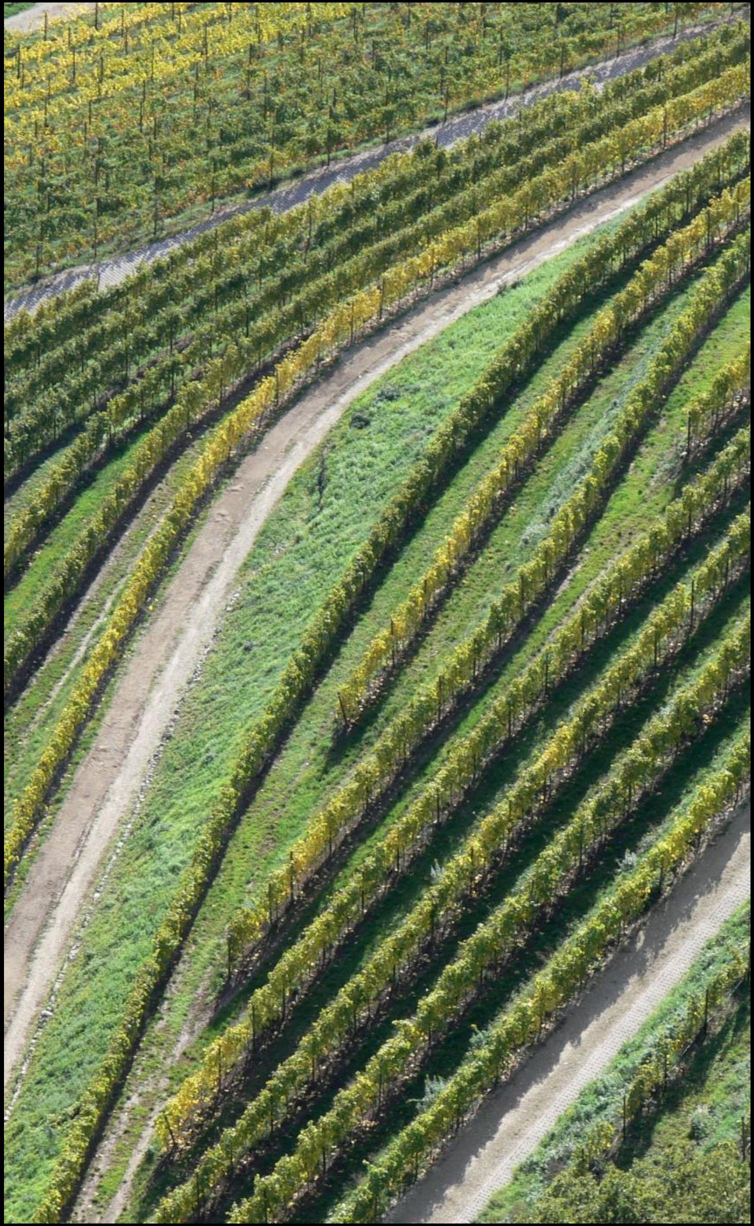
