



# Working with Cartographic Representations

Edie Punt

Chait Gaddam

Technical Workshop

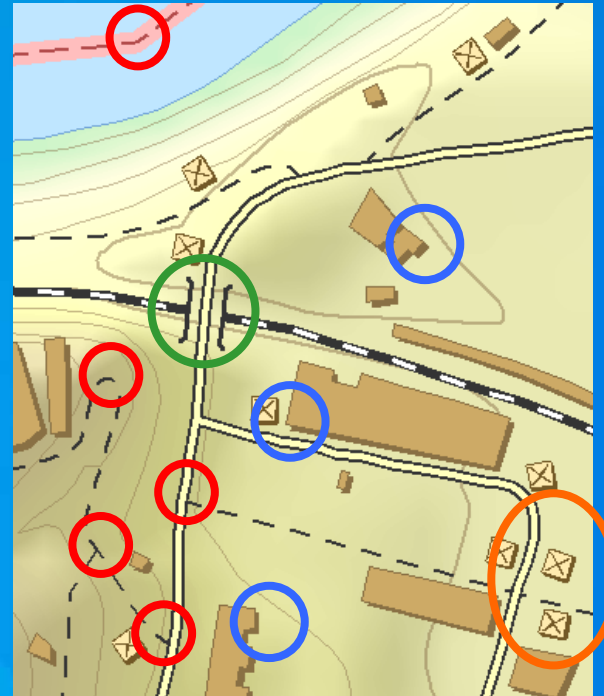
# What are cartographic representations?

- A way to symbolize features to solve cartographic challenges
- A storage model that stores symbols with data
- Part of a feature class, managed through a layer



# What can representations do?

Representations can draw features cartographically



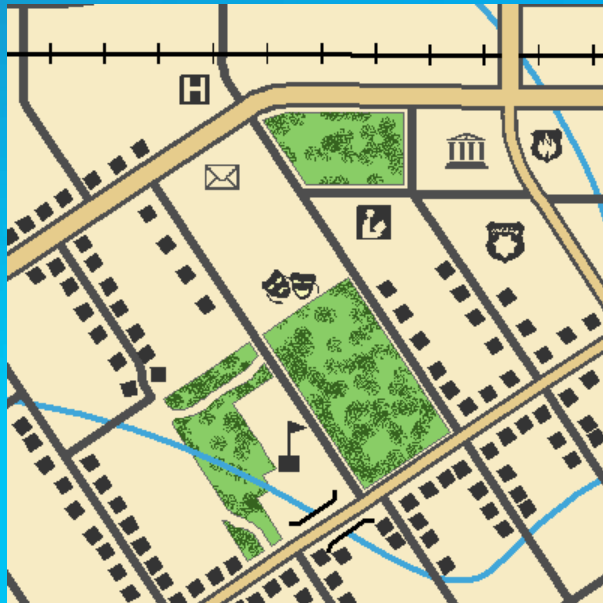
# What can representations do?

**Representations can produce dynamic geometry which may differ from spatial geometry**



# What can representations do?

Feature classes can have more than one representation to draw the same data in different ways





# What can representations do?

Representations can be data-driven to tailor symbols to feature attribution



Streets drawn  
with one symbol

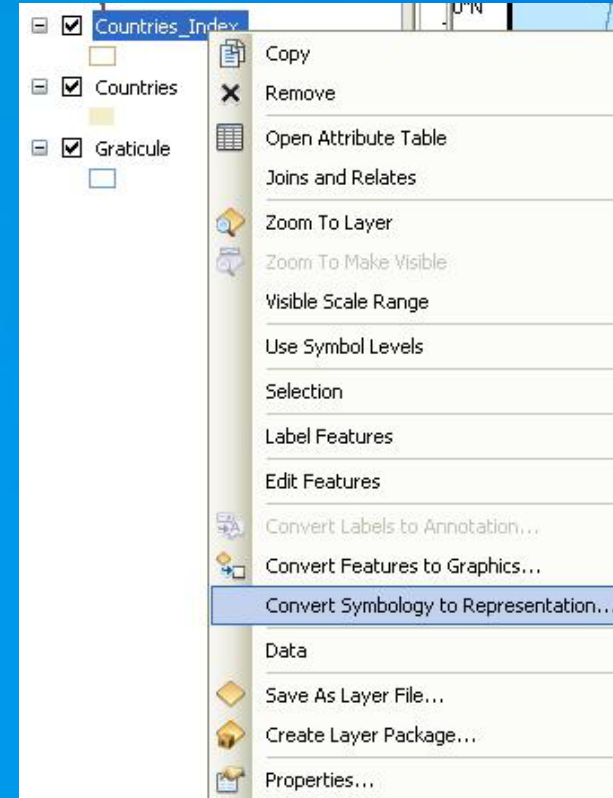


Casing and fill width  
from attributes

ID	Type	Casing	Fill	RuleID	Override
1	Street	1	0	StreetSym	<null>
2	Major Road	3	2	StreetSym	<null>
3	Minor Road	2	1	StreetSym	<null>
4	Minor Road	2	1	StreetSym	<null>
5	Street	1	0	StreetSym	<null>

