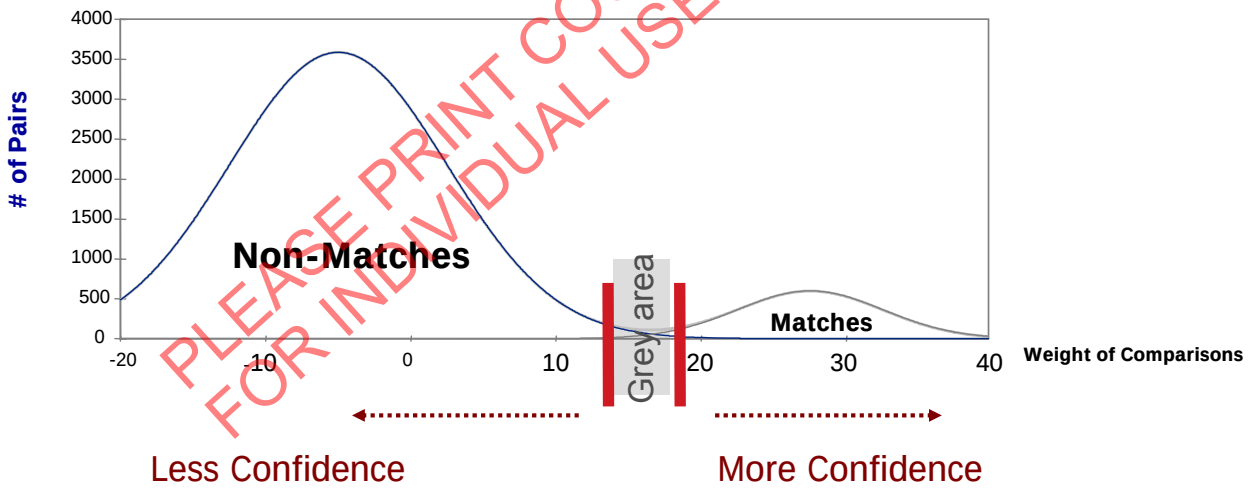
Now lets take a closer look at the Probabilistic process of measuring information content….

Distribution of weights

```
WILLIAM J     HOLDEN   128 MAIN  ST   02111 12/8/62
WILLAIM JOHN  HOLDEN   128 MAINE AVE  02110 12/8/62

 +1     +1     +17     +2   +4    -1     +7     +9     = 40
```

The weighted score is a relative measure of the probability of a match



Less Confidence

More Confidence

The scores (composites weights are relative to all the other scores). Plot the scores to “see” the distribution of the scores.

▲This is the distribution of weights for matched and unmatched records. The more variables added to the match, the further apart these two “humps” will be. It’s the point where the two groups intersect which can cause problems.

▲In our previous example the score of “31.64” based on the distribution this is a fairly high score indicating a high confidence in the match.

▲This is the distribution of weights for matched and unmatched records. The more variables added to the match, the further apart these two “humps” will be. It’s the point where the two groups intersect which can cause problems.

Including more fields is better as long as each field supports your matching goals

Consider how you would either use or omit fields depending on what your match goals are

Weights

- Measures the information content of a data value
- Each field contributes to the confidence (probability) of a match

- We measure the contribution with a weight.
- The more contribution the higher the weight, the less the lower the weight.
- Weight can also be defined as the “discriminating power” of a field

Types of weights

- If a field matches, the agreement weight is used
  - Agreement weight is a positive value
- If a field doesn't match, the disagreement weight is used
  - Disagreement is a negative value
- Partial weight is assigned for non-exact or "fuzzy" matches
- Missing values have a default weight of zero
- Weights for all field comparisons are summed to form a composite weight

▲It is important to combine the theory with "Business Knowledge" to obtain the desired goals.

Example: Statistic program have become so "easy" to use that anyone can build a regression and run it the achieve a result. But it takes an expert, "a knowledgeable business person" to understand the results and the relationship of the input to the result!

Now let's look at that match again and this time we will apply weights.....

Matching terminology

Informational Content	Measures the significance of one field value over another
Weight	Measures the informational content of a data value
Composite Weight	Measures the confidence of a match
Match Cutoffs	Distinguish matches from non-matches
False Positives	Records with a score above the High cutoff that really aren't a match
False Negatives	Records below the low cutoff that really are a match

Measuring the conditions of uncertainty

- Reliability of the data in a given field
  - Estimated as the probability that the field agrees given the record pair is a match
- Probability of a random agreement of values
  - Estimated as the probability the field agrees given the record pair is not a match

Reliability: How “correct” are the data values. How often are the filled-in (non-missing) and when they are filled-in how often are they correct.

▲Chance of Random Agreement: Measures the “rareness” or “uniqueness” of a value. The more frequent (or common) a value occurs in the data the less weight, “confidence” it contributes to the match.

Example: If another Barbara walked into the room would you think AH HAH they must be the same person? ...Well maybe but I’m not convinced. Now, if there were a Vladimir in the room and another Vladimir entered the room, instinctively I would have more confidence that they two Vladimir’s are a match than I would the Barbara’s.

Reliability (m-probability)

- Approximated as, 1 - error rate for the given field
- The higher the m-probability, the higher the disagreement weight will be for the field not matching since the data is considered reliable

- The disagreement weight is proportional to the reliability score
- The m-prob value is entered by the user. It does not need to be an exact measurement as QualityStage will use the user-entered m-prob and improve the measurement based on a sample of the data.
- The more reliable the data in the field the more records are penalized for not agreeing, since errors are relatively rare.

- Estimating the m-prob, if you really don't know then assume a 10% error rate, (m-prob =90% or .9)
- The m-prob is between .001 and .999. It can never be 1 (100%), there is always a chance of error. And it can never be 0 (completely reliable).

- Data is reliable = errors are rare
- Data is not reliable = errors are common

- Example: of M-PROB: If the variable street type has a 12% error rate, then the m-probability is 0.88

Chance agreement (u-probability)

- The u-probability can be approximated as the probability that a field agrees at random (by chance)
- QualityStage uses a frequency analysis to determine the probability of chance agreement for all values
- Rare values bring more weight to a match

▲Rare values have less chance of accidental agreement and contribute more to a match

▲Frequency analysis determines the probability of chance agreement for any values (INTEGITY calculates)

Example: If two records in a matched pair have a name of John Smith specified, you would be less sure that the record pair represented a true match than if Vladimir Horowitz were matched on both records

Rare events have more discriminating power than common events

Frequencies should not be calculated for fields such as individual identification numbers since all values are rare

Calculating weights

- Agreement weight is estimated as:
  - log2(m/u)
- Disagreement weight is estimated as:
  - log2 ((1-m)/(1-u))

M (m-prob) = .9

U (u-prob) = .01

Agreement weight	$\log_2$ (.9/.01)	**= 6.49**
Disagreement weight	$\log_2$ (1-.9)/(1-.01)	**= -3.31**

Note: Use a sample size of 10,000 with a value frequency of 100 for a u-prob of .01 (1 in a hundred)

Blocking

- Grouping together like records that have a high-probability of producing matches
- Only “like” records are compared to each other making the match more efficient and computationally feasible
- Records in a “block” match exactly on one to several blocking fields

Within each group of “blocked” records, each record is compared to every other record according to the matching variables.

Blocking example: sample data

- Block on NYSIIS of Last Name

NYSIIS LNAME	NAME	ADDRESS	ZIP
YANG	YUNG , WAYNE D	9000 SHEPARD DRIVE	78753
GARAS	GEROSA, FRAN X	29 AARONS CT	06877
YANG	YOUNG , JONATHAN A	1767 TOBEY ROAD	30341
GARAS	GERISA, FRANCIS	29 AARONS CT	06877
GARAS	GEROSA, FRANCIS XAVIER	29 AARONS COURT	06877
MATAC	MARCUS MATIC	100 SUMMER STREET	02111
GARAS	GEROSA, MARY	29 AARONS CT	06877
JANCAN	RENEE JENKINS	100 SUMMER STREET	02111
YANG	YOUNG THERESA C	1767 TOBEY ROAD	30341

1. We could compare every record to every other record or we could block them by: Last Name.
2. Enter to show block color-coding
3. Notice the Jerosa record did not make the same block group as the “Gerosa” records. It did not match exactly on Last Name. This is one reason we create the Phonetic coding of some fields in the Conditioning phase. The phonetic coding fields are very useful for blocking as they introduce “fuzziness” to a rigid set of criteria (blocking).

Blocking example – NYSIIS of Last Name

NYSIIS	NAME	ADDRESS	ZIP
YANG	YOUNG , WAYNE D	9000 SHEPARD DRIVE	78753
YANG	YOUNG , JONATHAN A	4220 BELLE PARK DR	77072
YANG	YOUNG THERESA C	1767 TOBEY ROAD	30341
GARAS	GEROSA, FRAN X	29 AARONS CT	06877
GARAS	GEROSA, FRANCIS XAVIER	29 AARONS COURT	06877
GARAS	GEROSA, MARY	29 AARONS CT	06877
GARAS	GARISA, FRANCIS	29 AARONS CT	06877
MATAC	MARCUS MATIC	100 SUMMER STREET	02111
JANCAN	RENEE JENKINS	100 SUMMER STREET	02111

Blocks with only one record are considered residuals

Blocks with one record are considered “residuals”. There are not any other records in the group to compare to.

“Due to an error (potentially) in the last name Jerosa it did not make the same block group as the Gerosa records.”

Balance scope and accuracy

Balance the scope and accuracy to compare a reasonable amount of “like” records

Accuracy

The quality of the candidate records

Scope

The number of records

- If the accuracy is loose the scope is very large
- If the accuracy is too tight then the scope is too small

Example: If you are matching bank records to customers, it is NEVER OK to match the wrong record to a customer. The tolerance for error is very low. The accuracy must be high which causes a narrower scope of records.

Marketing Campaign: In order to market a reasonable number of customers you might be willing to tolerate more error (less accuracy) to get a sufficiently wide scope, or “marketing list”.

Blocking strategy

- Choose fields with reliable data
- Choose fields with a good distribution of values
- Combinations of fields may be used

▲The goal of blocking is to group together like records that have a high probability of producing matches. The character discrete INV reports will help with these decisions, they tell you how often a field is populated.

▲If you choose fields with reliable data then you are “truly” grouping together like records since the data values are reliable (usually correct).

▲Choose fields that make business sense to meet you objective. If you are trying to identify unique customers then blocking by house number isn’t the best choice.

▲Gender usually doesn’t have enough values to break records into groups of 100-200 (guideline). If all the data is from a few states the state may not be the best field.

▲Again Inv reports tell use how often a field is populated, the distribution of the data.

Examples of blocking strategies

- Zip code for matching addresses
- NYSIIS of last name for matching individuals
- Brand name for matching products
- Combination of zip code and NYSIIS of street name for matching addresses
- Combination of NYSIIS of last name and first letter of first name for matching individuals

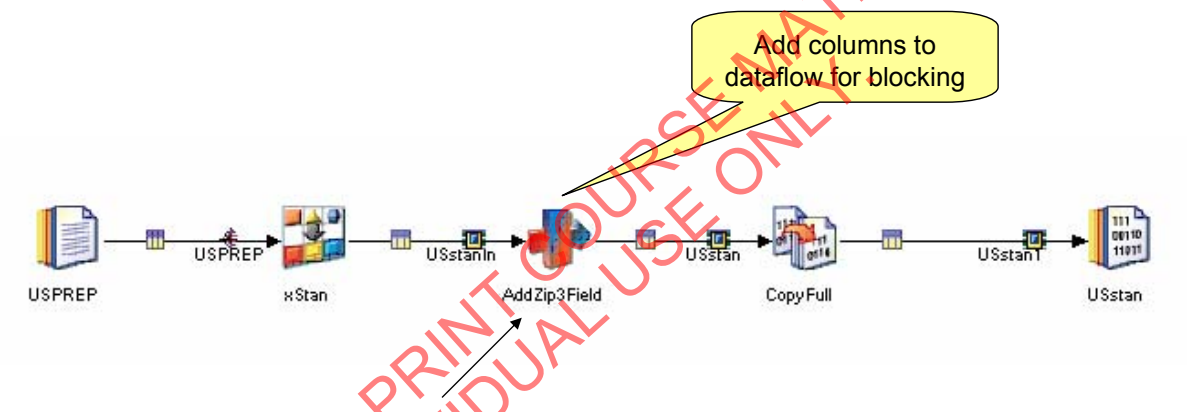
Blocking summary

- Blocking groups together “like” records
- Matching is more efficient for small block sizes
  - Blocks should have less than 1000 records
- Blocking fields must match exactly for a candidate set to be created/evaluated

Example:

100*100 records (10,000 comparisons) is much faster than 200 (200*200 or 40,000 records)

DataStage job with Transformer

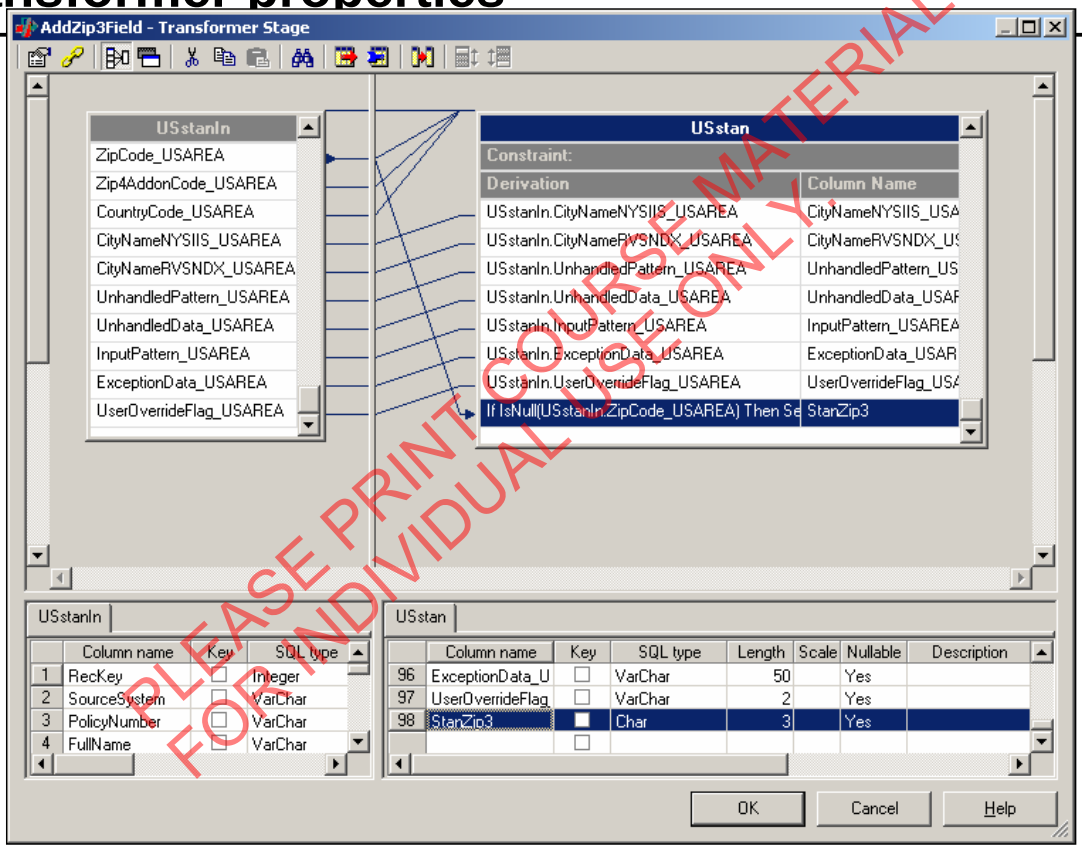


Column definition:

If IsNull(USstanIn.ZipCode_USAREA) Then SetNull() Else USstanIn.ZipCode_USAREA[1,3]

Transformer properties

IN



OUT

Exercise 16: Add fields using Transformer stage

- Create new field to be used in blocking

Match types

- Unduplication
  - Identifies duplicates candidates in one file
- Reference Match (Two File)
  - One-to-one correspondence
    - For every record on stream link we expect to find a match to one record on reference link
  - Many-to-one correspondence
    - More than one record on stream link can match to the same record on reference link

Examples:

One-to-one means Customer records from the billing system should have one-to-one correspondence with customers from the marketing database.

One-to-Many means Many Visa transactions will match to the same credit card number. Many addresses match to one postal code.