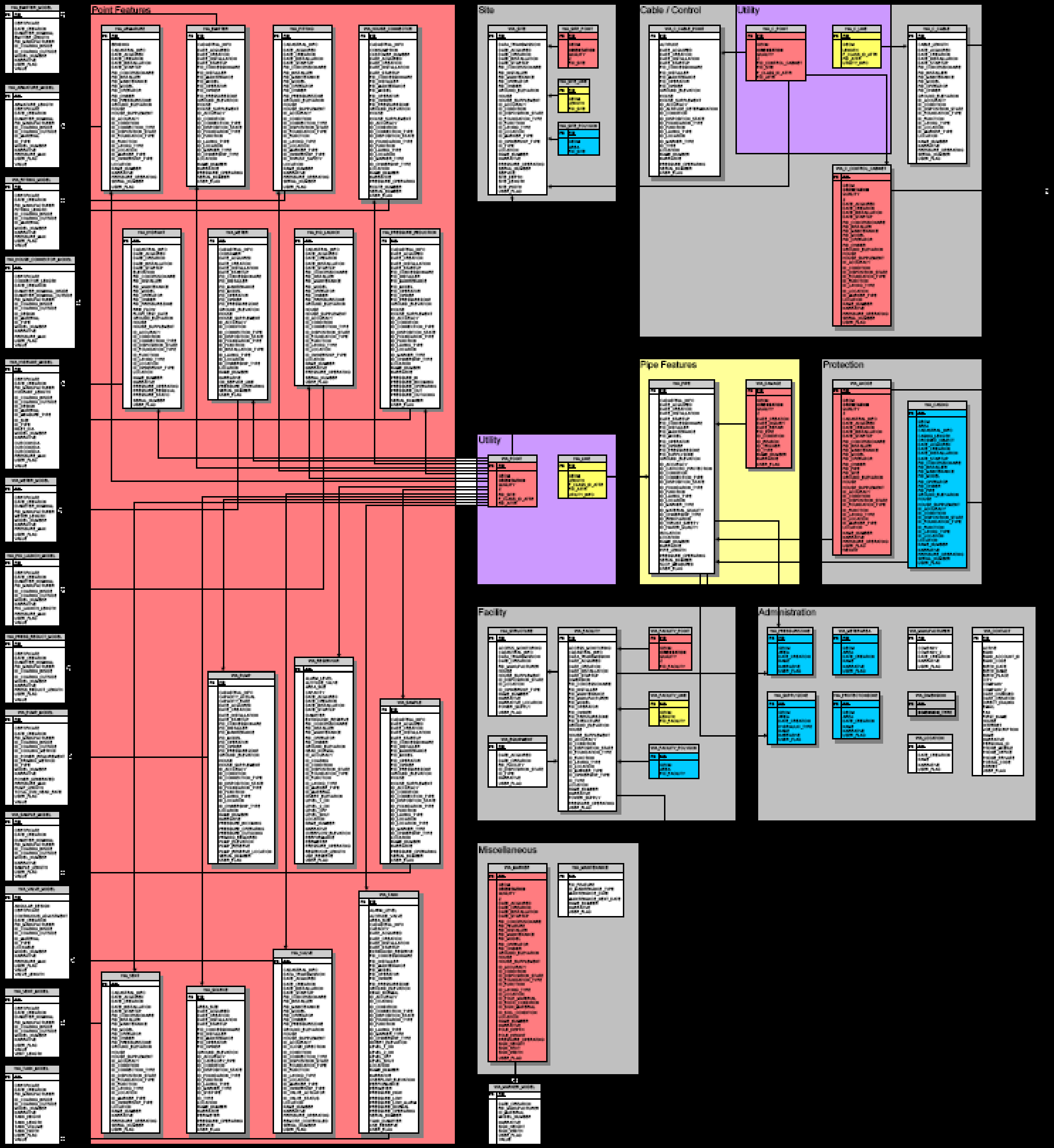
Archive – Technical Training