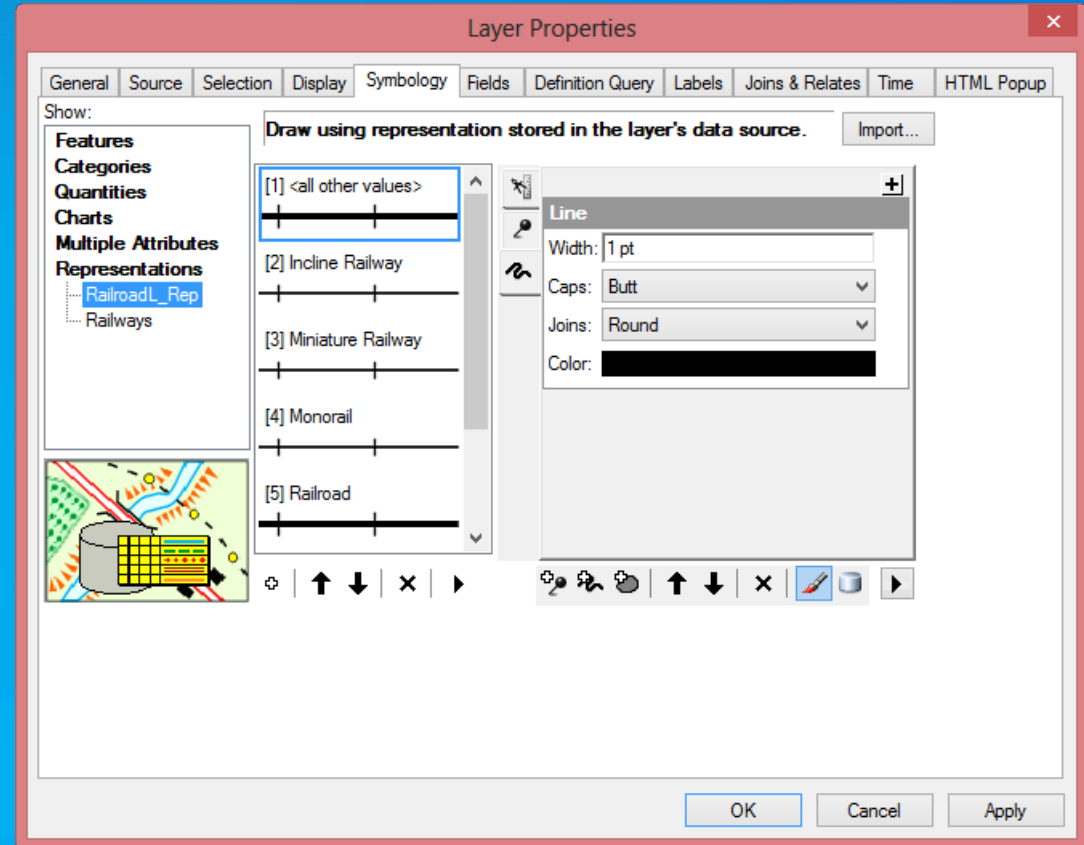
# How do I get started?

