Creating and Maintaining Geocoding Address Locators in ArcGIS

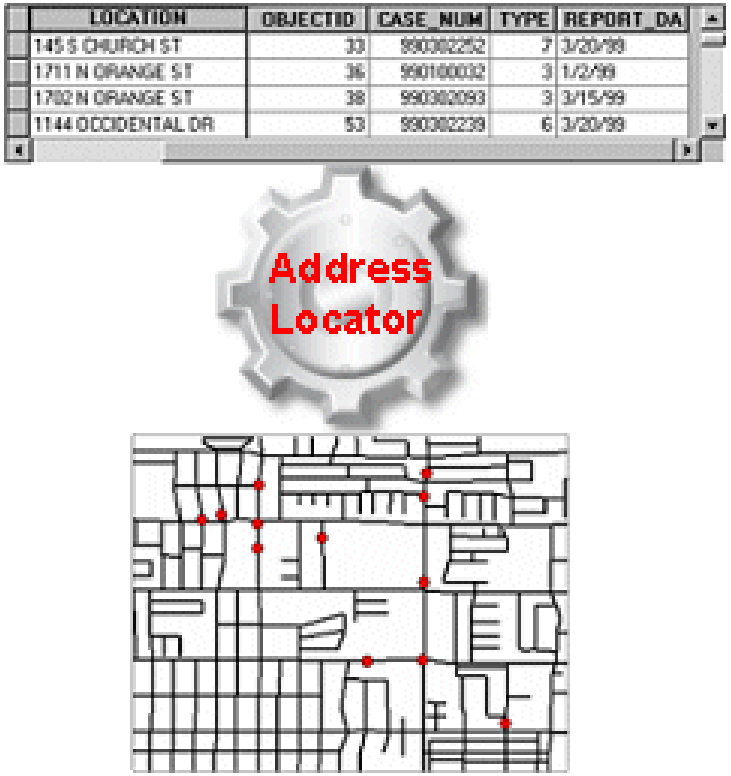
Kenneth Smith – ESRI San Antonio Office

Objectives

- Describe geocoding process
- Building and maintaining reference data
- Determine best address locator
- Creating custom address locator
- Customizing address locator files
- Wrap up

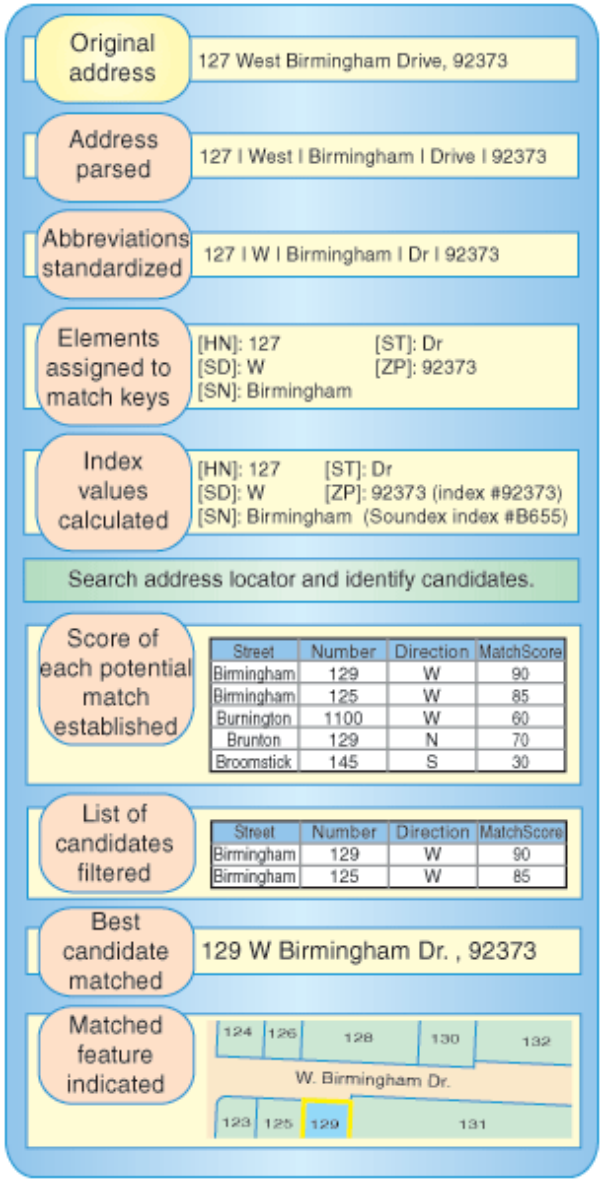
What is geocoding?