Comparing data values

- Different comparisons for different data
- Over 24 comparison methods
- Most common
  - CHAR - (character comparison) character by character, left to right.
  - UNCERT - (character uncertainty) tolerates phonetic errors, transpositions, random insertion, deletion, and replacement of characters
  - CNT_DIFF – Counts keying errors in numeric fields. You set a tolerance threshold
  - NAME_UNCERT – Can be used to compare and character values, if the strings are different lengths then the shorter of the two lengths is used

Over 24 ways to compare data values

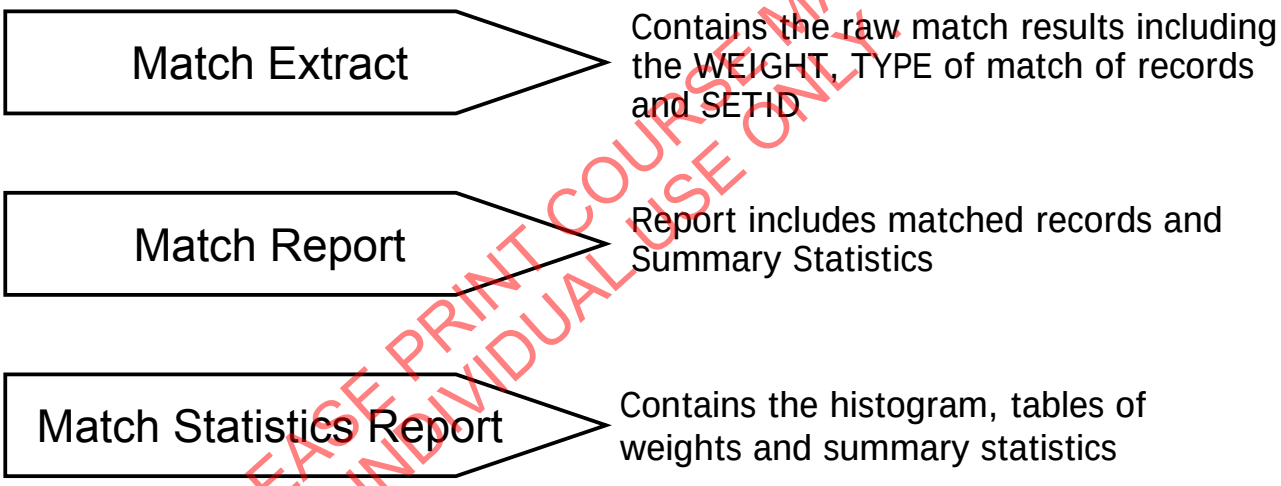
Char = Exact = Total agreement weight

Uncert = Fuzzy = Agreement weight is prorated based on how close to exact.

If records are not “close enough” then the disagreement weight is assigned.

These two are the most popular ways to compare data in fields.

Match output files



Must examine both sequential files and job log

Review Match extract for match results

Review the Match Report with clients and Business Analysts

Match Debug file, check for block overflow and review the Histogram

Match extract

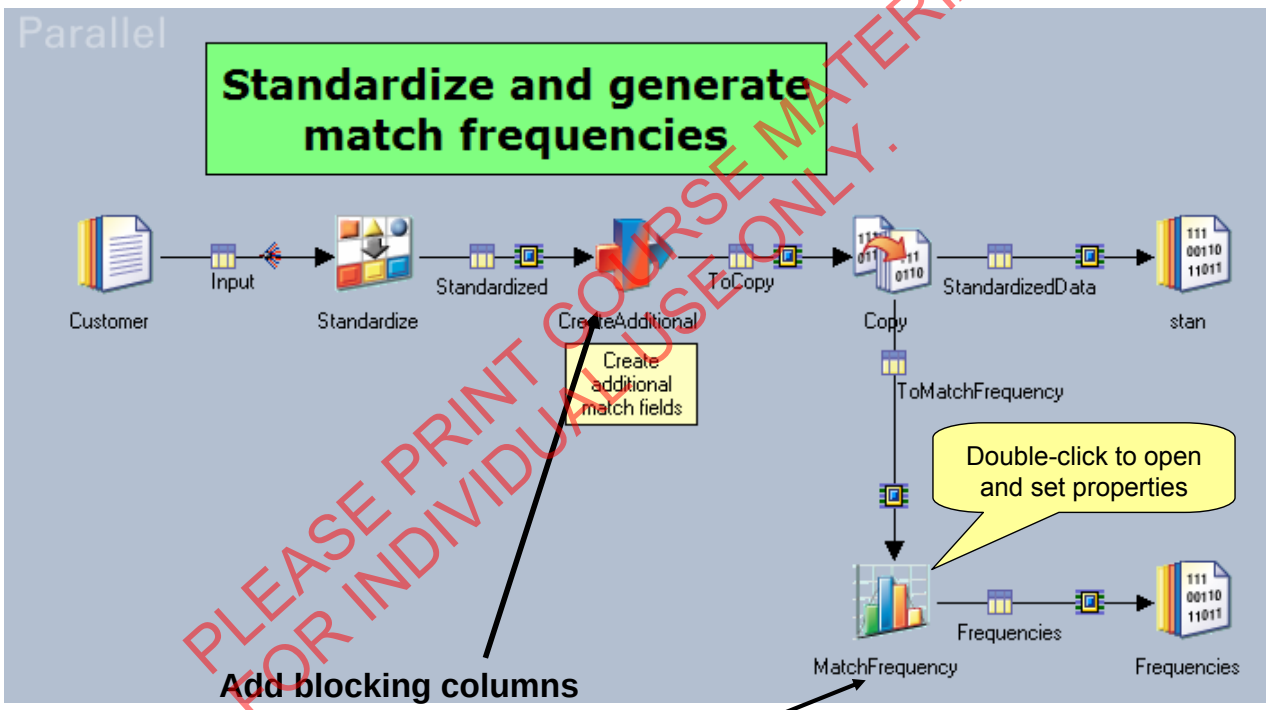
SETID	TYPE	PASS	WEIGHT	ALL_OF_THE_DATA
393	XA	1	55.32	MICHAEL F DOHERTY
393	DA	1	41.36	MICHAEL F DOUGHERTY
468	XA	1	50.40	EUGENE B BOROWITZ
468	DA	1	24.01	BOROWITZ FAMILY TRUST
468	DA	1	47.26	GENE BOROWITZ
520	XA	1	52.75	FRAN X GEROSA
520	DA	1	40.95	FRANCIS XAVIER GEROSA
520	DA	1	52.75	FRANCIS X GEROSA
520	DA	1	41.22	FRANK X GEROSA
1035	RA	1		DARRYL F LINDBERG

Match Implementation

Tasks required in match process

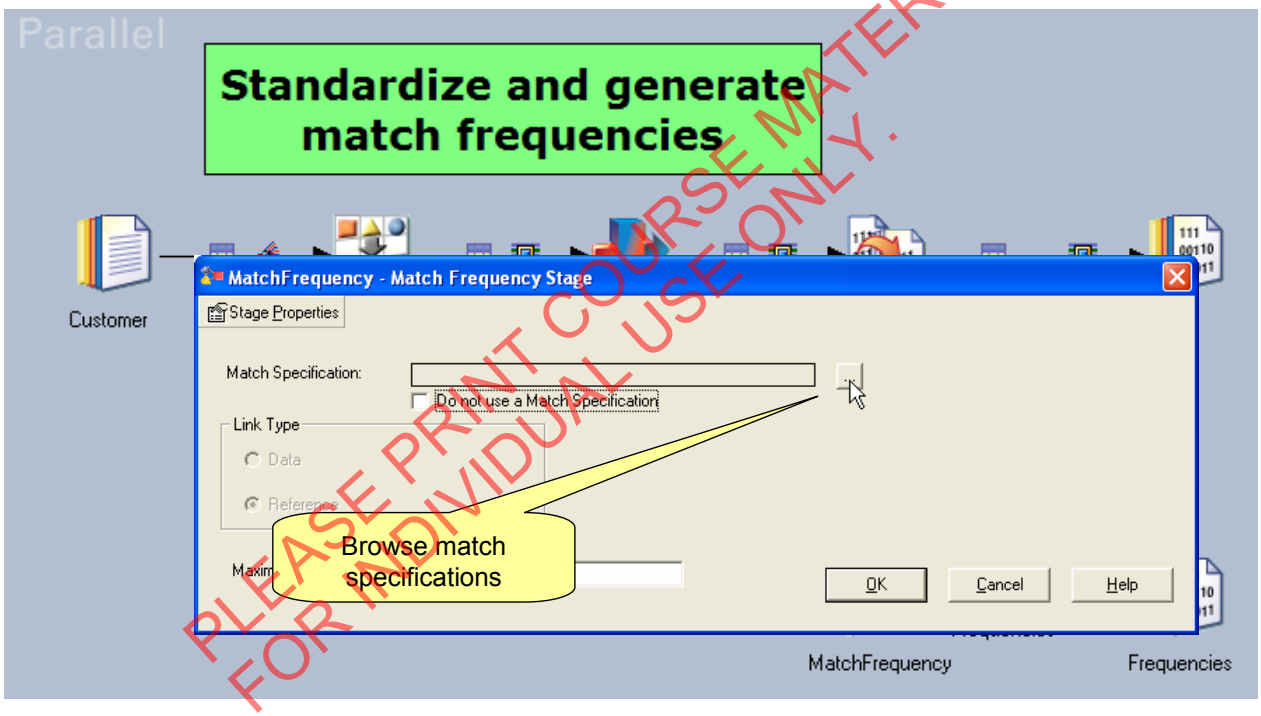
- Standardize the data
- Add data columns needed for blocking
- Generate match frequency report
- Build match specification in Match Designer
  - Add pass
    - Blocking columns
    - Match commands
  - Configure match test results environment
- Run pass
- Review results
- Tune the match
  - Add cutoffs
  - Set overrides
  - Add more passes
- Repeat steps until match results are acceptable

Add columns and generate match frequency



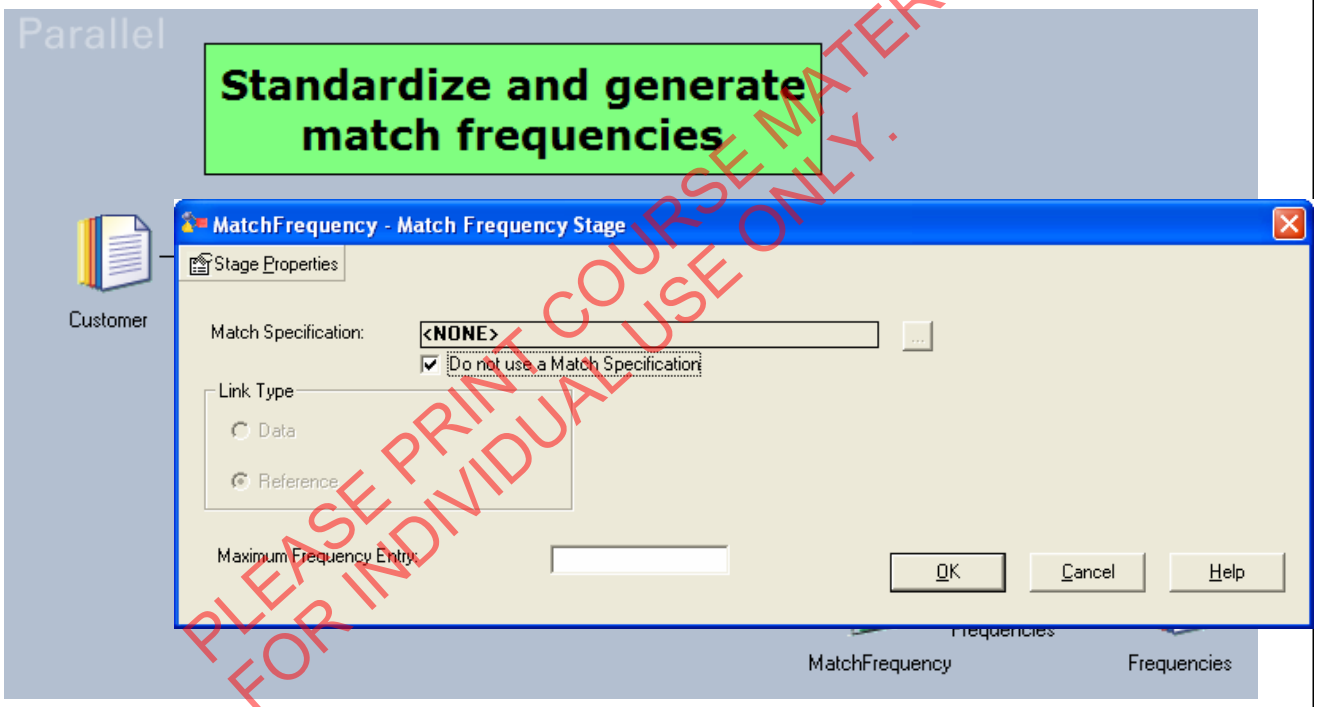
Frequency data is used by the match process.

Match frequency generation



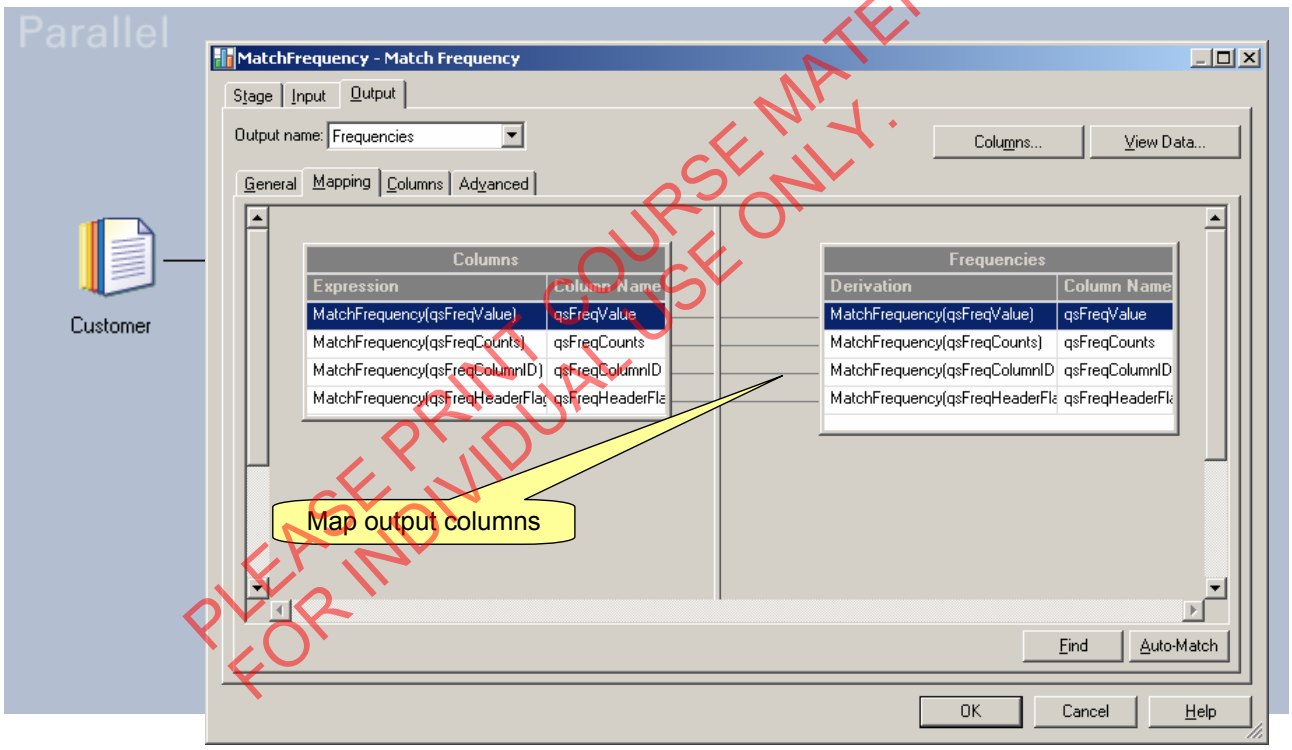
Option 1: Use an existing Match Specification