**Convert a symbolized layer to a representation on the source feature class from the Table of Contents**



# How do I draw representations?

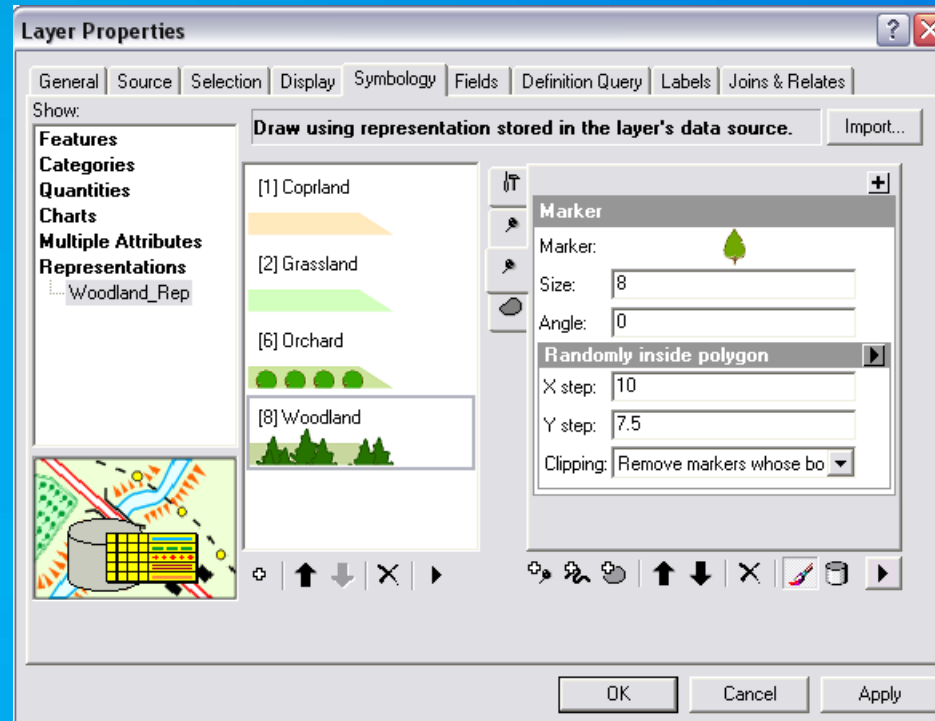
You can symbolize a layer by a representation if one (or more) exist on the source





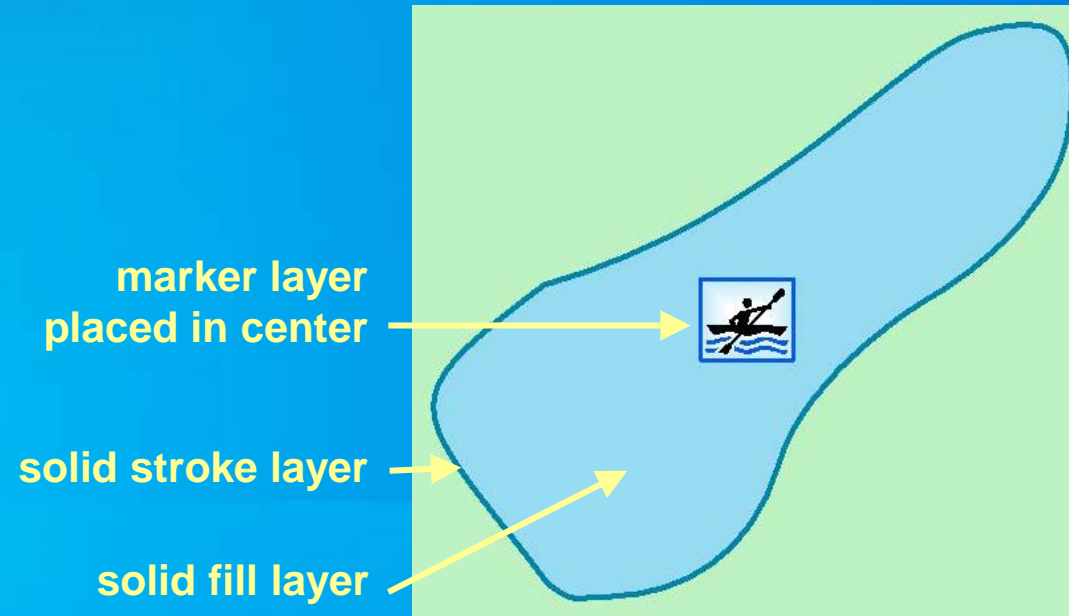
# What are representation rules?

- Like symbols: a set of drawing instructions
- Consist of:
  - symbol layers
  - geometric effects