- A GIS operation for assigning a location to a street address.
- Result can be displayed on a map as spatial data.



Geocoding process

- Parse Address
- Standardize Address
- Assign address element to category
- Calculate index values
- Search and identify candidates
- Assign score for each match
- Create list of candidates
- Match best candidate
- Indicate best matched feature



Address Locator

- A dataset in the geodatabase used to manage address information for features in order to perform geocoding.
- Address Locator contains
  - A snapshot of the reference data feature class
  - Rules and parameters that define an address format
  - Rules for standardizing and matching addresses
- Address Locator is independent of reference data

Reference data

- A dataset that contains both address and spatial information.
- Used to translate location descriptions into X,Y coordinates.
- Reference data sources include
  - Street centerlines
  - Zip Codes
  - Parcel map

Attributes of Streets

OBJECTID *	Shape *	L_F_ADD	R_F_ADD	L_T_ADD	R_T_ADD	PREFIX	NAME	TYPE	SUFFIX
5	Polyline	0	3626	0	3648		Amherst	Ave	
6	Polyline	3651	3650	3705	3806		Amherst	Ave	
8	Polyline	3209	3210	3309	3308		Amherst	Ave	
9	Polyline	3139	3138	3207	3206		Amherst	Ave	
24	Polyline	2705	2700	2717	2720		Aspen	Way	

Record: 0 Show: All Selected Records (0 out of 79 Selected) Options

Preparing reference data

- Recognize standardized values for abbreviations
  - Verify abbreviations in your location data matches reference data.
  - Use Standardize Addresses tool in ArcToolbox.
  - Use option to standardize reference data on-the-fly when Address locator is created.
- Check for spelling errors in reference data
  - If alternate spelling of address element, consider using alternate place names table.
- Incomplete or outdated reference data
  - Reference data may need to be updated from time to time.