Match frequency generation

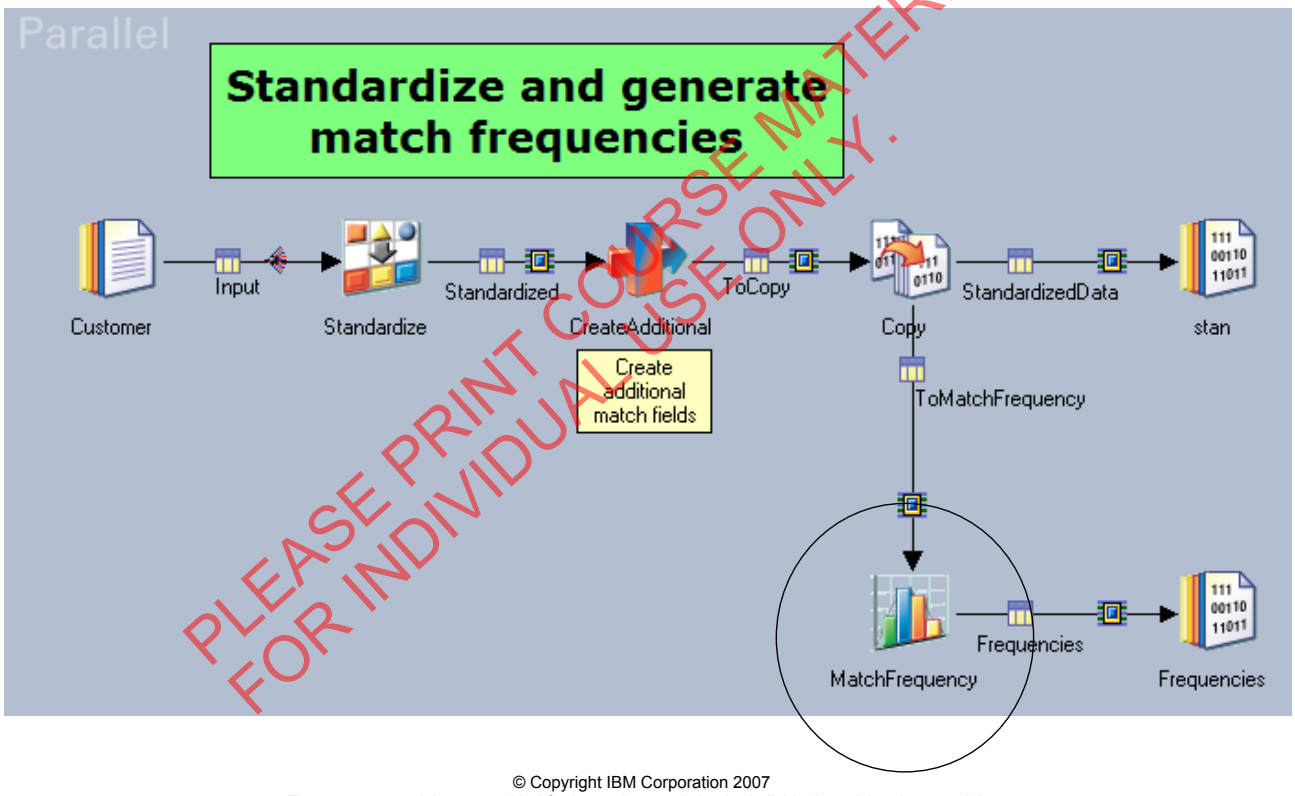


Option 2: Build the frequencies and develop the match

Match frequency generation



Match frequency generation



Match frequency generation

qsFreqValue	qsFreqCounts	qsFreqColumnID	qsFreqHeaderFlag
ADAMSON	00000003 00000000 00000000	97	1
ARMSTRONG	00000002 00000000 00000000	97	1
AVINGER	00000003 00000000 00000000	97	1
BAKER	00000002 00000000 00000000	97	1
BARNETTE	00000003 00000000 00000000	97	1
BARR	00000002 00000000 00000000	97	1
BELL	00000003 00000000 00000000	97	1
BERNEY	00000009 00000000 00000000	97	1
BERNEY FNCL SYSTEMS	00000003 00000000 00000000	97	1
BRUNSON	00000003 00000000 00000000	97	1
BULLINGTON	00000003 00000000 00000000	97	1
CANNON	00000003 00000000 00000000	97	1
CARLISLE	00000002 00000000 00000000	97	1
CARNELL WILKES	00000003 00000000 00000000	97	1
CARROLL	00000004 00000000 00000000	97	1
CHANCE	00000002 00000000 00000000	97	1
CHENEY	00000004 00000000 00000000	97	1
CHRISTENBERRY	00000006 00000000 00000000	97	1
COGBORN	00000005 00000000 00000000	97	1

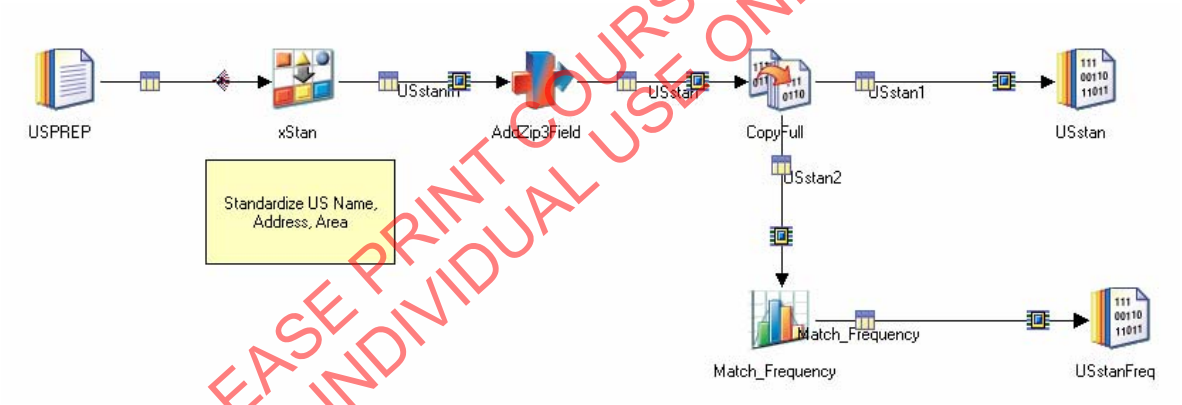
stan

MatchFrequency

Frequencies

Exercise 17: Match frequency

- Use Match Frequency stage in a match job



Match Designer

- Used to build a match specification that will be addressed in a match job

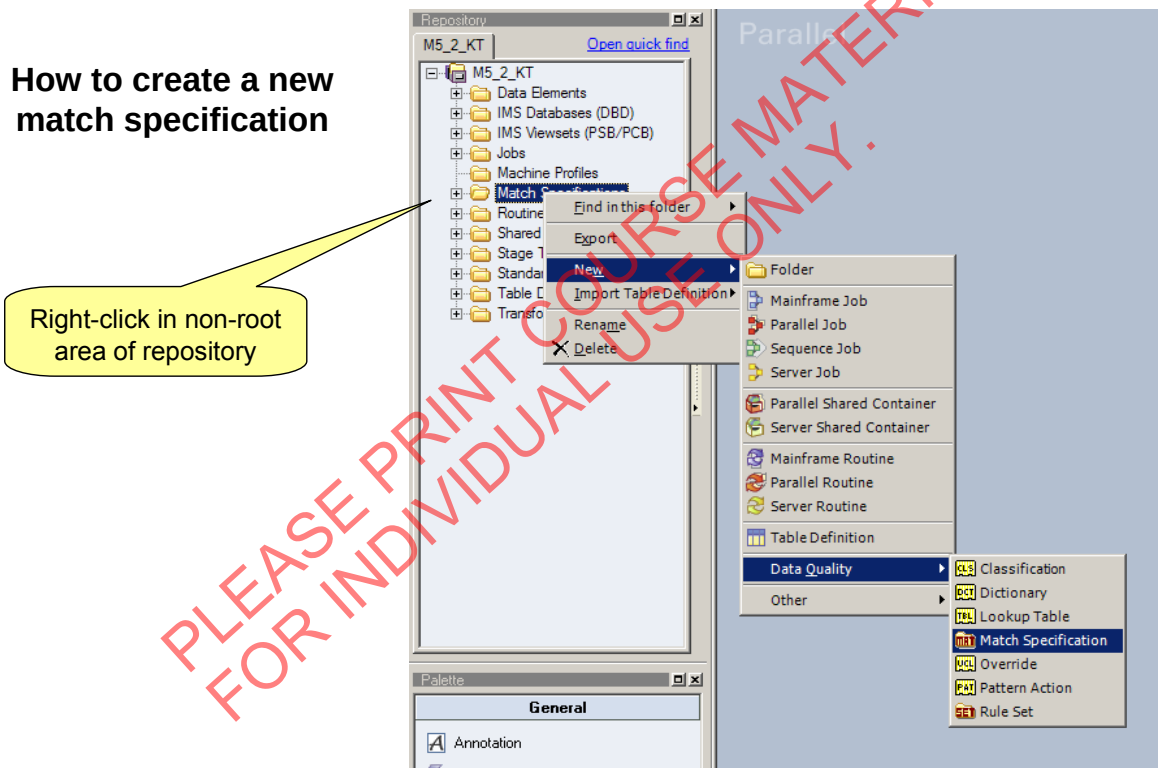
Features

- Design control center
- A data-centric, graphical, self-contained environment
- Graphical representation of statistics
- Match design is independent of job design
- Reusable components
- Separate design from the physical data representation
- Iterative development

Match Design - Unduplicate

Match design - unduplicate

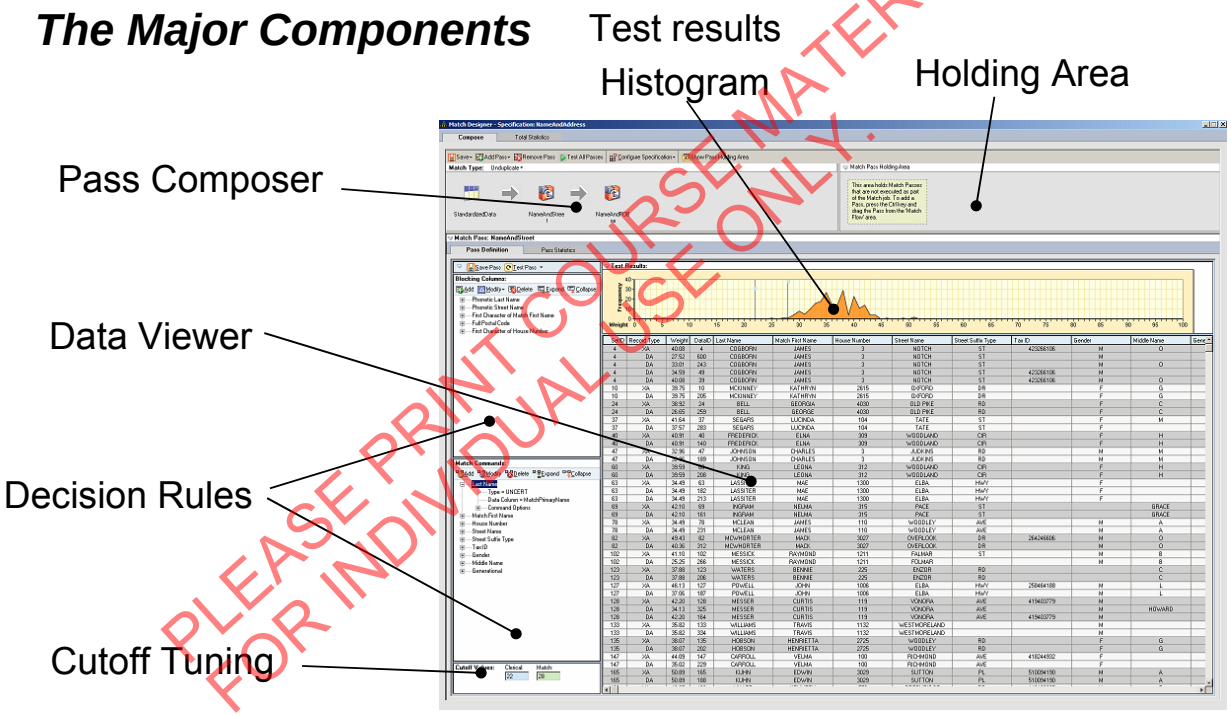
How to create a new match specification



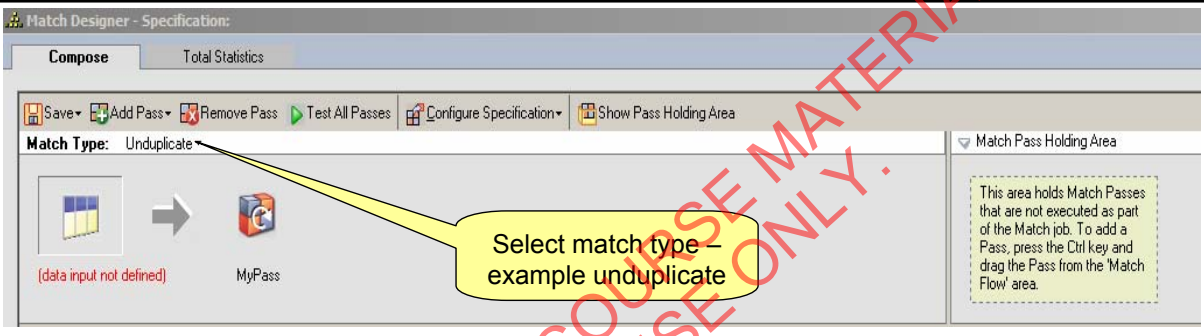
To start the Match Designer, right-click in the repository view.