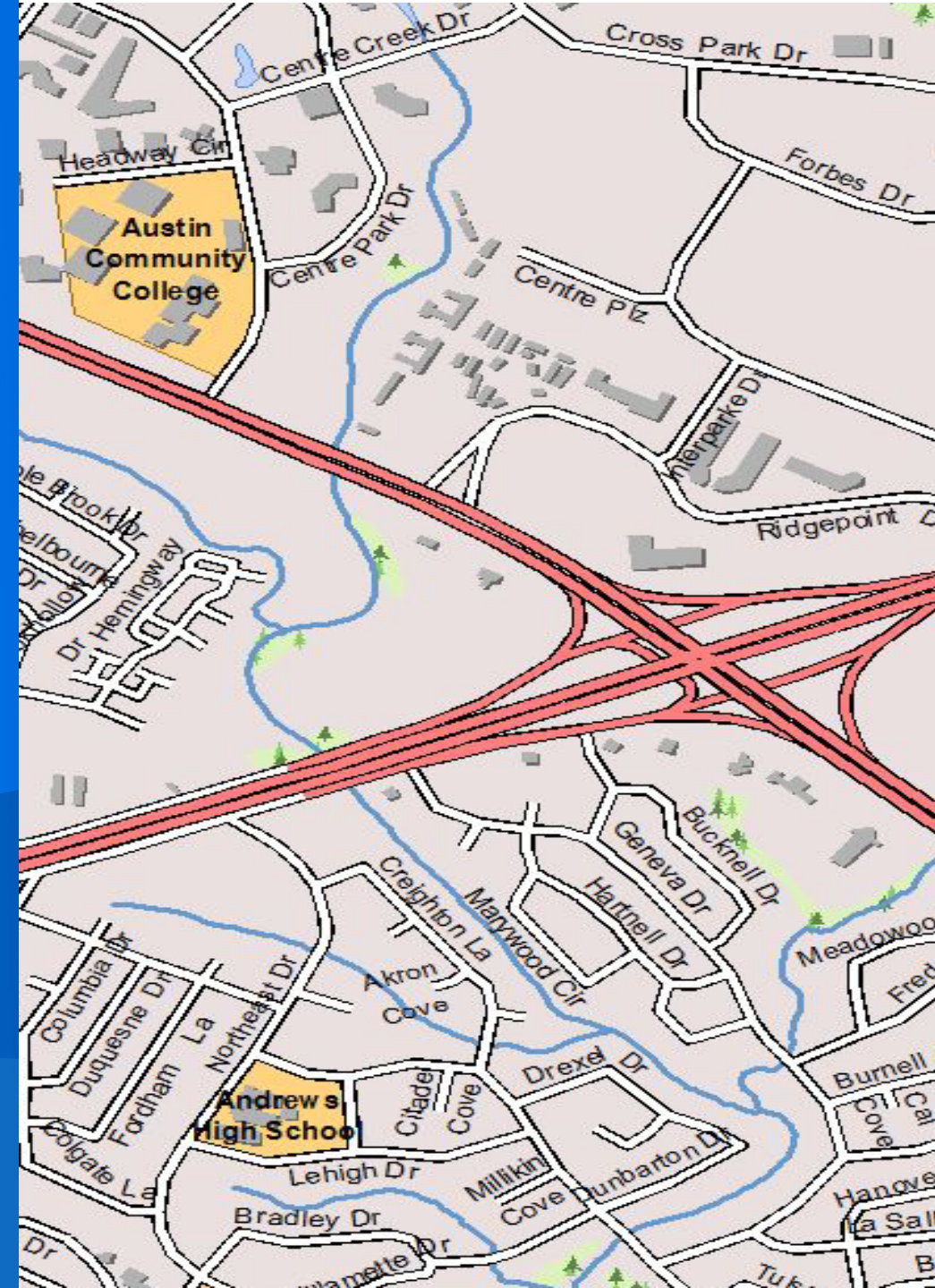
# Sample representation rule

- **Lake representation rule:**
  - **marker layer**
  - **stroke layer**
  - **fill layer**



Demo

# Working with representations



# Geometric Effects and Marker Placement Styles

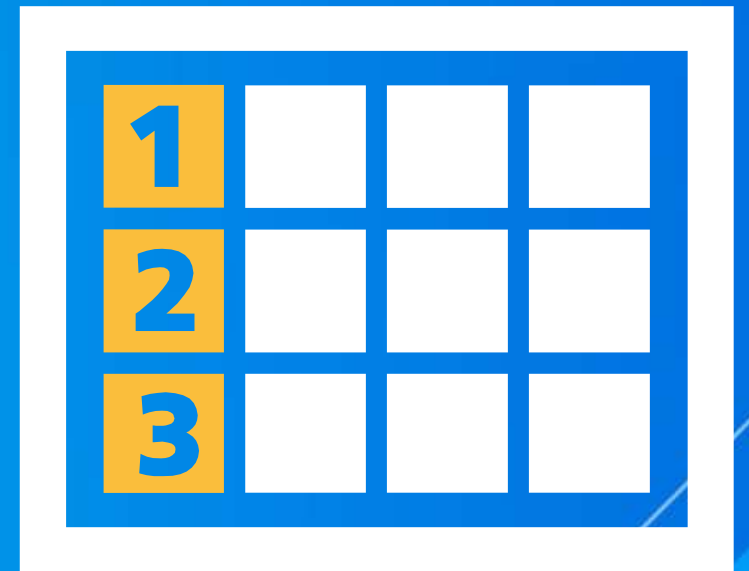




# Representation rules

Representation rules contain:

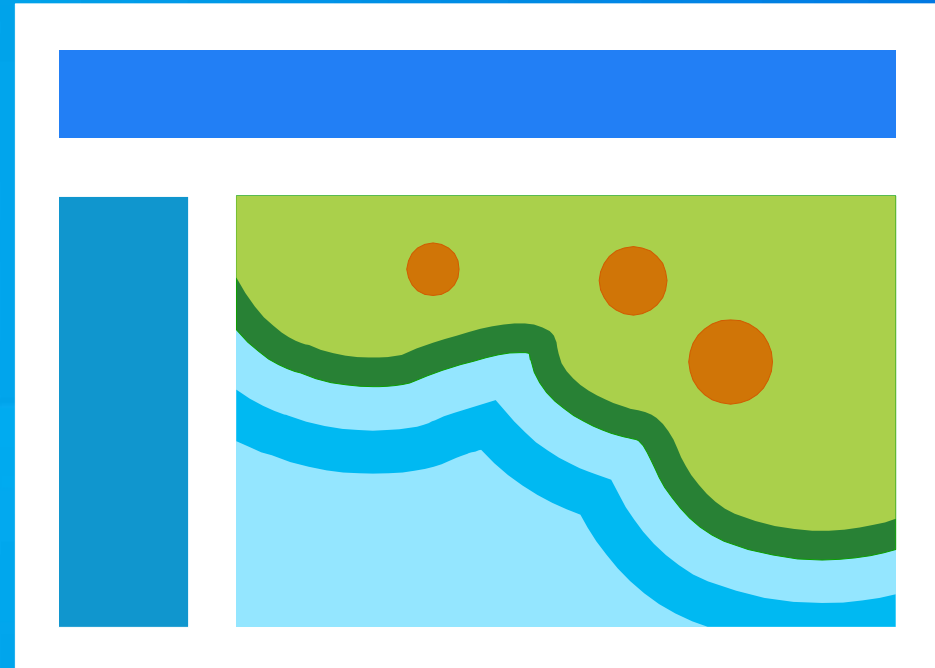
- symbol layers- *drawing instructions*
- geometric effects- *dynamic geometry changes*
- marker placements- *marker position instructions*





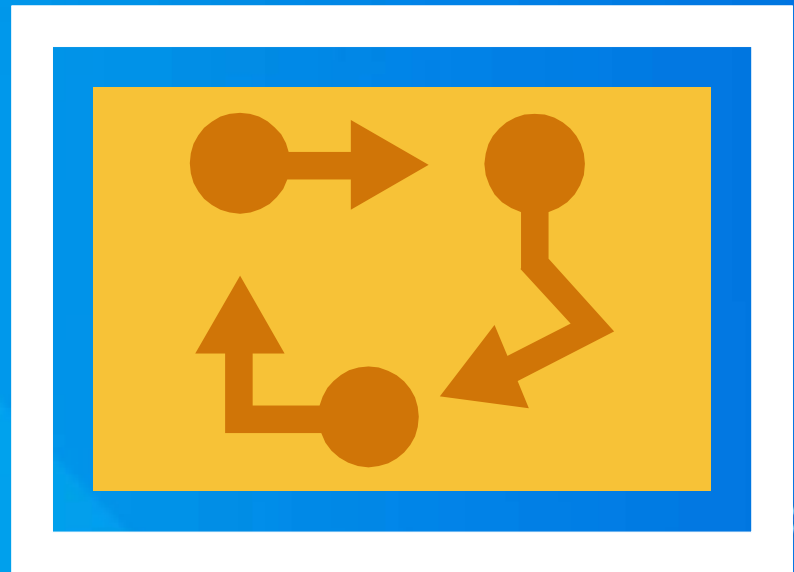
# What are geometric effects?

- **Dynamically change geometry before symbol applied**
  - Affect all symbol layers, or just one
  - Can convert geometry type
- **Can work sequentially**
- **Extensible: write your own**



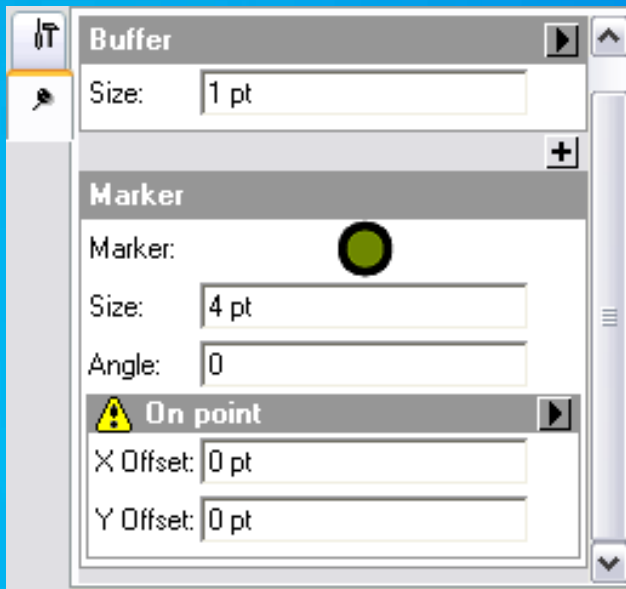
# Chaining geometric effects

- The dynamic output of one geometric effect becomes the input geometry for the rule
- Geometric effects can be chained together
- Geometric effects can be applied to the whole representation rule symbol, or just one symbol layer