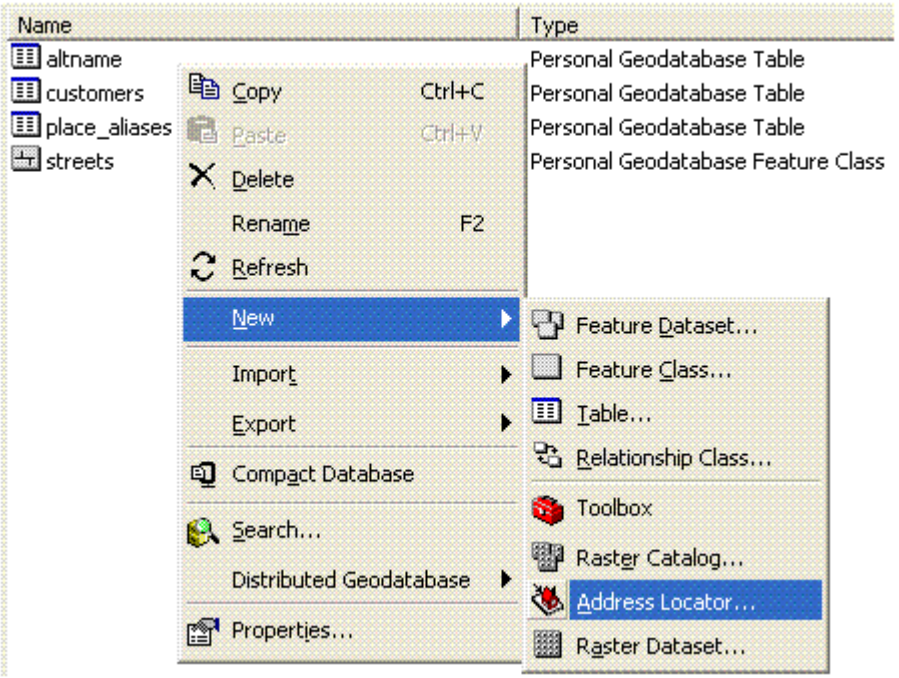
Determining Address Locator Style

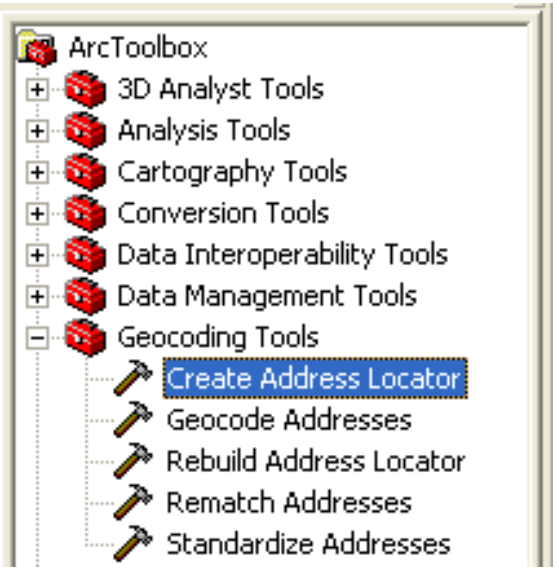
- Which address locator style to use
  - What type of geometry is in reference data
- Format of address data
  - US Streets
  - US Alphanumeric Ranges
  - US Hyphenated
  - US One Address
  - US One Range

Style	Geometry	Representation	Search Params	Example
US Streets	Polyline	Left and right side	Address in one fld	234 N Main St.
US Alphanumeric	Polyline	Address with Grid code info	Address in one fld	N2W1700 County Rd
US Hyphenated	Polyline	Address with cross-street	Address in one fld	105-25 Union Blvd
US One Address	Polygon/Point	One address per feature	Address in one fld	71 Cherry Ln
US One Range	Polyline	One address per feature	Address in one fld	71 Cherry Ln

Creating a custom address locator

- Create Address Locator in ArcCatalog or ArcToolbox
- Can be created inside of geodatabase or file location
- Choose the address style to work with





Modifying address locator rule base files

- Accommodate additional elements
- Address locator rules files
  - .mat – match rules
  - .dct – match key dictionary
  - .stn – standardization commands
  - .cls – classification table
  - .pat – pattern rules and actions

Match Rules

- Defines address fields from reference data for matching
- Defines method for address-to-reference data comparing
- Defines weights for each address field
- Defines probabilities for score comparison of candidates
  - m-probability compares candidate address with original address
    - North in candidate address would get a higher score if North was also in original address than if West was in original address and North was in candidate address
  - u-probability randomly compares candidate with original

Match file components

- **VAR, MATCH, and VARTYPE commands**

The Match (.mat) file

```
; @(#)us_addr1.mat
;
; Full geocoding match rules with left and right zip codes
;
VAR LeftFrom    1 10  X ; Left from house number
VAR LeftTo     11 10  X ; Left to house number
VAR RightFrom  21 10  X ; Right from house number
VAR RightTo    31 10  X ; Right to house number
VAR PreDir     41  2  X ; Prefix direction
VAR PreType    43  6  X ; Prefix street type
VAR StreetName 49 30  S ; Street name
VAR StreetType 79  6  X ; Suffix street type
VAR SufDir     85  2  X ; Suffix direction
VAR LeftZone   87 20  X ; Left zone
VAR RightZone 107 20  X ; Right zone
;
MATCH LR_UNCERT ZN LeftZone RightZone 0.9 0.01 700.0 EITHER
MATCH UNCERT  SN StreetName 0.9 0.01 700.0
MATCH CHAR    PD PreDir     0.8 0.1
MATCH CHAR    PT PreType    0.7 0.1
MATCH CHAR    ST StreetType 0.85 0.1
MATCH CHAR    SD SufDir     0.85 0.1
MATCH D_INT   HN LeftFrom LeftTo RightFrom RightTo 0.999 0.05 ZERO_VALID
;
VARTYPE LeftFrom NOFREQ
```