Autodesk Topobase Administrator

Disclaimers

- This course was created for Autodesk Topobase 2009. The contents of this course are not intended for other versions of Autodesk Topobase.
- We may make statements regarding planned or future development efforts for our existing or new products and services. These statements are not intended to be a promise or guarantee of future delivery of products, services or features but merely reflect our current plans, which may change. Purchasing decisions should not be made based upon reliance on these statements.
- The Company assumes no obligation to update these forward-looking statements to reflect events that occur or circumstances that exist or change after the date on which they were made.




Point Features
Site
Cable / Control
Utility
Utility
Pipe Features
Protection
Facility
Administration
Miscellaneous

Chapter Overview

- Chapter provides information on workflows and feature rules in Autodesk Topobase 2009.
  - Workflows and feature rules allow you to enforce business rules and maintain data consistency.

Workflows and Feature Rules

Chapter Objectives

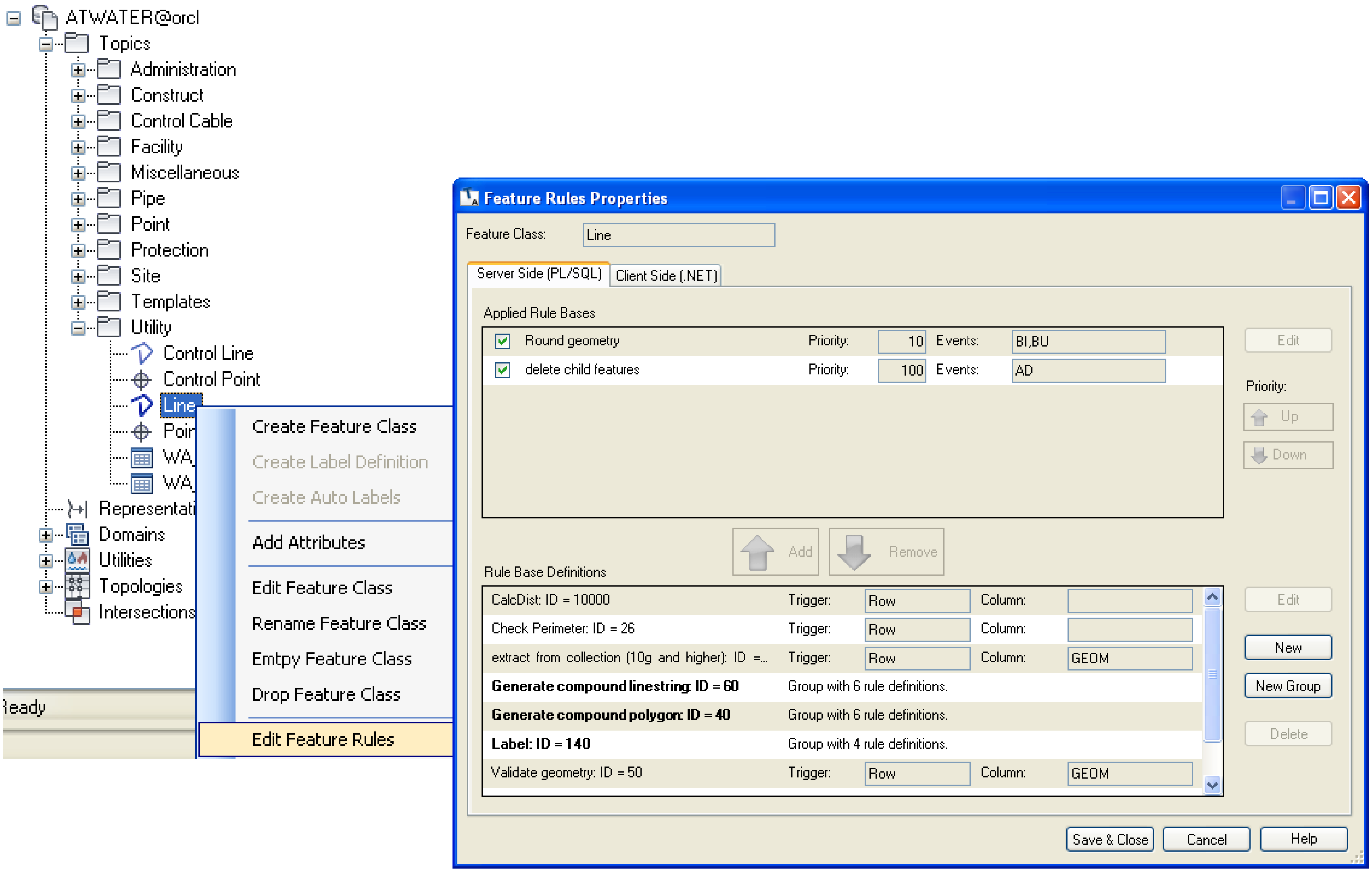
- By the end of this chapter, you will be able to:
  - Understand the purpose of feature rules
  - Create a server-side feature rule
  - Understand the value of using workflows
  - Start a workflow and follow workflow guidelines

7.2 What Are Feature Rules?

- Feature rules are methods automatically executed when a change is made to the database.
  - They enforce *business rules*, which control the operation of an enterprise to ensure the quality of the data.

- Autodesk Topobase 2009 supports both server-side and client-side feature rules.