Match design - unduplicate

The Major Components

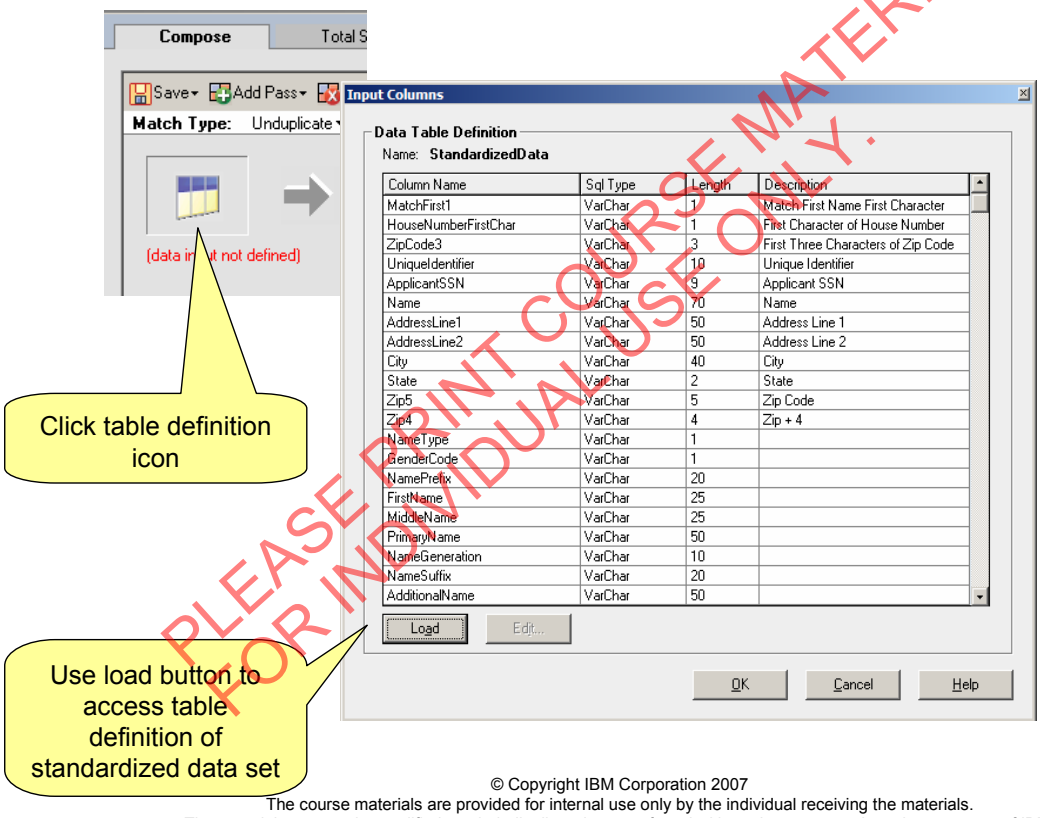


Match design - unduplicate

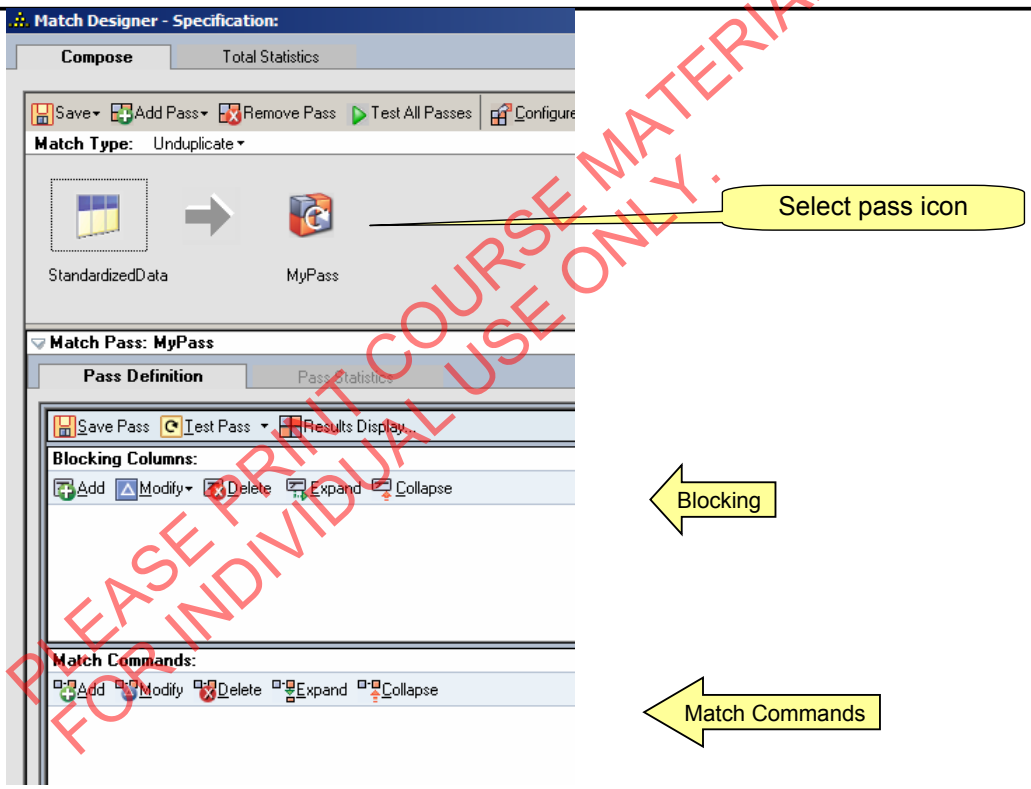


Will initially get one pass called MyPass

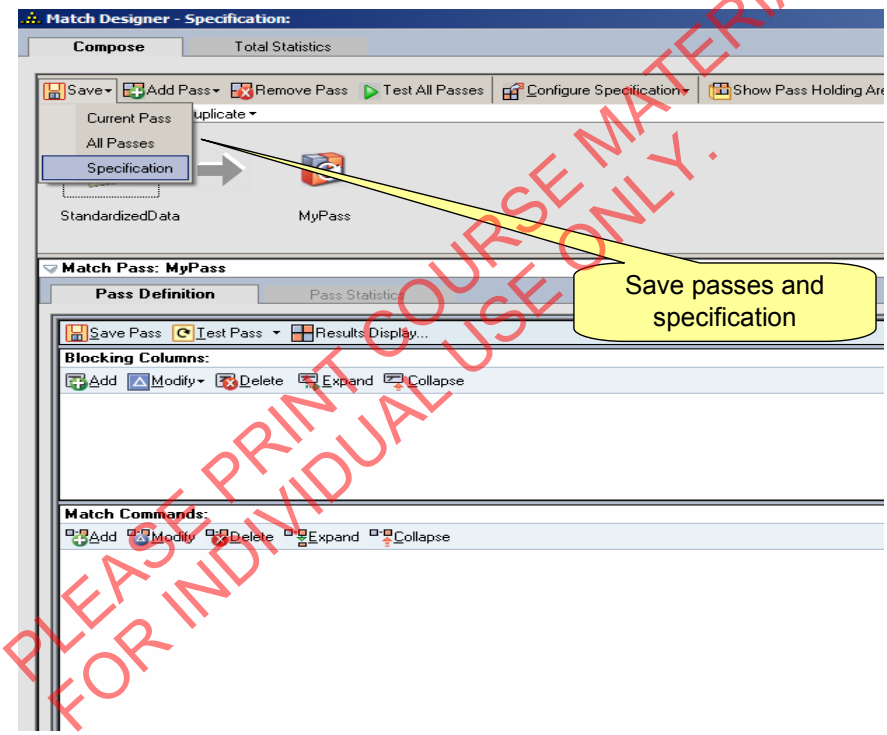
Match design - unduplicate



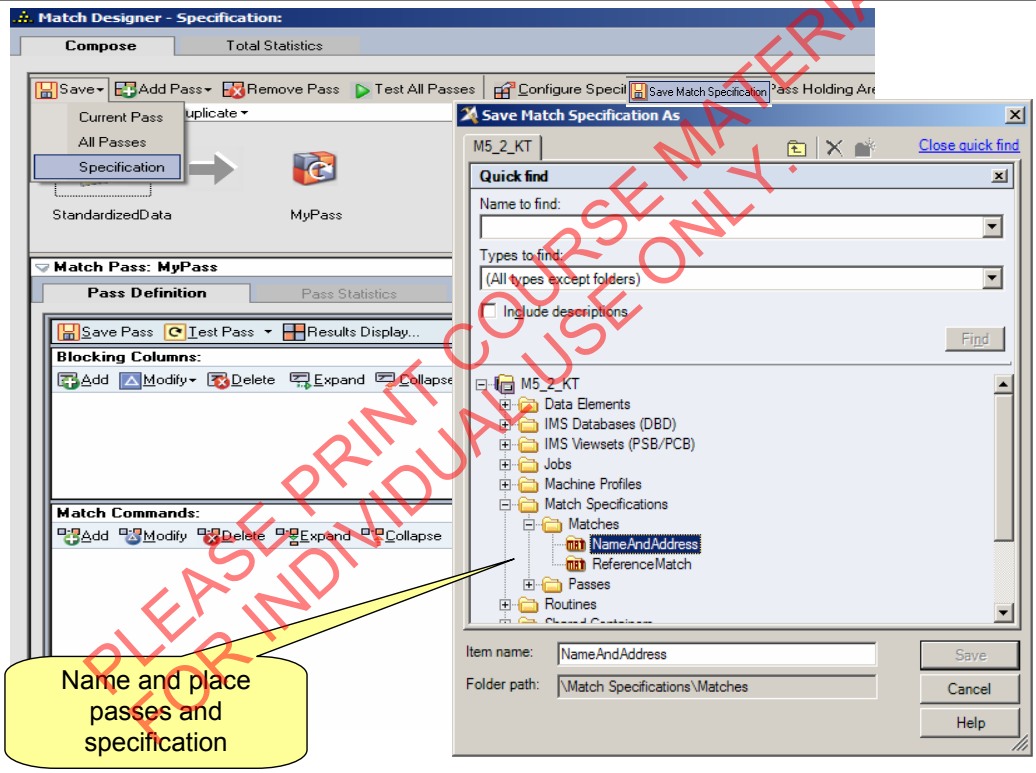
Match design - unduplicate



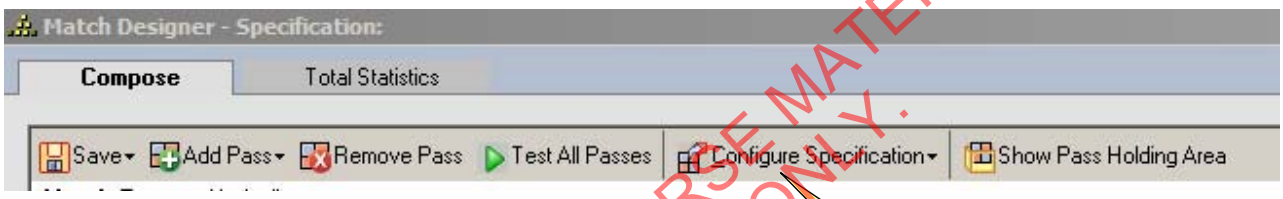
Match design - unduplicate



Match design - unduplicate



Match design - unduplicate

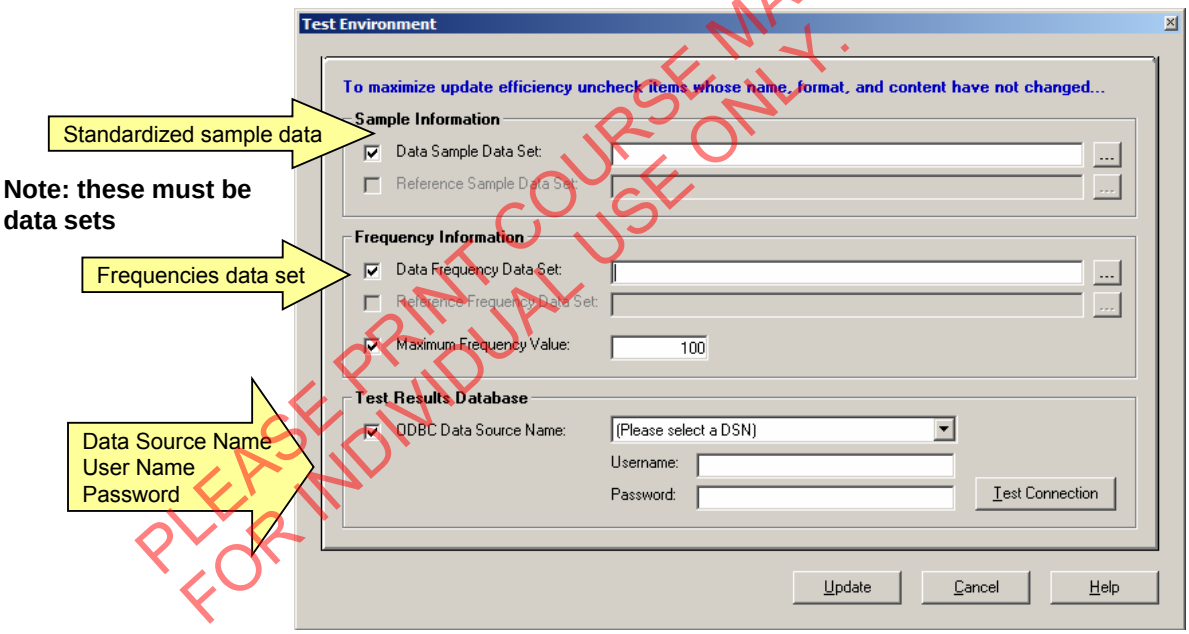


Set up test results area

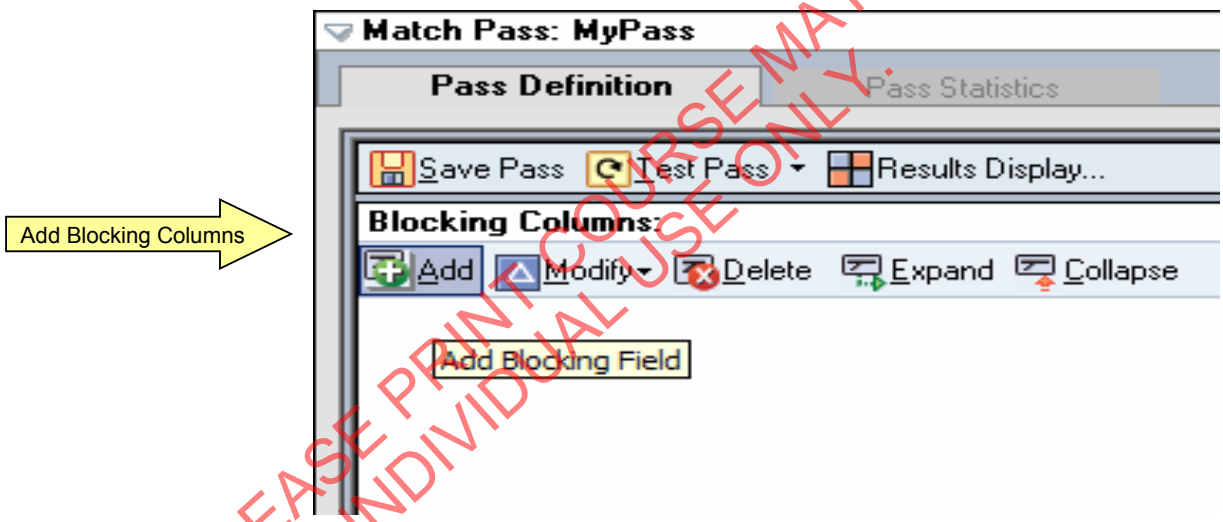
Questions:

- **Where is the standardized data?**
- **Where is the frequency report?**
- **What ODBC-accessed database will store test results?**