VAR commands

MATCH commands

u probability

m probability

VARTYPE command

VAR Command

- VAR commands specify variable names, field position, and missing codes in the match file.
- VAR format:

  <variable name> <begin column> <length>

  <missing-value code> -

  ; comments

  where <missing-value code> S = spaces, Z – zero or no spaces, N = negative numbers, 9 = all nines, X = no missing values

  Example: `VAR StreetName 37 28 S ; Street Name`

MATCH Command

- MATCH commands specify comparison types, match key field, variable name, probabilities, and additional parameters
- MATCH format:

  <comparison-type> <match key field> <reference file var name>

  <m-probability> <u-probability> [<additional parameters>]

  [<mode>]

  where <comparison types> CHAR = Character, D_Int = Left/Right intervals, LR-CHAR = Left/Right Characters, LR_UNCERT = Left/Right uncertanty, NUMERIC = Numeric, UNCERT = uncertantity, INTERVAL_NOPAR = interval without parity

MATCH Command (cont)

- **MATCH format:**

  where :

  <match key field> is the two character match key field from the .dct file

  <reference file var name> is the variable name defined in the `VAR` command

  <m-probability> is the probability the field agrees

  <u-probability> is the probability the field randomly agrees

  [<additional parameters>] are for `UNCERT` and `LR_UNCERT`.

  [<mode>] is for `D_INT`

MATCH Command (cont)

- **MATCH format examples:**

```
MATCH LR_UNCERT ZN ZipLeft ZipRight 0.9 0.01 800 Either

MATCH CHAR PD PreDir 0.8 0.1

MATCH CHAR SD SuffixDir 0.7 0.1

MATCH D_INT HN FromLeft ToLeft FromRight ToRight 0.999 0.5
  ZERO_VALID
```

VARTYPE Command

- **VARTYPE command indicates if frequency analysis is not performed on a field**
- **VARTYPE format:**

  <match variable name> <action>

  where <action> = NOFREQ

  Example: `VARTYPE FromLeft NOFREQ`

m and u probabilities

- **MATCH commands all have m and u probabilities**
- **m probability is the probability that a field in the original address matches the standardized address, given a match. If StreetName mismatches 10 percent of the time, then the m probability should be set to 0.90 (1 - .10)**
- **u probability is the probability that a field in the original address matches the standardized address, given that both are unmatched randomly. This will usually be a very low value, so set to 0.01 or 0.1**
- **Example:** `MATCH CHAR PD PreDir 0.90 .01`

Candidate Scoring

- **Candidate scores are calculated from the field weights**
- **Field weights are calculated from the ratio of the m & u probabilities.**
  - **`log²m/u` if there is a match, `log²1-m/1-u` if there is no match**
- **Example:**

Candidates					Composite Score
101 +	199 +	N +	MAIN +	ST +	100
101 +	199 +	 -	MAIN +	ST +	90
101 +	199 +	N +	MAIN +	AVE -	85
101 +	199 +	 -	MAIN +	 -	60

Standardization file (.stn)

- The standardization file defines the commands used to parse and standardize an address. There is one .stn file for each address locator.
- Components:

  RECORD <recordsize> - always 256

  TYPE <file-type> - always ASCII

  INTERACTIVE

  {DEBUG} – optional

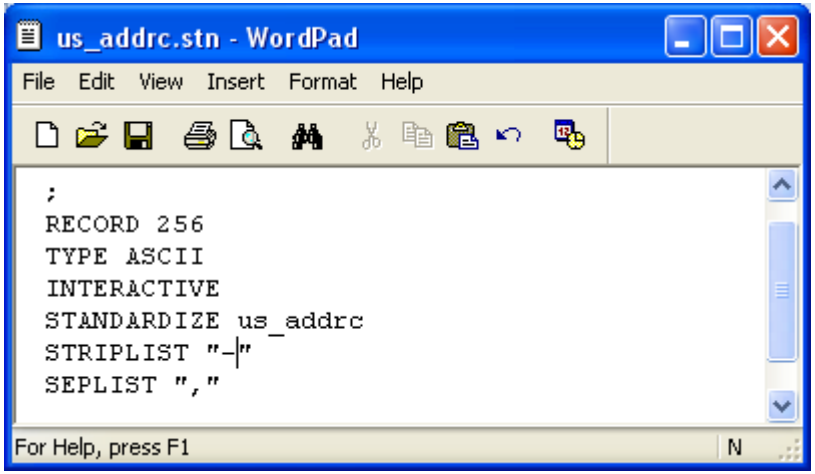
  STANDARDIZE < process> - example: STANDARDIZE us_streets

  {OUTFILE <output-file>} – optional, results of pattern match

  {<parsing-parameters>} – these override the defaults

Standardization file (.stn)

- **Example .stn file for us_addr locator:**



- **STRIPLIST contains characters that will be stripped from the address. Default characters include `,.\;:'`**
- **SEPLIST contains characters that tokenize the address. Default characters include `()-/,#&;:`**

Match Key Dictionary (.dct)

- The match key dictionary file defines information for the match key field in the pattern file.
- Components:

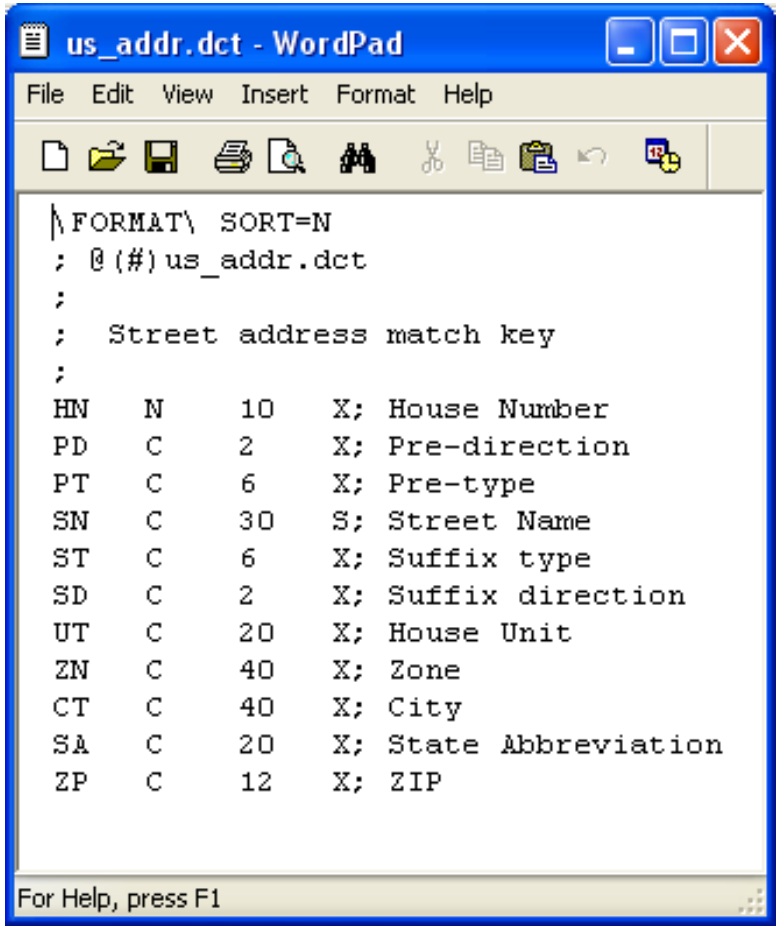
  <field-identifier>

  <field-type>

  <field-length>

  <missing-code value>

  <comments>



Match Key Dictionary (.dct)