7.2.1 Browsing Feature Rules



Feature Class: Line

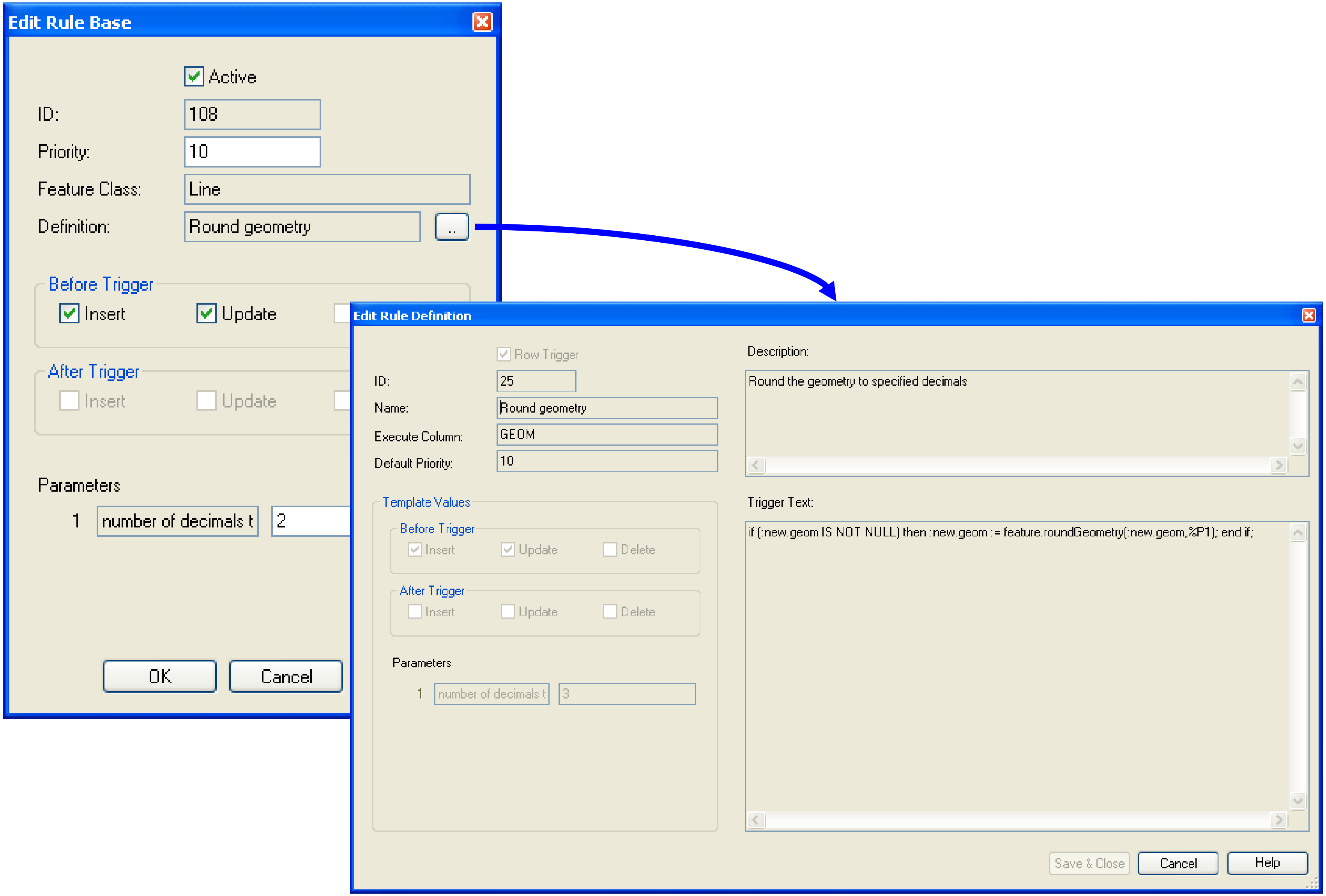
Applied Rule Bases

☑	Round geometry	Priority:	10	Events:	BI,BU
☑	delete child features	Priority:	100	Events:	AD

Rule Base Definitions

CalcDist: ID = 10000	Trigger:	Row	Column:	
Check Perimeter: ID = 26	Trigger:	Row	Column:	
extract from collection (10g and higher): ID =...	Trigger:	Row	Column:	GEOM
Generate compound linestring: ID = 60	Group with 6 rule definitions.			
Generate compound polygon: ID = 40	Group with 6 rule definitions.			
Label: ID = 140	Group with 4 rule definitions.			
Validate geometry: ID = 50	Trigger:	Row	Column:	GEOM

7.2.1 Browsing Feature Rules +



Edit Rule Base

☑ Active

ID: 108

Priority: 10

Feature Class: Line

Definition: Round geometry

Before Trigger

☑ Insert ☑ Update □

After Trigger

□ Insert □ Update □

Parameters

1 number of decimals t 2

Edit Rule Definition

☑ Row Trigger

ID: 25

Name: Round geometry

Execute Column: GEOM

Default Priority: 10

Template Values

Before Trigger

☑ Insert ☑ Update □ Delete

After Trigger

□ Insert □ Update □ Delete

Parameters

1 number of decimals t 3

Description:

Round the geometry to specified decimals

Trigger Text:

```
if (:new.geom IS NOT NULL) then :new.geom := feature.roundGeometry(:new.geom,%P1); end if;
```

7.2.2 Triggers

Trigger events

The triggers are executed or fired in response to changes in the data or database events; these are the *triggering events*.

Trigger event	**Description**
INSERT	It fires when a record is inserted in the table.
UPDATE	It fires when a change is made in any record in the database. UPDATE triggers can additionally specify an OF clause to restrict firing only when updating specific columns. (i.e. in the syntax when creating a trigger UPDATE OF <column1>, <column2>)
DELETE	It fires when a record is deleted from a table. Note that the trigger doesn’t fire with a TRUNCATE of the table.

7.2.2 Triggers +

Types of execution

- Triggers can be executed BEFORE, AFTER or INSTEAD OF the triggering event.
  - BEFORE triggers can change the data in the affected rows.
  - AFTER triggers are more suited for data validation, they cannot change the new data.
  - INSTEAD OF triggers replace the triggering event.