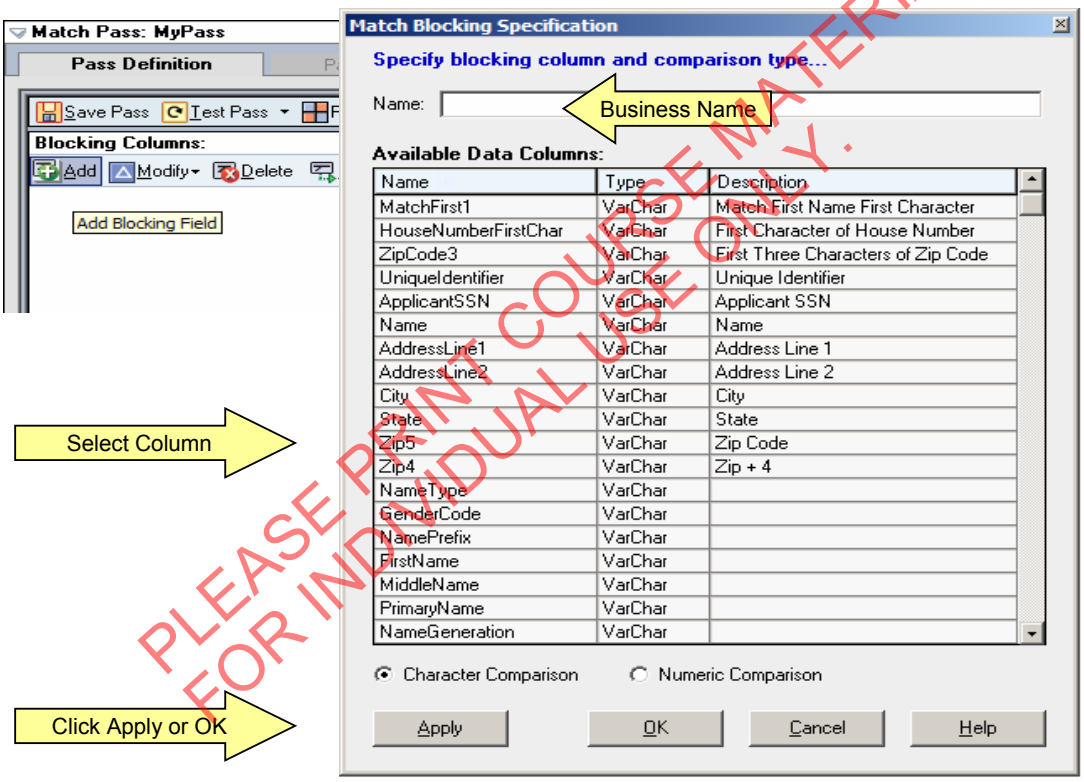
Match design - unduplicate



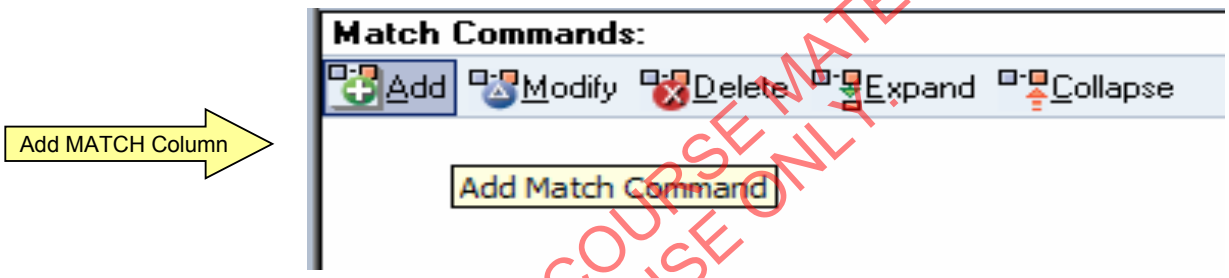
Match design - unduplicate



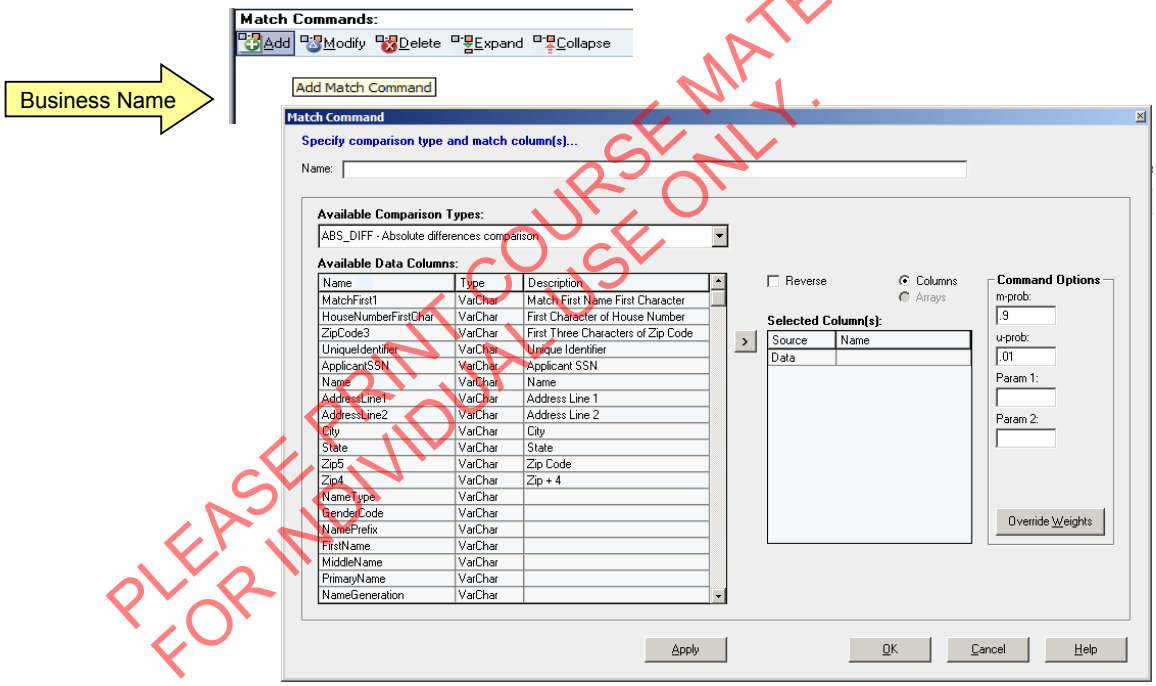
Match design - unduplicate



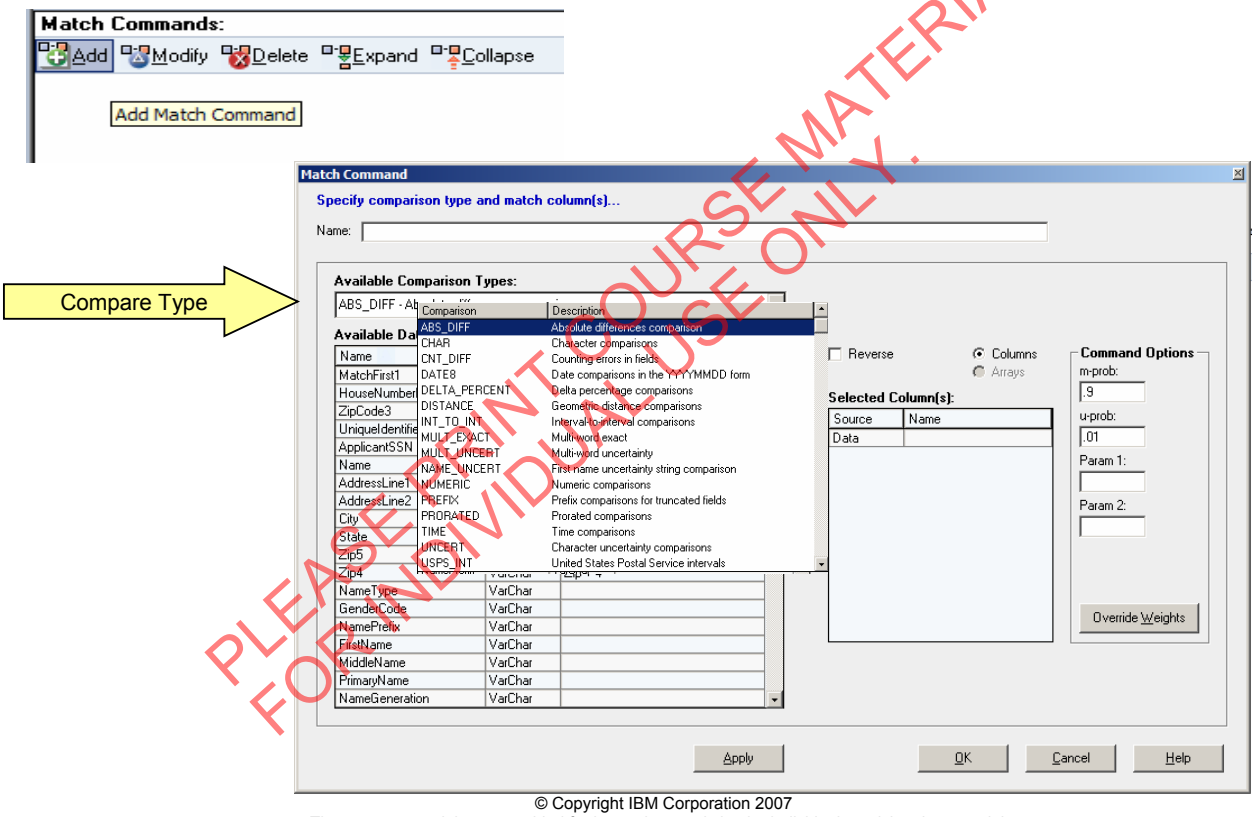
Match design - unduplicate



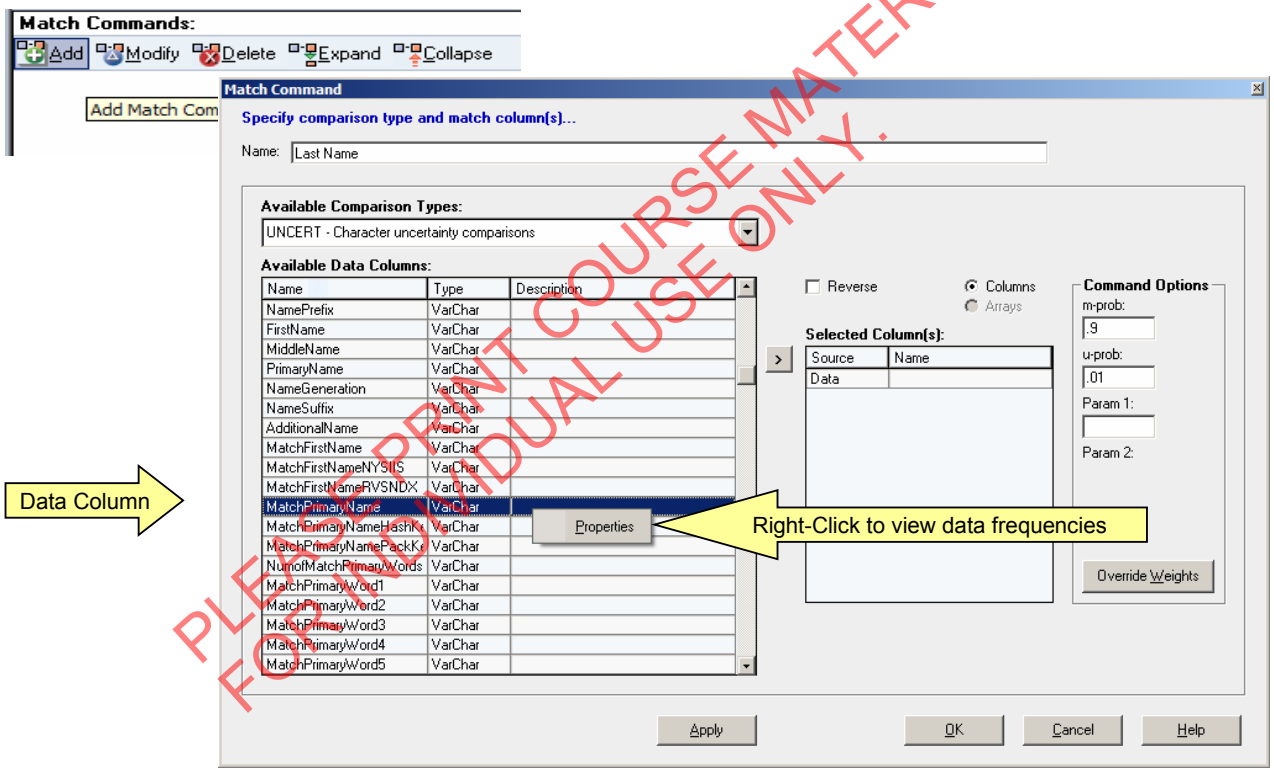
Match design - unduplicate



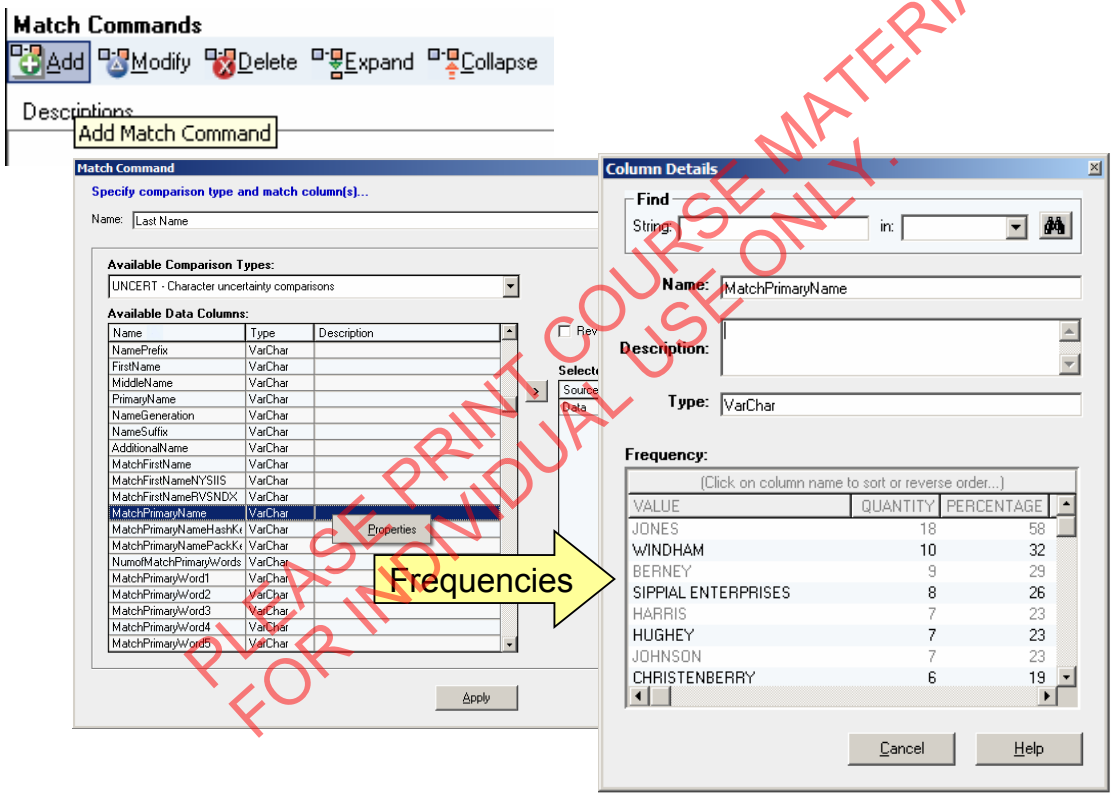
Match design - unduplicate



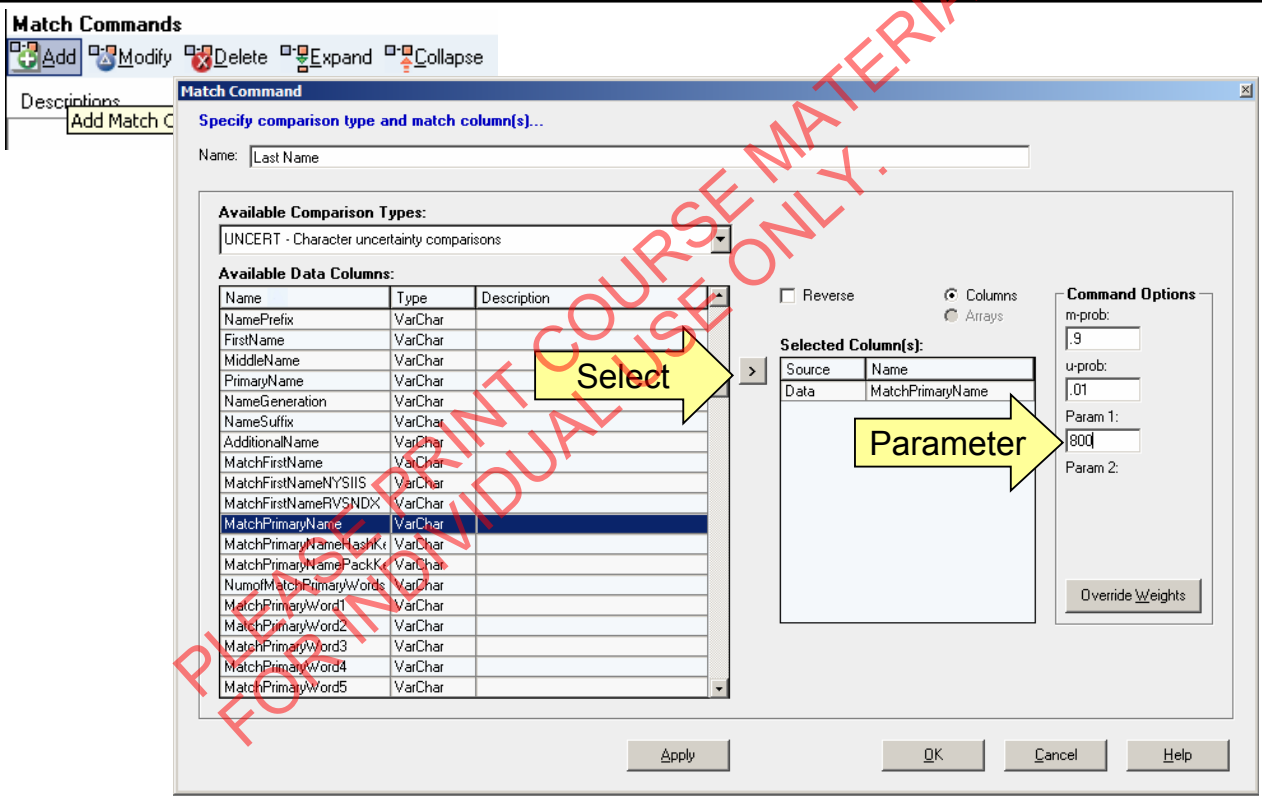
Match design - unduplicate



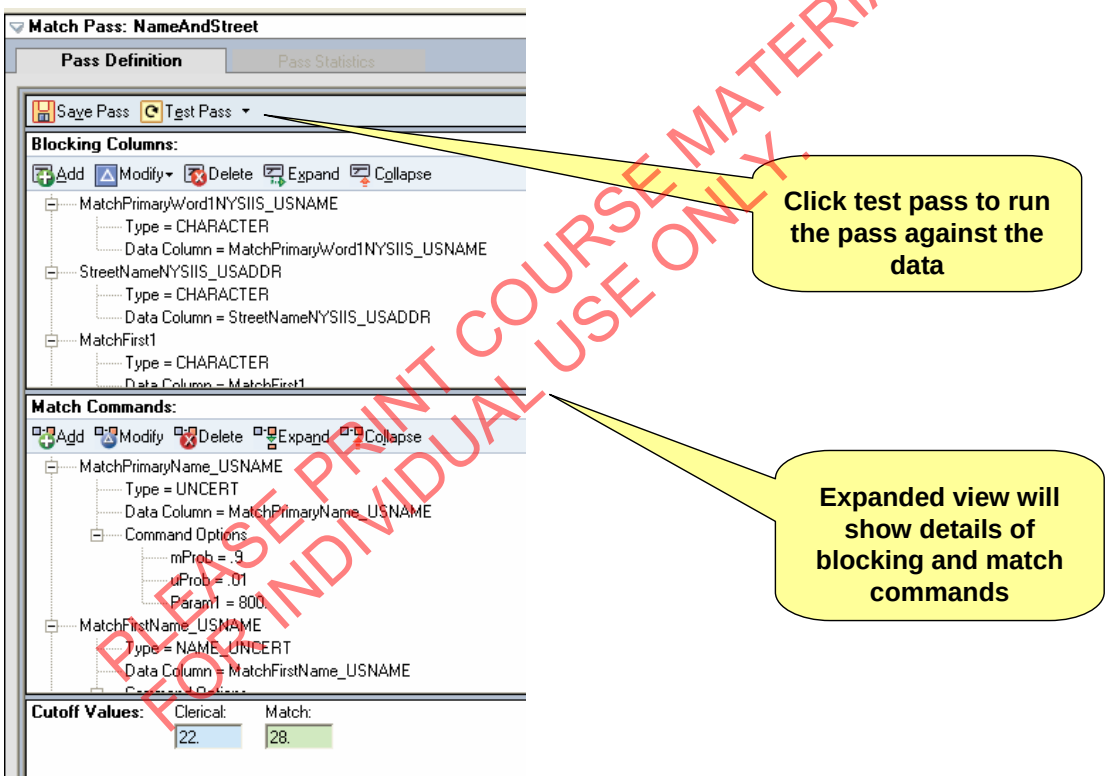
Match design - unduplicate



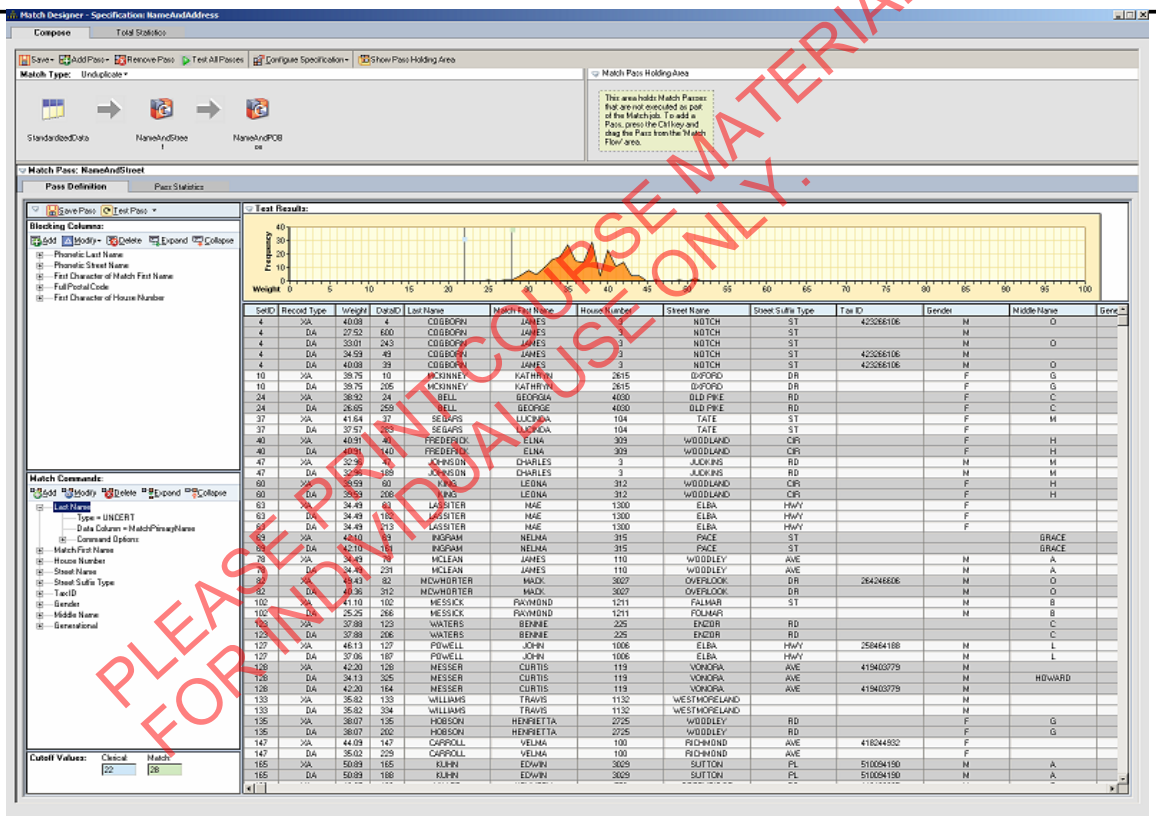
Match design - unduplicate



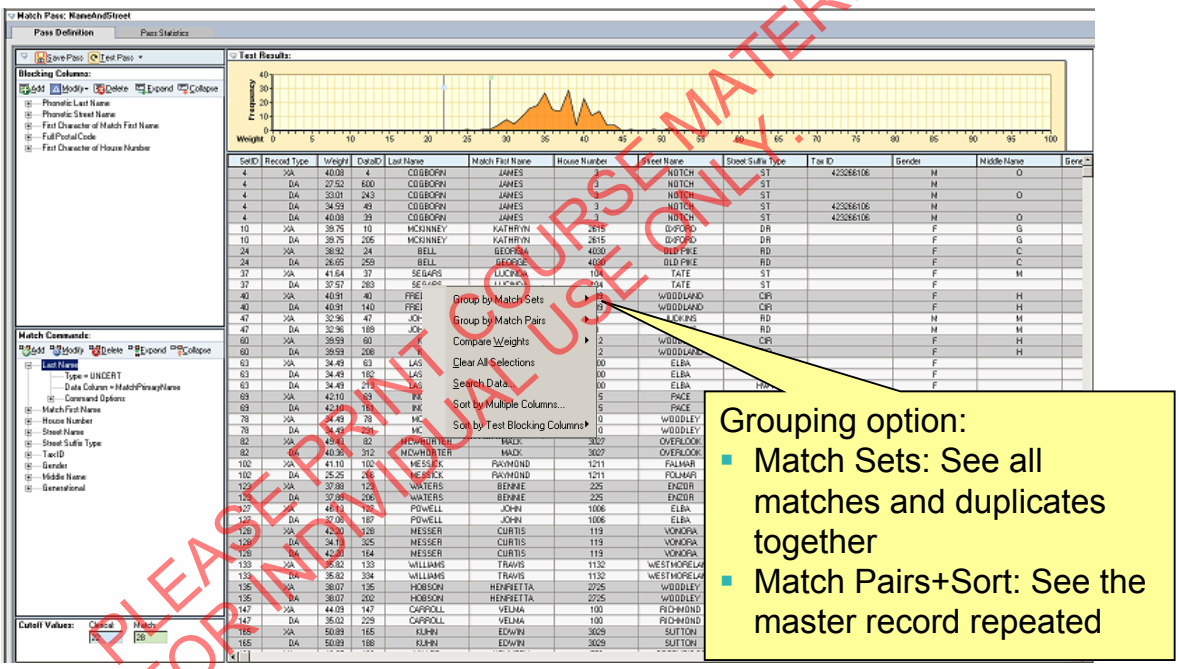
Fully configured pass



Match design – after test pass run



Match design - unduplicate

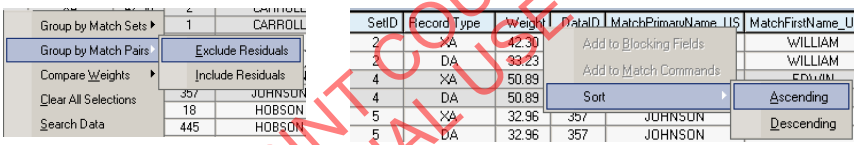


Match design - unduplicate

Default Display (Grouped by Match Sets)

SetID	Record Type	Weight	DataID	MatchPrimaryName_US	MatchFirstName_USNA	HouseNumber_USADD	StreetName_USADDR	StreetSuffixType_U
314	XA	40.08	314	COGBORN	JAMES	3	NOTCH	ST
314	DA	33.01	59	COGBORN	JAMES	3	NOTCH	ST
314	DA	40.08	349	COGBORN	JAMES	3	NOTCH	ST

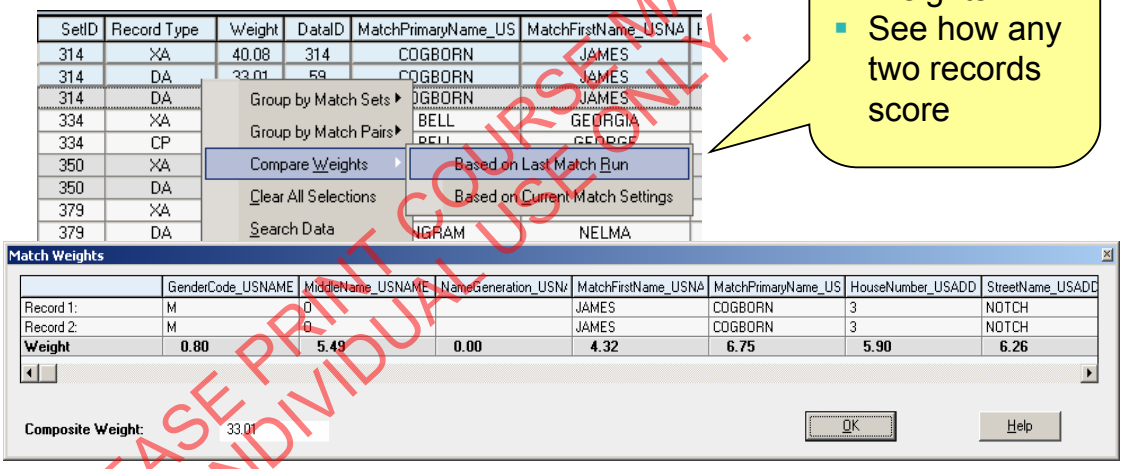