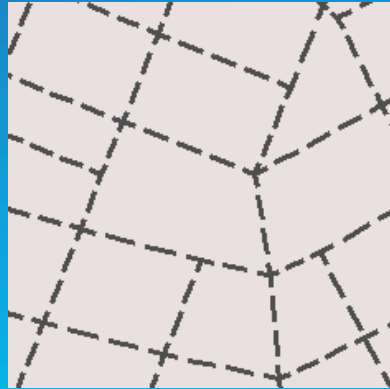


# Representation geometry logic

- The geometry of the last effect in the chain must match the geometry of the symbol layer
  - Else a geometry logic warning appears



# Geometric effects examples

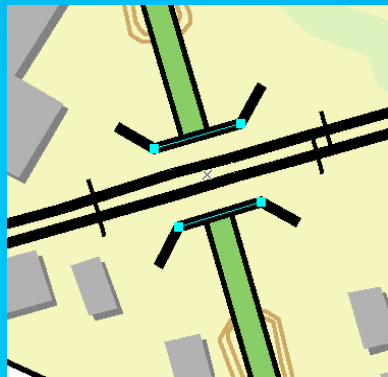


Dashes

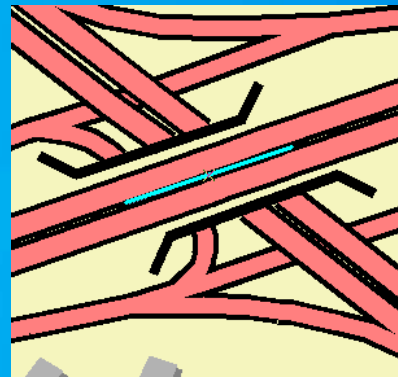


Move

Extension



Offset Tangent

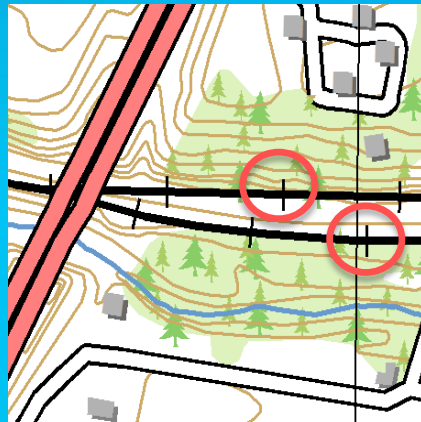


Suppress



# What are marker placements?

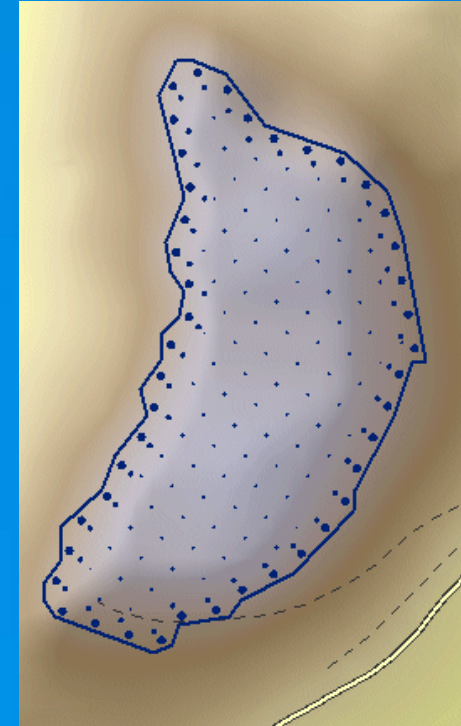
- **Position representation markers**
  - Along lines and polygon outlines
  - Within polygons
  - In relation to points
- **Extensible: write your own**





