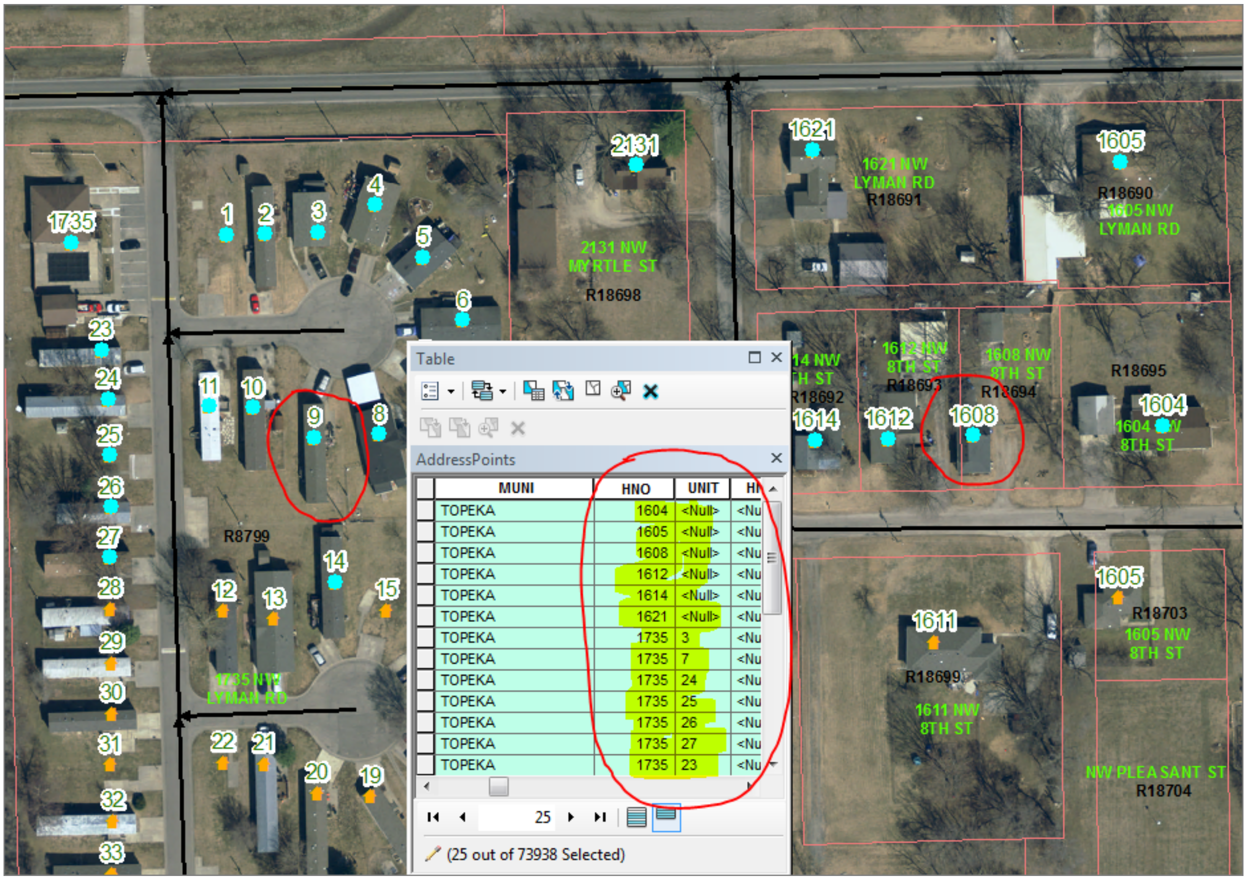
KANSAS NG911 Data Labeling
AddressPoints

In many applications, it is appropriate to only label the house number of an address on the map. This can easily be done using the [HNO] field of the AddressPoints data. But if there are units that need to be labeled instead of house numbers if they exist, it gets a little more complicated. To achieve this labeling effect (see below), a more advanced labeling script it needed.