- **Component definitions:**

  <field-identifier> is a two character unique field name

  <field-type> defines how the information is placed in the field.
  C=character, left justified, filled with trailing blanks

  N=numeric, right justified, filled with leading blanks

  NS=numeric, leading zeros are stripped off

  M=mixed alpha-numeric, alpha is left justified, numeric is right justified, leading zeros are retained.

  MN=mixed name, where field values starting with character are left justified, field values starting with number are indented

  <field-length> defines the field length in characters

  <missing-code value> where X = no missing code, typical

  <comments> starts with a semi-colon, any characters can follow

Classification table (.cls)

- The classification table (.cls) is used in the address standardization process. You can add or change how words in your reference file are standardized by modifying the .cls file.
- The .cls file contains keywords that are used to identify and classify parts of an address, such as street types (ST, AVE, BLVD, HWY) or directions (N, S, W, E).
- Classification Table Components:

  <keyword> <standardized-abbreviation>

  <keyword-class> <comparison-threshold>

Classification table (.cls)

- **Component descriptions:**

  <keyword> - single word, no spaces, comment if not used

  <standardized abbreviation> - abbreviation used to standardize address

  <keyword-class> - used in pattern matching to specify the rules

  values are: 0 – null, D – direction, T – street type, M – multi-unit,

  B – P.O. Box, O – ordinals (FIRST, SECOND), C – cardinals (ONE, TWO)

  <comparison threshold> - degree of uncertanty, values are:

  900 – exact match, 800 – almost the same characters,

  750 - probably the same characters, 700 – probably different

Classification table (.cls)

- **Example .cls table – us_addr.cls**

us_addr.cls - Notepad

File Edit Format View Help

```
\FORMAT\ SORT=Y
; @(#)us_addr.cls
;
; Explanation of classes
;
; 0 = NULL word (THE, OF)
; B = Box  (BOX)
; Q = Post  (POST)
; Y = Office  (OFFICE)
; K = FPO APO GENDEL
; L = OLD
; M = Multiunit (APT)
; E = Building type
; F = Floor
; G = Directional modifier (END POINT VIEW) for
EAST END
; C = Cardinal number (ONE TWO)
; O = Ordinal number (FIRST SECOND)
; D = Direction (NORTH)
; T = Street type (ST AV)
; R = Rural route (RR)
; X = Route modifier (US, STATE)
; S = St
; N = Number which may be followed by either an
ordinal or a cardinal
;     (FIFTY, SIXTY, etc.)
; Z = Number suffix (TH, ND)
; H = Mile
; J = RURAL, STAR
; I = Comppany suffix (INC., AGENCY)
; A = Abbreviations to expand
; V = State names or abbreviations
; P = used internally
;
```

us_addrc.cls - Notepad

File Edit Format View Help

```
CNTRL           CENTRAL       A
COL             COLONEL       A
EXEC            EXECUTIVE     A
FX              FOX           A
GOV             GOVERNOR      A
GT              GREAT         A
HGHDS           HIGHLANDS     A
HLL             HILL          A
HLLS            HILLS         A
JFK             "JOHN F KENNEDY"      A
RFK             "ROBERT F KENNEDY"    A
MLK             "MARTIN LUTHER KING"    A
KENNEDY         KENNEDY       A
MKT             MARKET        A
MTNG            MEETING       A
MEM             MEMORIAL      A
MEMRL           MEMORIAL      A
MIL             MILITARY      A
MILIT           MILITARY      A
MLTARY          MILITARY      A
MLTRY           MILITARY      A
MNT             MOUNT         A
MT              MOUNT         A
MNTAIN          MOUNTAIN      A
MNTN            MOUNTAIN      A
MTN             MOUNTAIN      A
MNTNS           MOUNTAINS     A
PLEAS           PLEASANT      A
PLSNT           PLEASANT      A
RESERV          RESERVATION   A
RESERVATION     RESERVATION   A            800.0
SN              SAN           A
TOWNHSE         TOWNHOUSE     A
TWNHOUSE        TOWNHOUSE     A
```