Grouped by Match Pairs and then sorted Ascending by Weight



SetID	Record Type	Weight	DataID	MatchPrimaryName_US	MatchFirstName_USNA	HouseNumber_USADD	StreetName_USADDR	StreetSuffixType_U
5	XA	32.96	357	JOHNSON	CHARLES	3	JUDKINS	RD
5	DA	32.96	357	JOHNSON	CHARLES	3	JUDKINS	RD
314	XA	40.08	59	COGBORN	JAMES	3	NOTCH	ST
314	DA	33.01	59	COGBORN	JAMES	3	NOTCH	ST
582	XA	33.15	583	WINDHAM	AMY	2219	GEORGIA	RD
582	DA	33.15	583	WINDHAM	AMY	2219	GEORGIA	RD

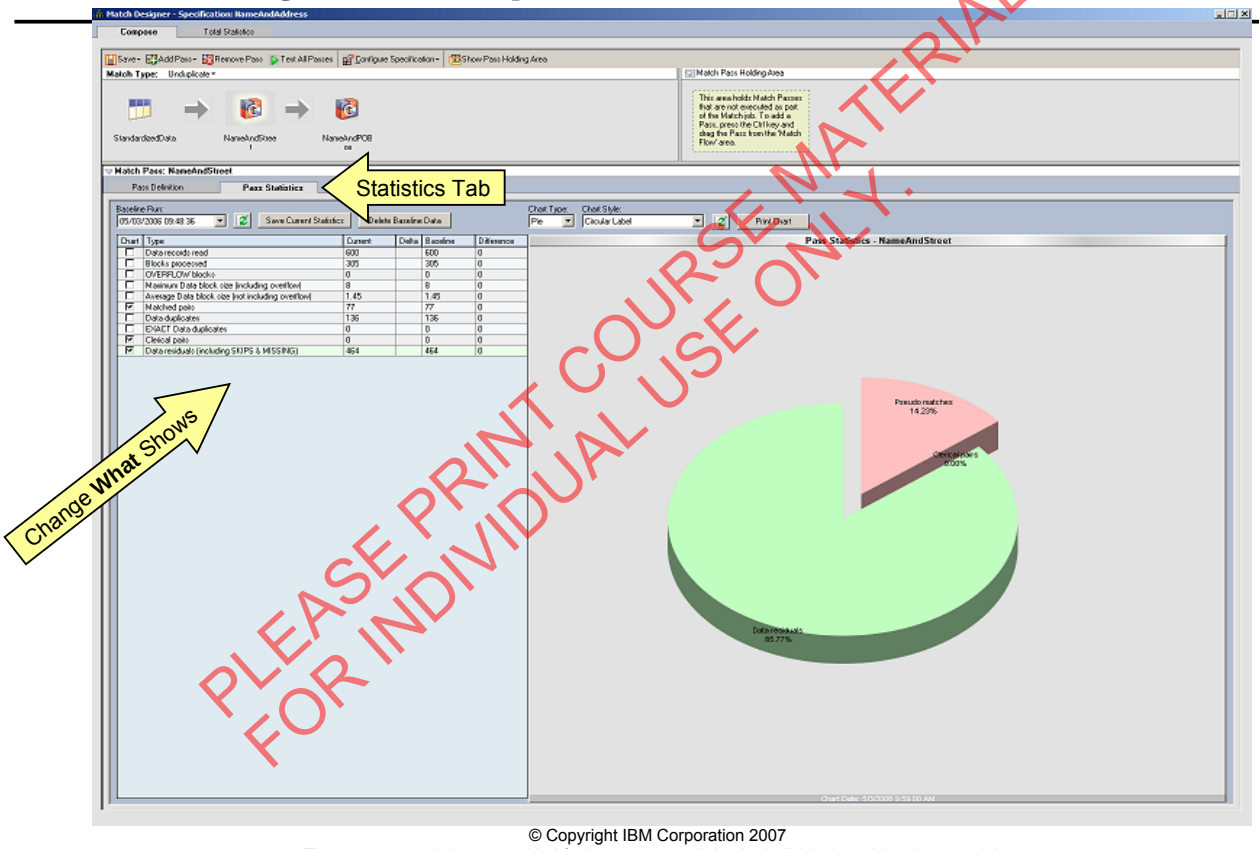
SetID	Record Type	Weight	DataID	MatchPrimaryName_US	MatchFirstName_USNA	HouseNumber_USADD	StreetName_USADDR	StreetSuffixType_U
314	XA	40.08	349	COGBORN	JAMES	3	NOTCH	ST
314	DA	40.08	349	COGBORN	JAMES	3	NOTCH	ST

Match design - unduplicate

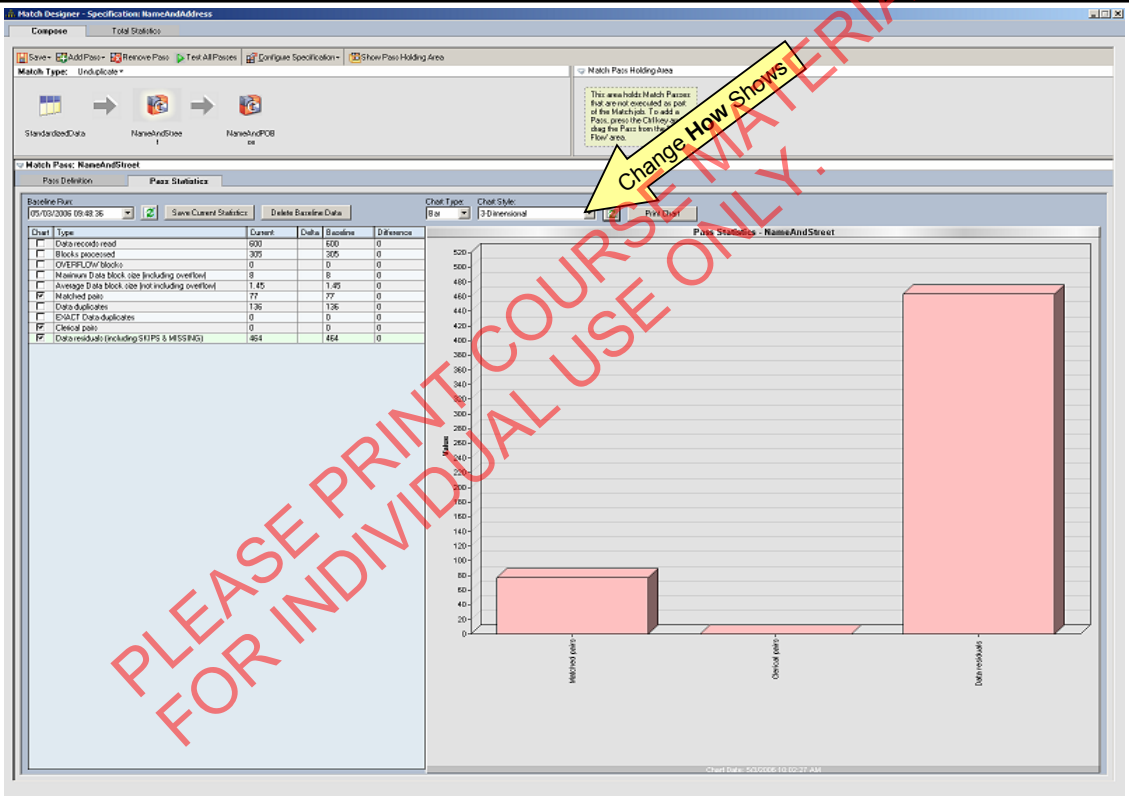


- Compare Weights:
- See how any two records score

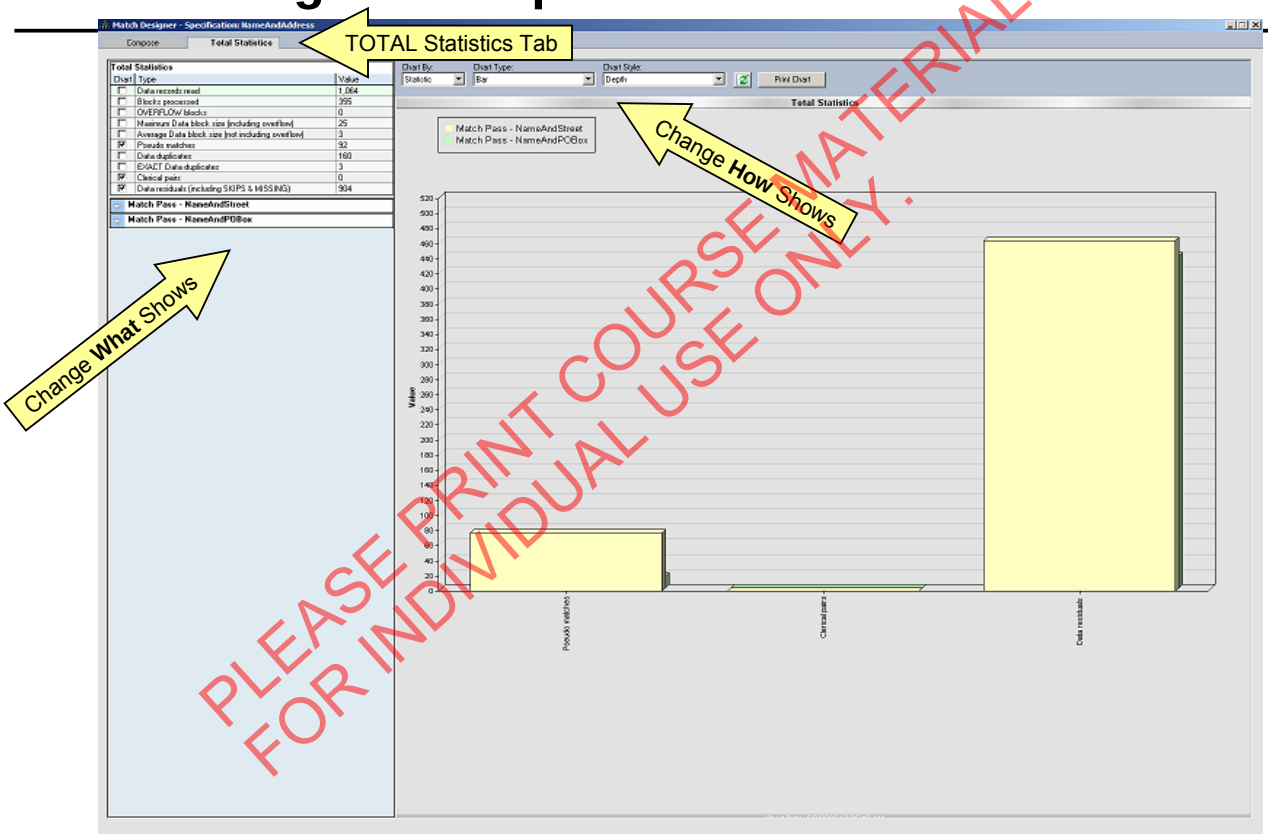
Match design - unduplicate



Match design - unduplicate



Match design - unduplicate

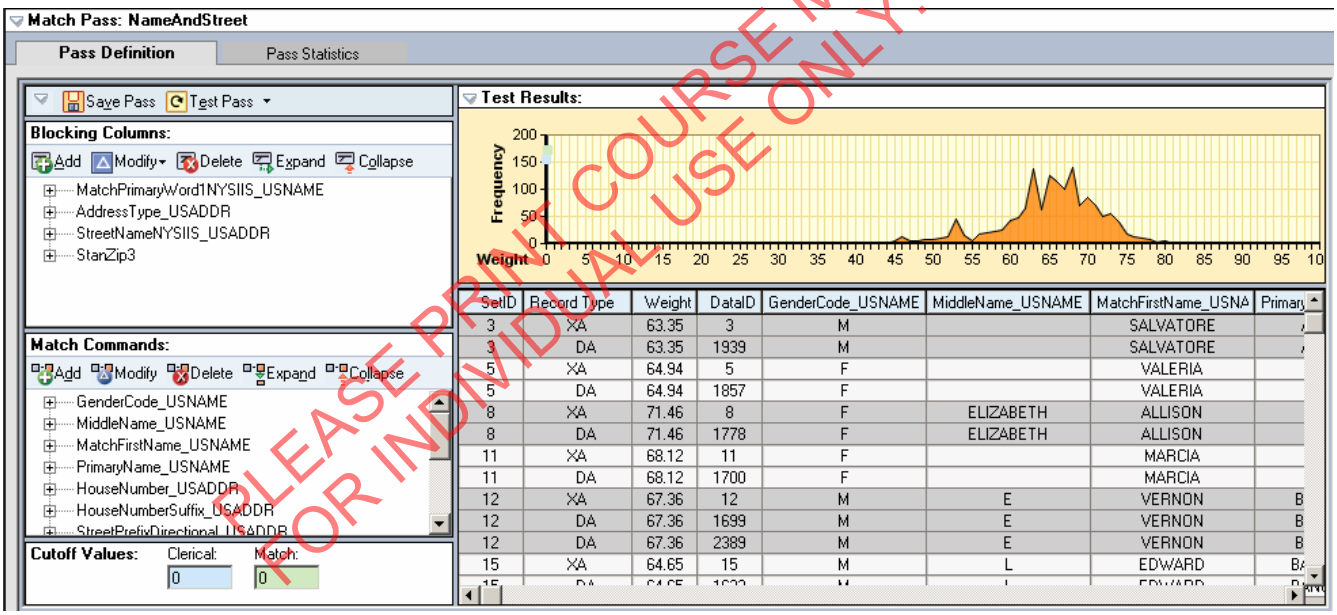


Exercise 18: Configure test results database

- Build a DB2 database to contain match test results
- Build an ODBC source to connect the database to QualityStage