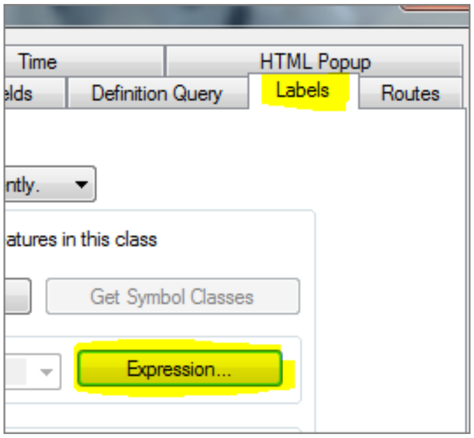


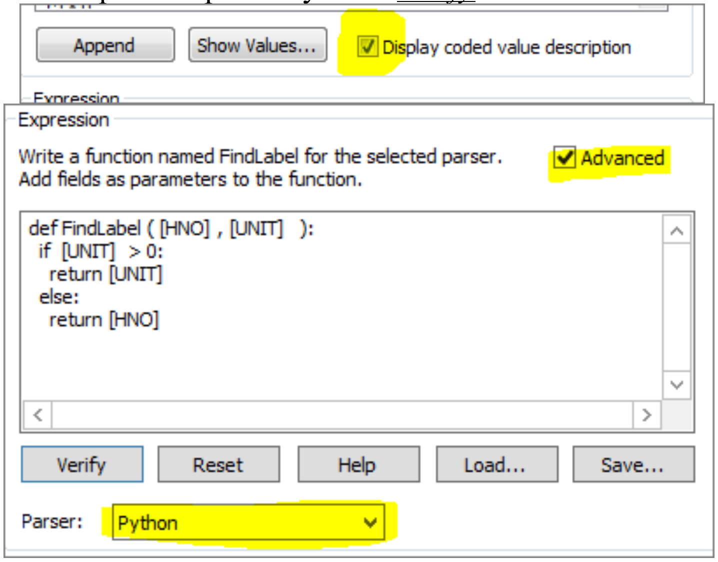
MUNI	HNO	UNIT	HI
TOPEKA	1604	<Null>	<Nu
TOPEKA	1605	<Null>	<Nu
TOPEKA	1608	<Null>	<Nu
TOPEKA	1612	<Null>	<Nu
TOPEKA	1614	<Null>	<Nu
TOPEKA	1621	<Null>	<Nu
TOPEKA	1735	3	<Nu
TOPEKA	1735	7	<Nu
TOPEKA	1735	24	<Nu
TOPEKA	1735	25	<Nu
TOPEKA	1735	26	<Nu
TOPEKA	1735	27	<Nu
TOPEKA	1735	23	<Nu

25

(25 out of 73938 Selected)

Included in the folder is a label expression file (AddressPoint_Number_Label.lxp).
To use this file, open the labeling properties of AddressPoints and click ***Expression***. Set settings like shown below, then click ***Load…***. Navigate to the .lxp file. Optionally click *Verify*.





☑ Display coded value description

Write a function named FindLabel for the selected parser.
Add fields as parameters to the function.

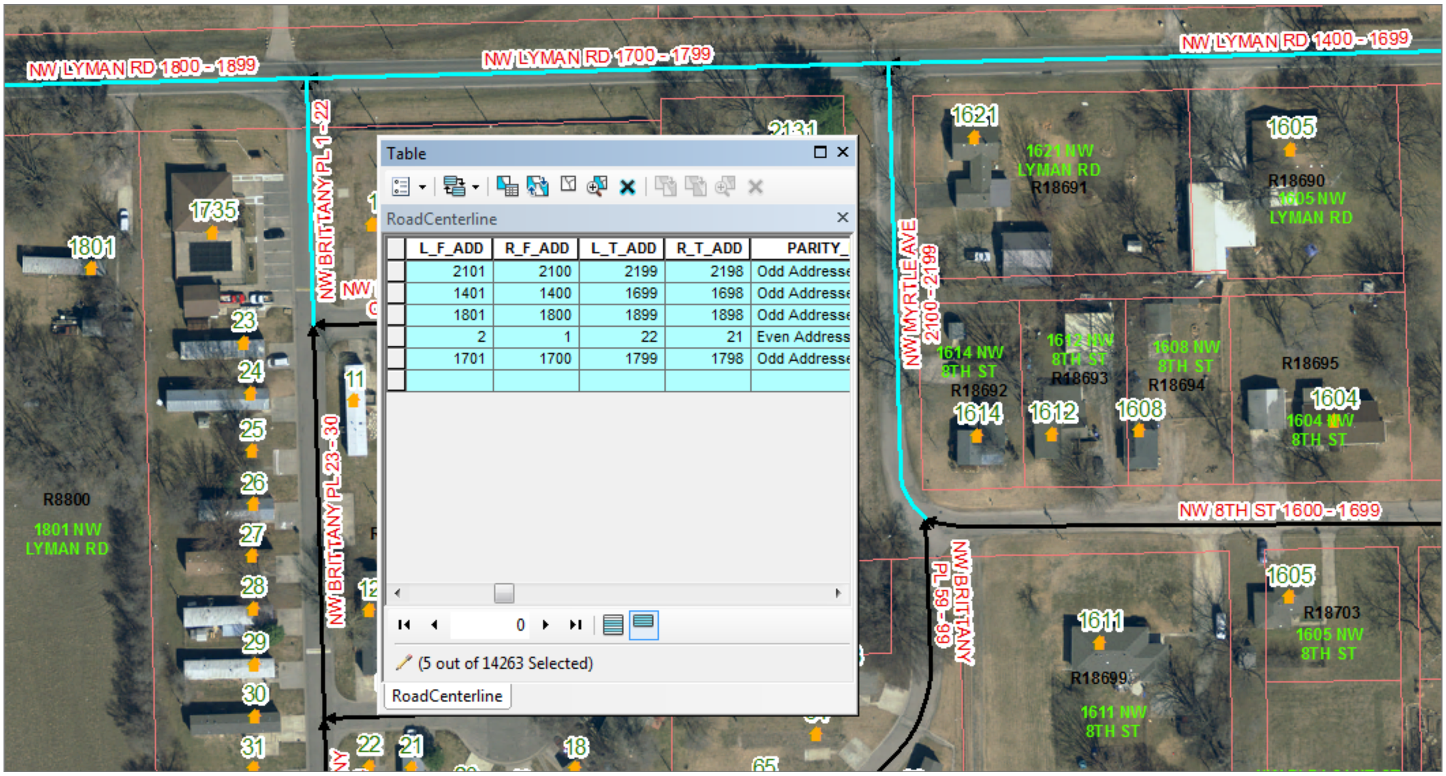
☑ Advanced

```
def FindLabel ( [HNO] , [UNIT]  ):
  if [UNIT]  > 0:
    return [UNIT]
  else:
    return [HNO]
```