Pattern file (.pat)

- The pattern file defines pattern rules and actions used to standardize addresses. This file is critical to the standardization process.
- The pattern file recognizes certain address format patterns and specifies how the address elemtns are assigned to different fields.
- Pattern files are encrypted, so geocoding tools such as the Standardizer Editor or encodepat.exe are used to edit and encrypt the pattern file. These tools are available as part of the ESRI Geocoding Developer Toolkit from ESRI Developer Network

Pattern file (.pat)

- **Example: 123 Terrace West Drive poses problems for the standard us_addr address locator. We can add the pattern Terrace West to the pattern file.**
- **The pattern is formatted with a series of operands separated by vertical bars `^ | D | ? | T | $`**
- **Typical operands are:**

  ^ - numeric value

  D - direction (from classification file)

  ? – unknown

  T – street type (from classification file)

  $ - match up to end of field.

  Example use with ^ | $: `TX 78232 returns 78232`

Pattern file (.pat)

- **Example pattern file before encoding:**

COPY – Copy value

COPY_A – Copy standardized

COPY_S – Copy w/ spaces

CONCAT – Concatenate

EXIT – Exit pattern

special_cases.xat - Notepad

File Edit Format View Help

```
; special cases.xat
; example: 123 Terrace West Drive
; "Terrace West" is the street name
^ | T="Terrace" | D="West" | $
COPY [1] {HN}
COPY [2] temp
CONCAT " " temp
CANCAT [3] temp
COPY temp {SN}
EXIT


^ | & | S | T ; ex: 123 23rd St. Ct.
COPY [1] {HN}
COPY 2 temp
CONCAT temp
CONCAT_A [3] temp
COPY temp {SN}
COPY_A [4] {ST}
EXIT

^ | D | ? | T ; ex: 123 North Main
Avenue
COPY [1] {HN}
COPY_A [2] {PD}
COPY_S [3] {SH}
COPY_A [4] {ST}
EXIT
```

Pattern file (.pat)

- **Handling Intersections:**

\& - "\" + intersection delimiter

**** - universal class, used to reset tokens**

RETYPE – clear token value

COPY_A – Copy standardized

COPY_S – Copy w/ spaces

EXIT – Exit pattern

intersections.xat - Notepad

File Edit Format View Help

```
; intersections
; example: N Main Ave E Clark Blvd

D | ? | T | \&
COPY_A [1] {P1}
COPY_S [2] {S1}
COPY_A [3] {T1}
RETYPE [1] 0
RETYPE [2] 0
RETYPE [3] 0

; use the following pattern to clear out
; everything before the second part
; (E Clark Ave)

** | \&
RETYPE [1] 0

D | ? | T
COPY_A [1] {P2}
COPY_S [2] {S2}
COPY_A [3] {T2}
EXIT
```

Where to learn more

- **ArcGIS Desktop Help**
  Contents > Geocoding and Address Management > Building an address locator
- **ESRI Geocoding Rule Base Developer Guide**
  http://edn.esri.com/index.cfm?fa=downloads.detail&downloadId=22
- **ESRI Virtual Campus Training**
  - **Geocoding with ArcGIS Desktop**
- **ArcGIS Online Resource Center**
  - **North American Address Locator Task**
  http://resources.esri.com/arcgisonlineservices/index.cfm?fa=content
- **Sample ArcScript for creating address locators**
  http://arcscripts.esri.com/details.asp?dbid=16186

Questions?