## Marker placement example

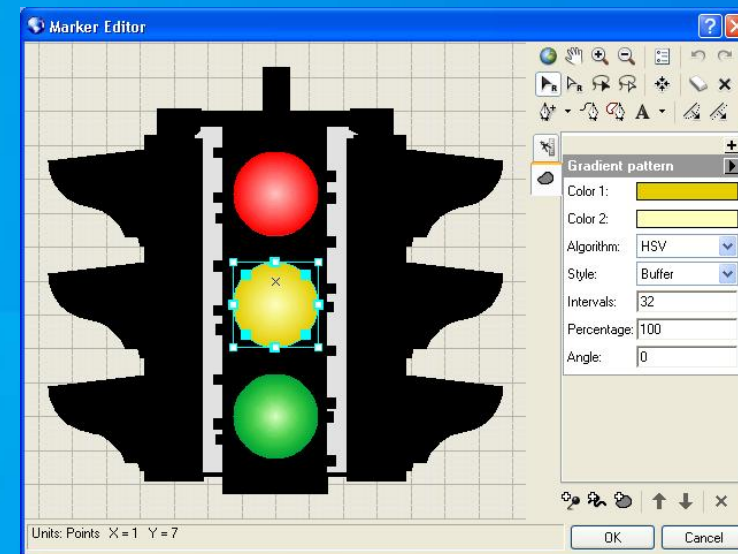
- **stroke layer-** *solid outline*
- **marker layer-** *large dots*
  - placement = along outline
  - offset = -3 pt
- **marker layer-** *medium dots*
  - placement = along outline
  - offset = -7 pt
- **marker layer-** *small dots*
  - placement = inside polygon
  - offset = -10 pt



# Creating representation markers

## Marker Editor

- Import font glyphs, edit vertices
- Draw polygons and lines
- Align, order, group, rotate
- Resize, erase, warp



Demo

# Geometric effects and marker placements



# Overriding representations



# What are representation overrides?

## Exceptions to the rule

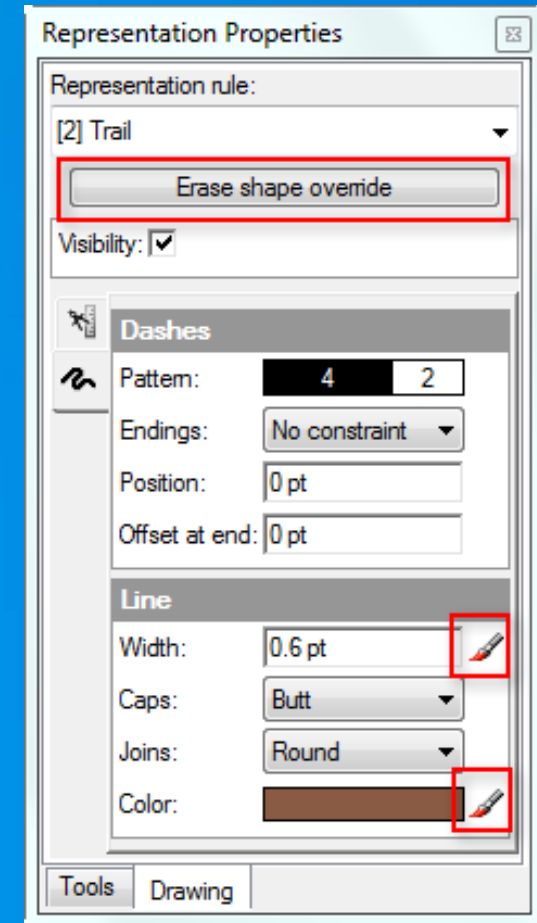
- Customize individual features
- Made while editing

## Property overrides

- Change any property of the rule

## Geometry overrides

- Store alternate geometry





# Property override



**Create an exception to the rule**

# Geometry overrides

Must enable representation to handle geometry overrides

New Representation

Name: RoadL\_Rep2

Rule ID Field: RuleID\_2

Override Field: Override\_2

Behavior When Representation Geometry Is Edited

- Store change to geometry as representation override
- Change the geometry of the supporting feature

Representation Rules

Import rules from a layer (.lyr) file:

No layer defined Select...

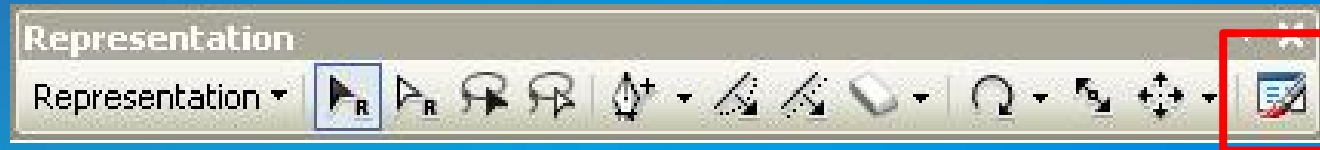
Assign rules to features to match the layer