Parser: Python

KANSAS NG911 Data Labeling
RoadCenterline

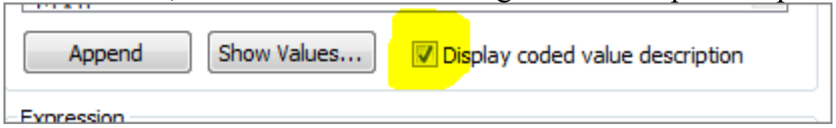
While editing AddressPoints data, it is helpful to see the map ranges of the street segment to verify that the address of the point fits in the map range and will geocode. The label expression below automatically calculates the lowest number and highest number in the ranges. The result looks like below.



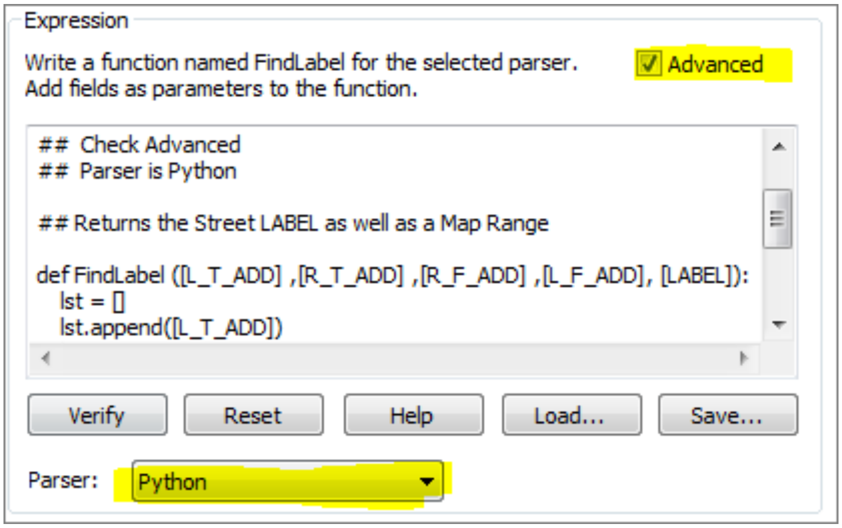
L_F_ADD	R_F_ADD	L_T_ADD	R_T_ADD	PARITY_
2101	2100	2199	2198	Odd Addresse
1401	1400	1699	1698	Odd Addresse
1801	1800	1899	1898	Odd Addresse
2	1	22	21	Even Address
1701	1700	1799	1798	Odd Addresse

(5 out of 14263 Selected)

Included in the folder is a label expression file (RoadCenterline_Name_and_Map_Range.lxp).
To use this file, open the labeling properties of RoadCenterline feature and click ***Expression***. Set settings like shown below, then click ***Load…***. Navigate to the .lxp file. Optionally click *Verify*.



☑ Display coded value description



Write a function named FindLabel for the selected parser.
Add fields as parameters to the function.

☑ Advanced

```
## Check Advanced
## Parser is Python

## Returns the Street LABEL as well as a Map Range

def FindLabel ([L_T_ADD] ,[R_T_ADD] ,[R_F_ADD] ,[L_F_ADD], [LABEL]):
  lst = []
  lst.append([L_T_ADD])
```

Parser: Python