Exercise 19: Match specification

- Use Match Designer to build specification for unduplicate job
- Configure test results area



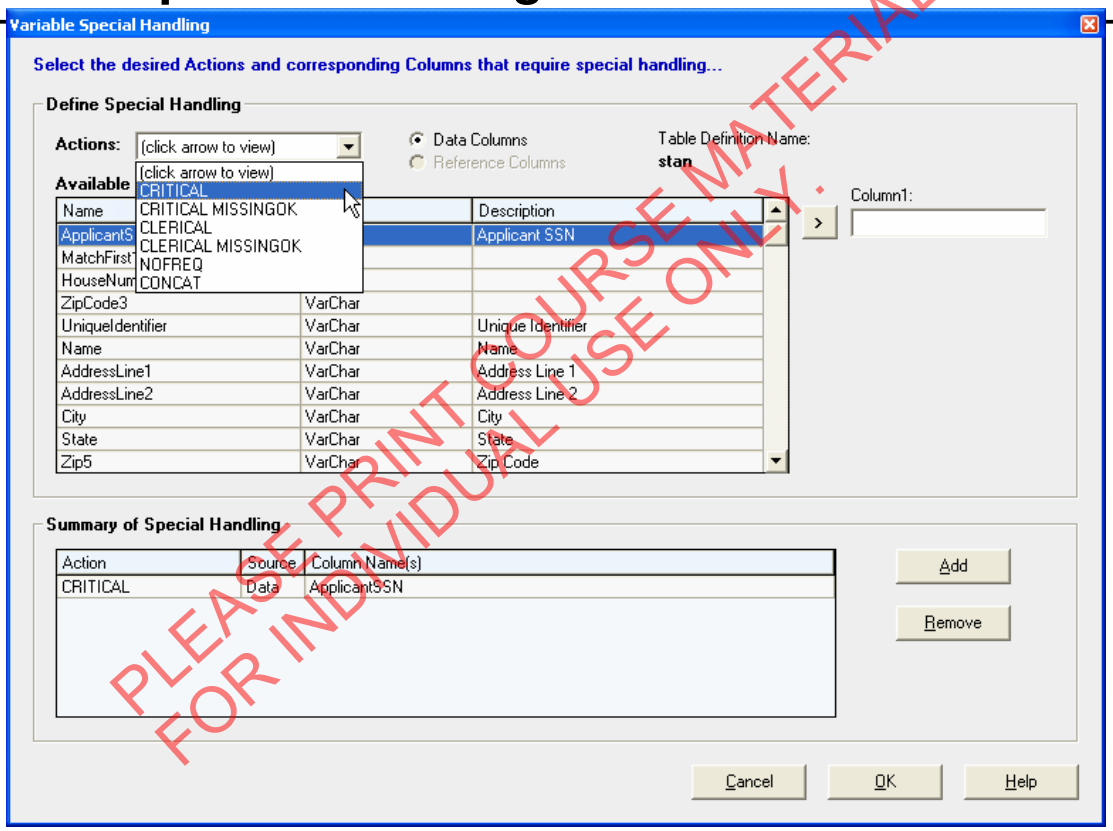
Match improvement strategy

1. Set critical values for important fields
2. Review calculated weights
   - Adjust weights using weight overrides
3. Set cutoffs
4. Add additional passes

Critical fields

- Used to identify fields that must agree in order for records to be linked
  - Critical – Fields values must agree exactly or the records cannot be linked (considered a match)
  - Critical Missing OK – Field values must agree exactly on values not considered “missing values”
- QualityStage feature: VARTYPE

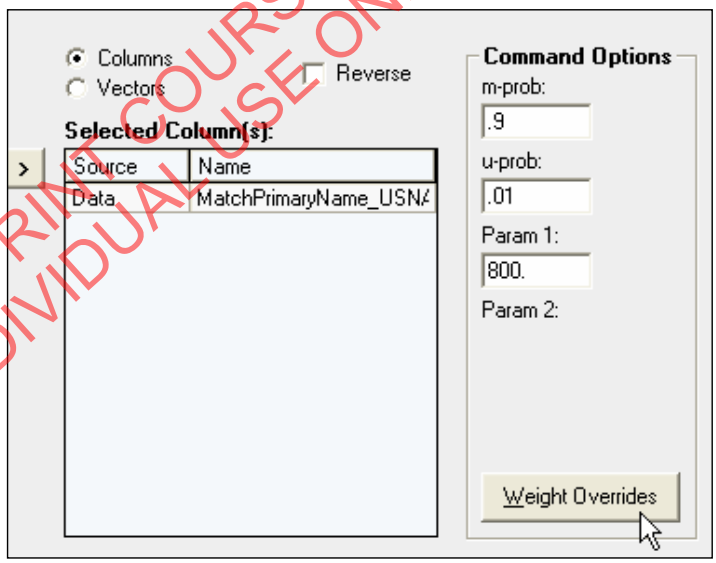
Variable special handling



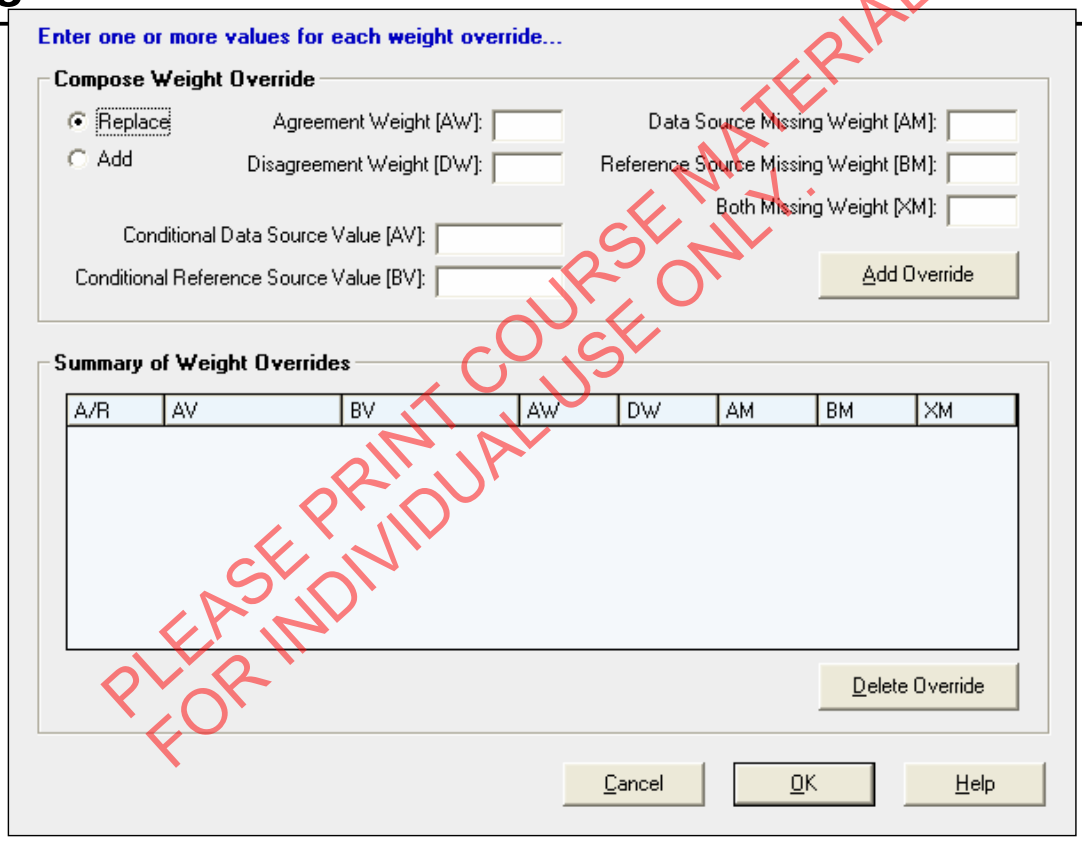
Weight overrides

- Allows you to adjust both the agreement and/or disagreement weights for specific situations
  - Add to calculated weight
  - Replace weight

On Match Commands screen



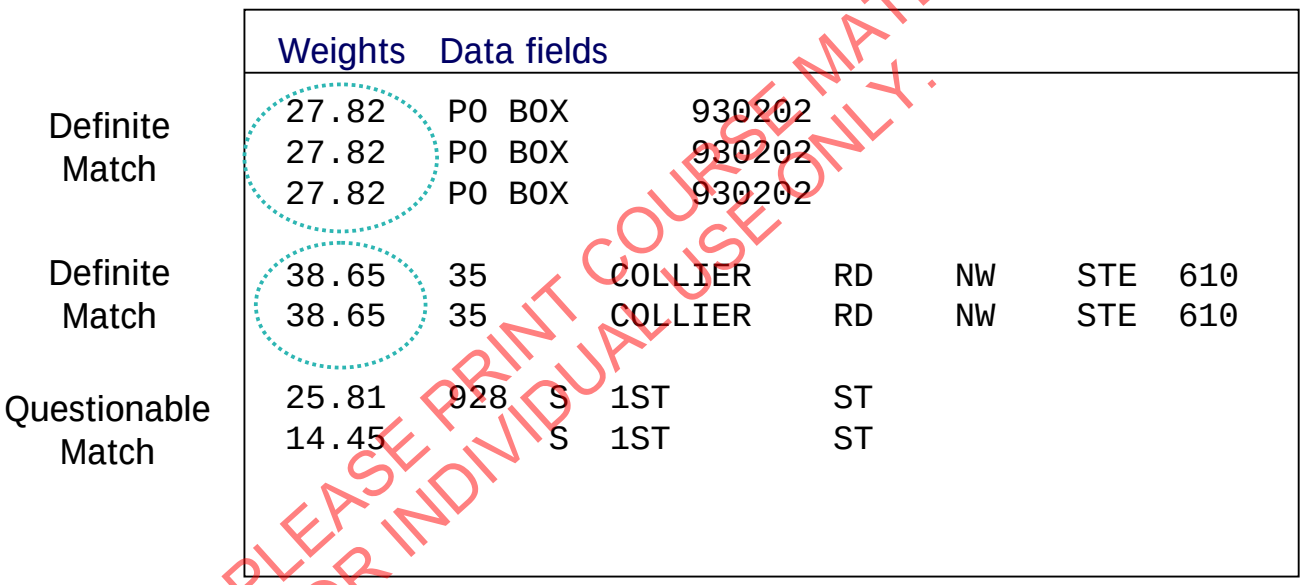
Weight override screen



Cutoffs

- There are two cutoffs
  - Match cutoff (high cutoff)
  - Clerical cutoff (low cutoff)
- Records with a weight equal to or above the Match cutoff are considered matches
- Records with a weight below the low cutoff are not matches
- Records with a weight greater than or equal to the low cutoff and less than the high cutoff are considered clerical records for manual review
- Cutoffs can be set at the same value eliminating clerical records

Setting the match cut-off



Multiple match passes

- Additional passes are helpful in overcoming data errors and missing values in block fields
- You should always create at least two match passes
- Change blocking strategies for each pass

Note: You may create up to seven match passes. Usually 2-3 are sufficient.

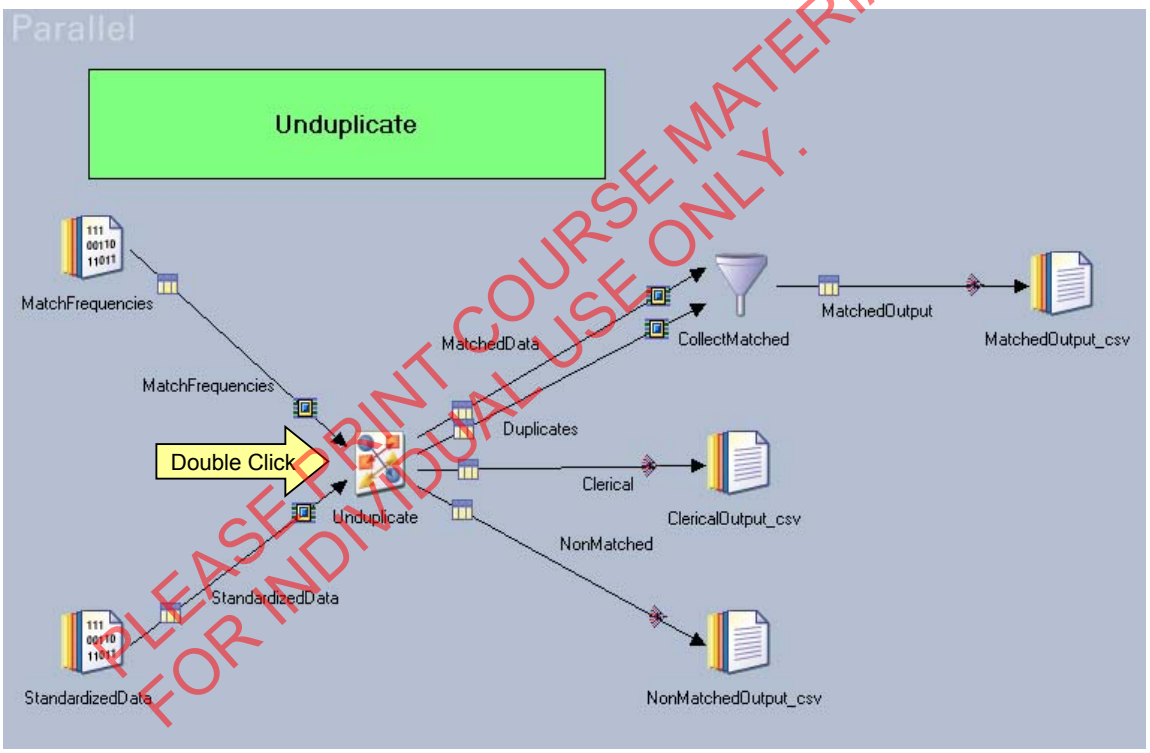
Example: multiple match passes

Pass	Weights	Data fields				
1	26.31	JASON BIRCH	1350 WALTON	WAY	30901	
1	26.31	JASON BIRSH	1350 WALTON	WAY	30901	
1	20.42	JOHN SMITH	2047 PRINCE	AVE	30604	
1	10.83	MARY SMITH	2047 PRINCE	AVE	30604	
1	RES A	JOHN SMITH	P.O. BOX 123		30604	
2	20.42	JOHN SMITH	2047 PRINCE	AVE	30604	
2	10.19	JOHN SMITH	P.O. BOX 123		30604	