< Back Next > Cancel

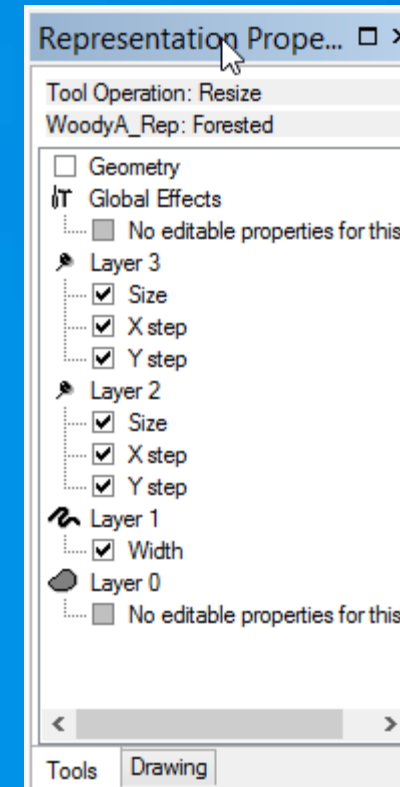
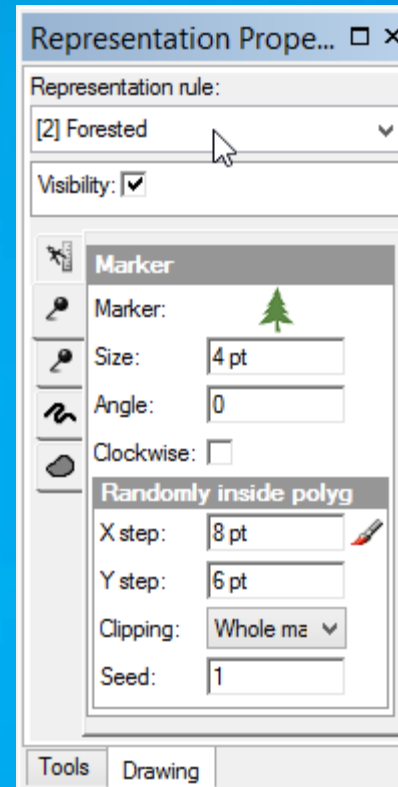
# Creating overrides

- Interactively with the representation toolbar
- Explicitly change rule property
- Use field in the attribute table to define override
- Cartographic Refinement geoprocessing toolset

# Interactive overrides



Use the Representation Properties window to manage feature overrides



# Overrides in explicit fields

## Link symbol layer properties to feature attributes

- Size, width, angle, visibility, etc.

## Draw symbols based on current data

## Any rule property can use attribute values

- Must use valid field data type

## Reduce duplication of rules

The screenshot shows a 'Marker' property panel on the left and a data table on the right. The panel has fields for 'Marker' (a square icon), 'Size' (6), 'Angle' (0), 'On point' (a play button icon), 'X Offset' (0), and 'Y Offset' (0). A yellow box highlights the 'Angle' field in the panel, and a yellow arrow points from it to the 'Angle' column in the table. The table has a header row with 'Angle' and five data rows with values 35, 55, 22, 45, and 0.

Angle
35
55
22
45
0



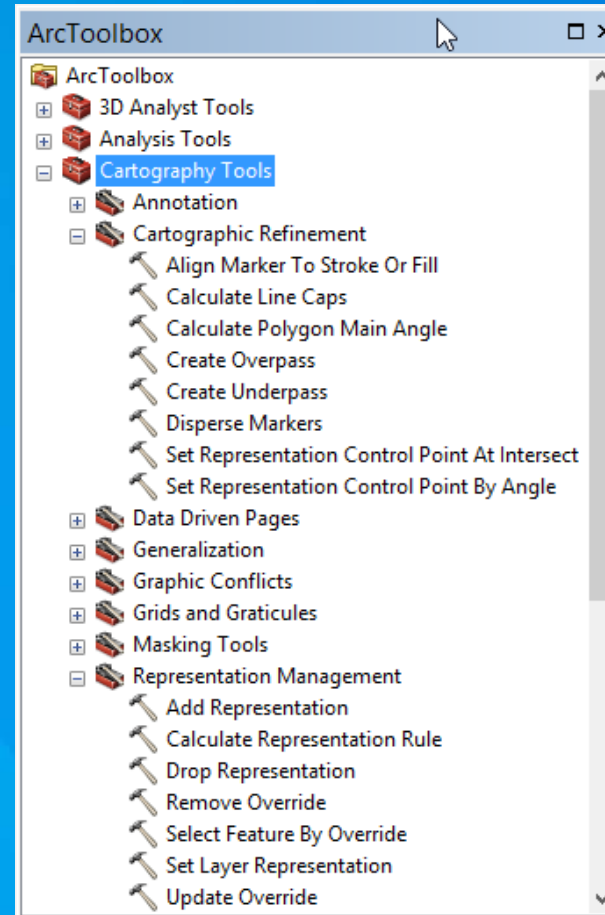
# Using geoprocessing

## Cartographic Refinement

- feature-level

## Representation Management

- representation-level



Demo

# Overriding representations





Understanding our world.