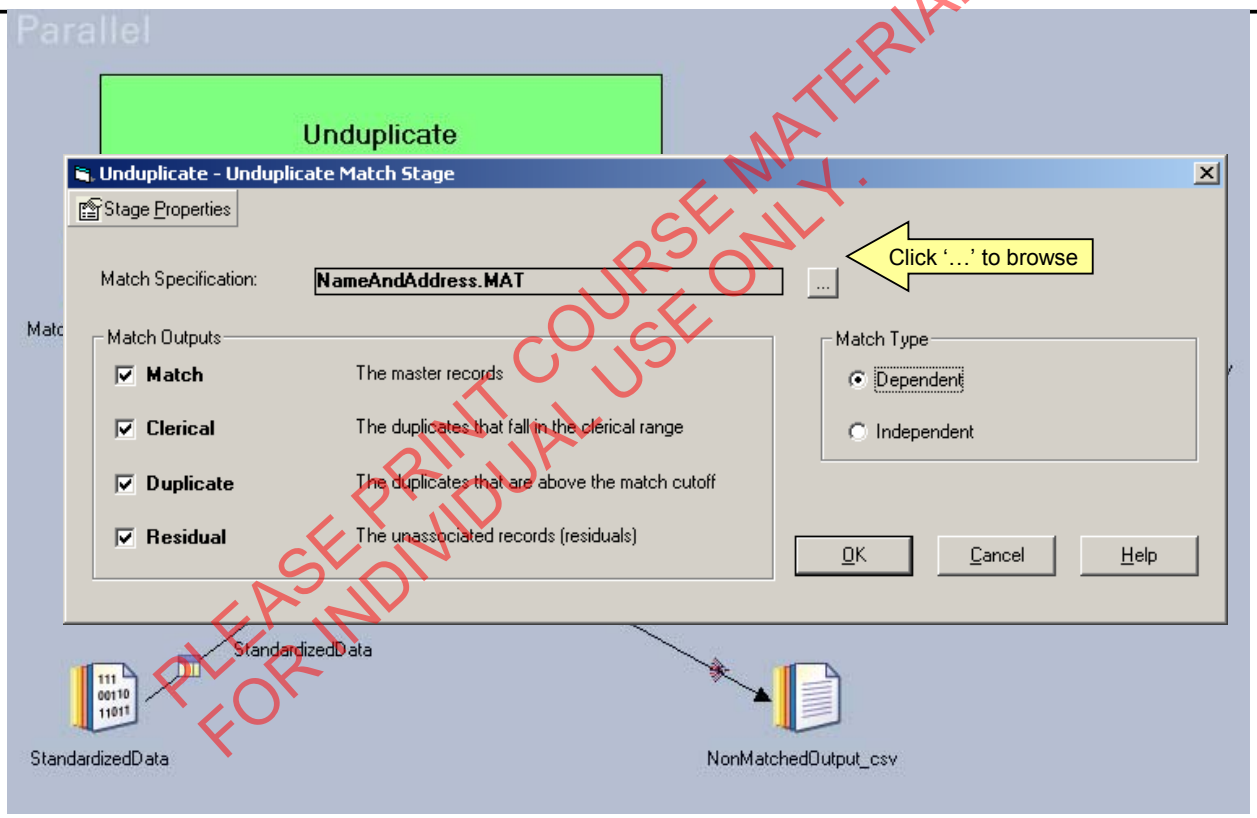
- Pass 1 blocked on street name
- Pass 2 found additional matched records in which the street name was different but the names were the same

Match Implementation – Unduplicate job

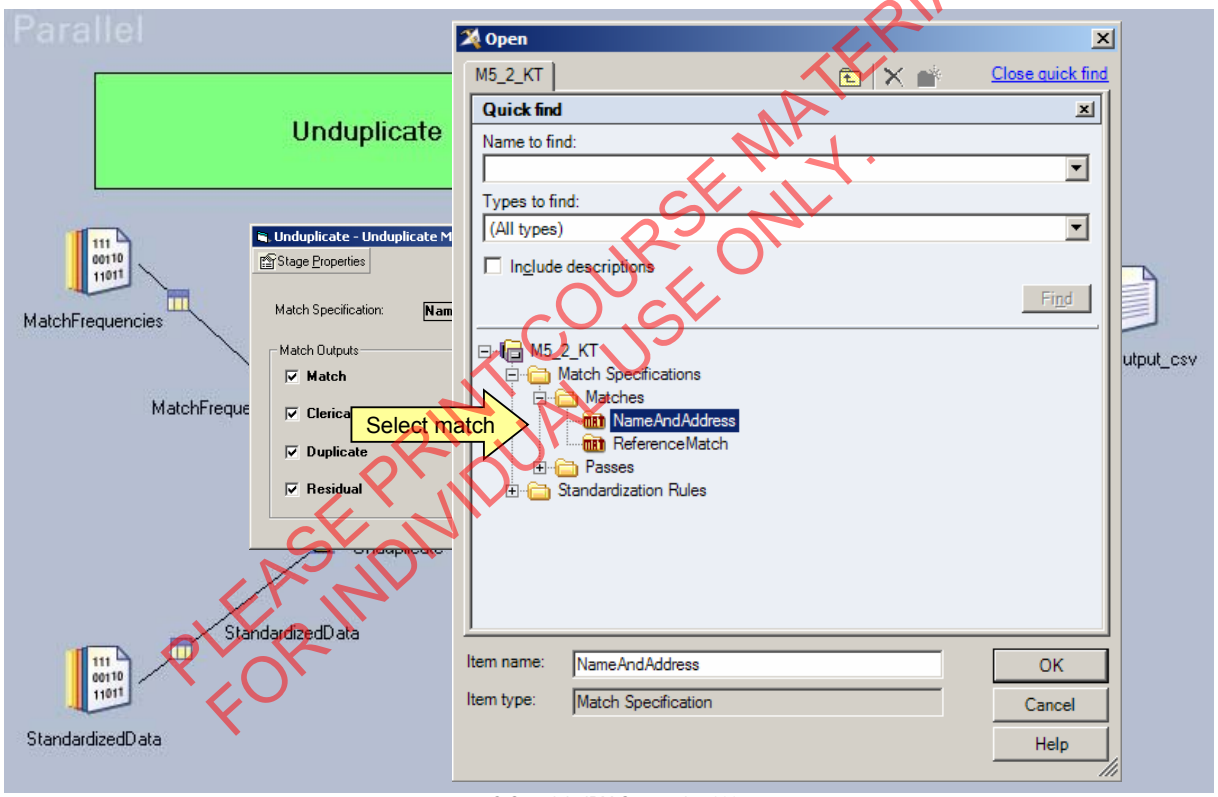
Unduplication Implementation



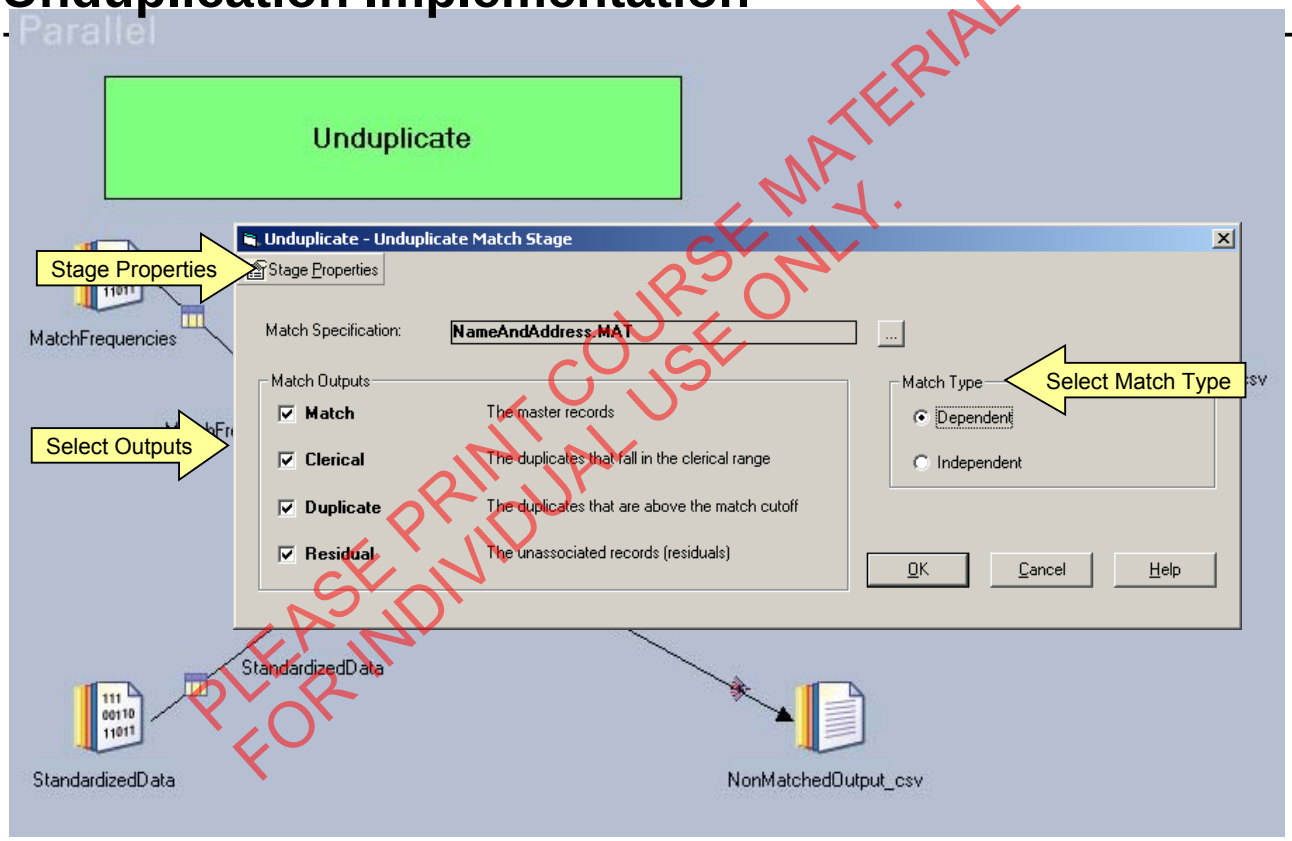
Unduplication Implementation



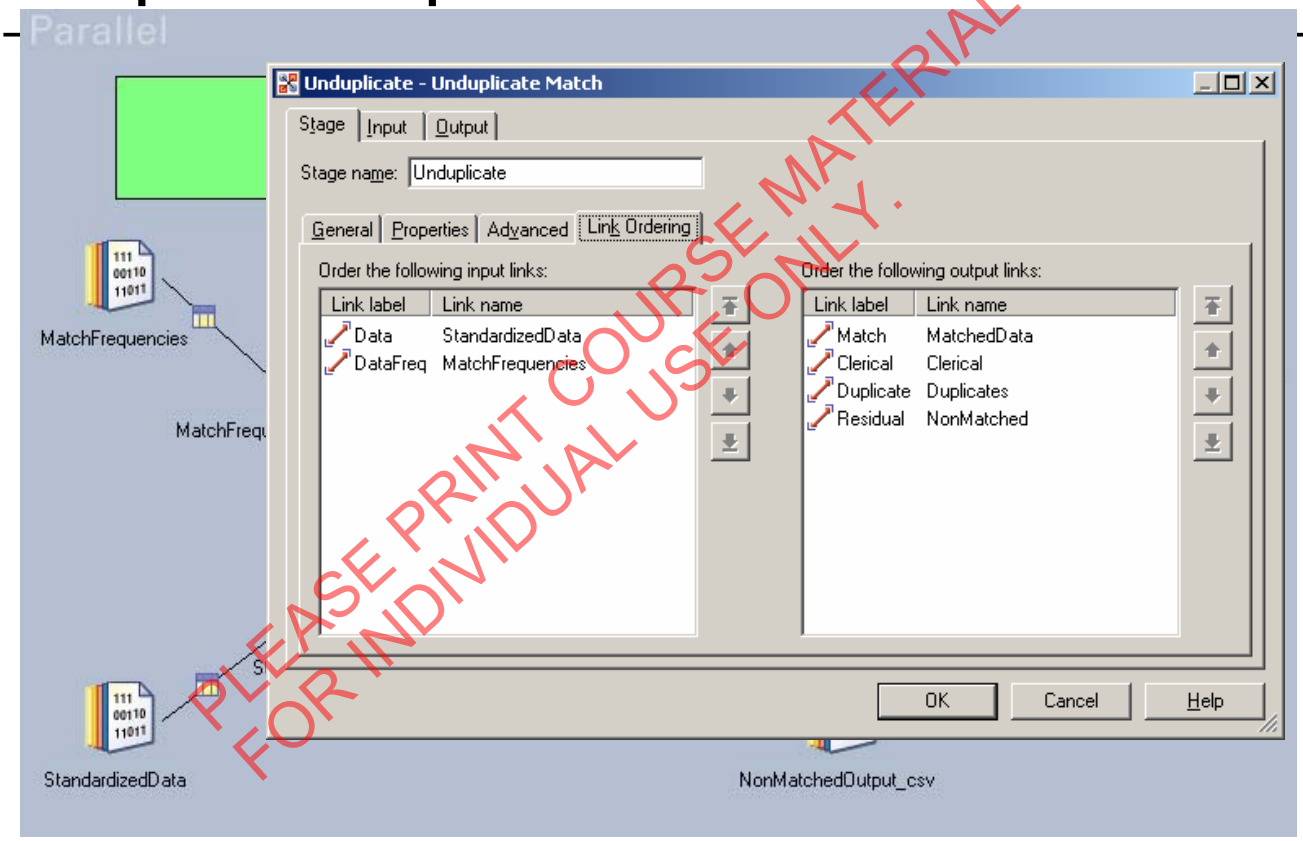
Unduplication Implementation



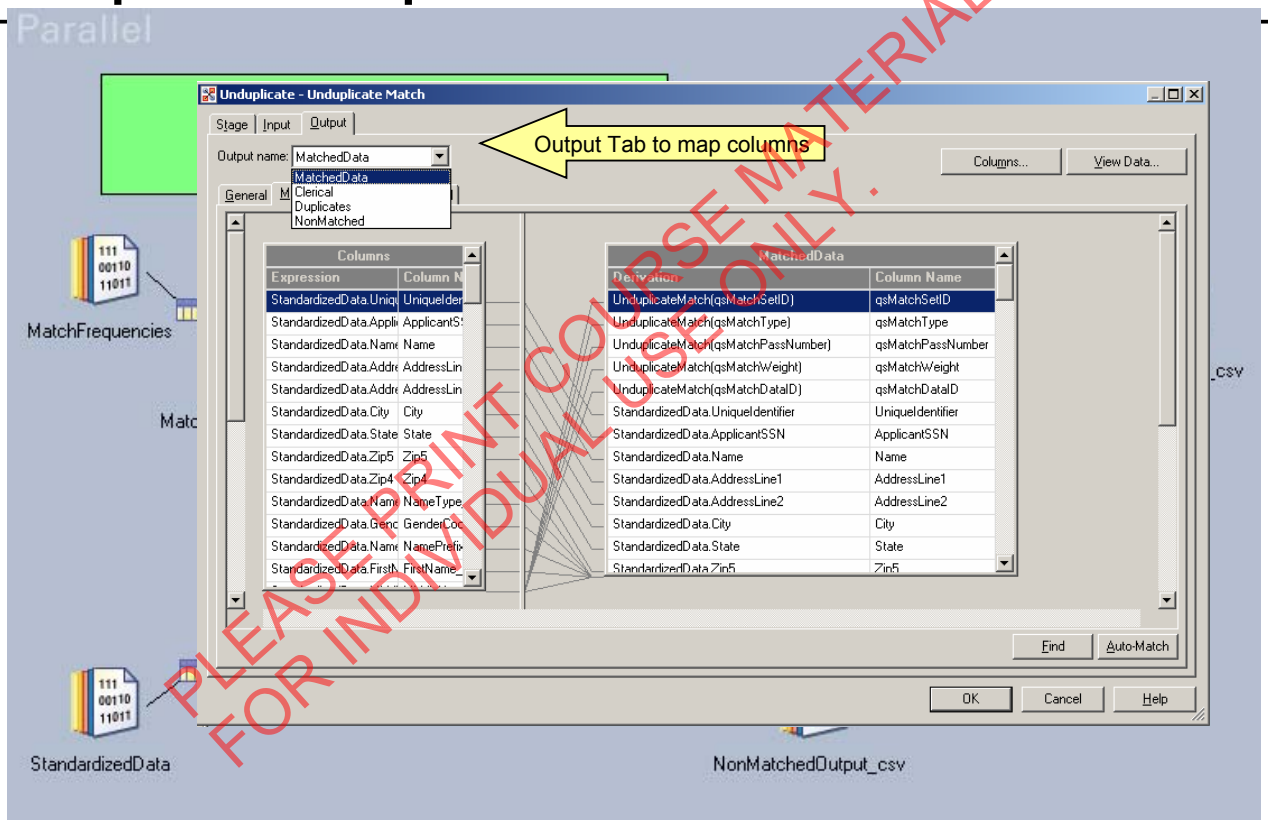
Unduplication Implementation



Unduplication Implementation



Unduplication Implementation



Checkpoint

1. (T/F) Match specifications are created using Designer.
2. (T/F) An unduplicate match can be used against two files.
3. Which match specification component determines the extent of the clerical review records?

Checkpoint solutions

1. (T/F) Match specifications are created using Designer.
   Answer: True
2. (T/F) An unduplicate match can be used against two files.
   Answer: False
3. Which match specification component determines the extent of the clerical review records?
   Answer: cutoff values

Unit summary

Having completed this unit, you should be able to:
- Describe where Match fits in the Data Re-engineering Methodology
- Describe QualityStage Match concepts
- Define the type of matching algorithms
- Describe the importance of blocking
- Apply multiple match passes to increase efficiency/efficacy
- Interpret and improve match results

Exercise 20: Unduplicate

- Build unduplicate job using the match specification