Execution levels

- There are two levels: statement-level or row-level.
  - Row-level triggers fire once per each row affected.
  - Statement-level triggers are the default; they fire once for each triggering statement.

7.2.2 Triggers ++

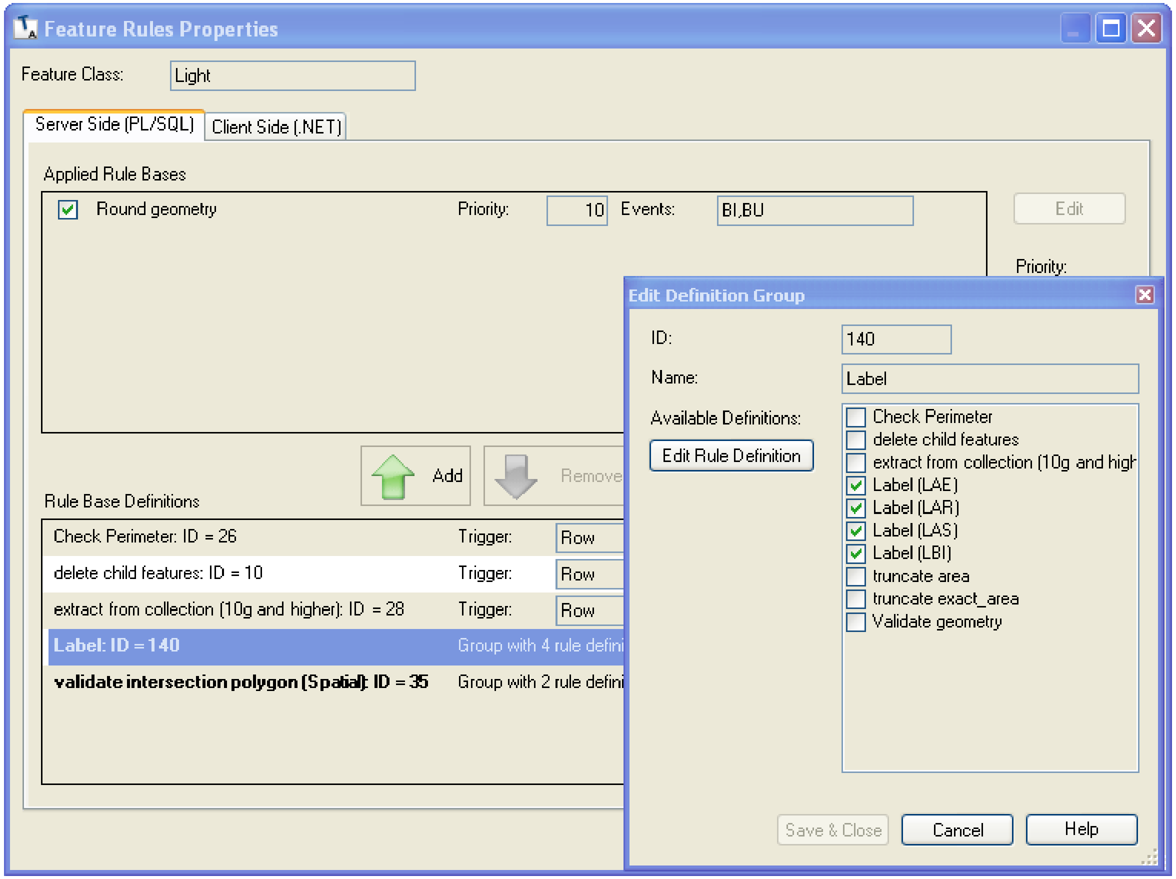
Syntax

- The general syntax to create a trigger on a DML event is:

```
CREATE [OR REPLACE] TRIGGER <trigger_name>
{BEFORE|AFTER|INSTEAD OF} <trigger_event>
ON <table_name>
[FOR EACH ROW] [WHEN <trigger_condition>]
<trigger_body>;
```

7.2.3 Groups of Feature Rules

- The feature rules can be grouped. It is possible to have a mixture of AFTER and BEFORE feature rules, and row and statement-level feature rules.



Feature Rules Properties

Feature Class: Light

Server Side (PL/SQL) | Client Side (.NET)

Applied Rule Bases

☑ Round geometry Priority: 10 Events: BI,BU

Rule Base Definitions

Check Perimeter: ID = 26	Trigger:	Row
delete child features: ID = 10	Trigger:	Row
extract from collection (10g and higher): ID = 28	Trigger:	Row
Label: ID = 140	Group with 4 rule defini	
validate intersection polygon (Spatial): ID = 35	Group with 2 rule defini	

Edit Definition Group

ID: 140

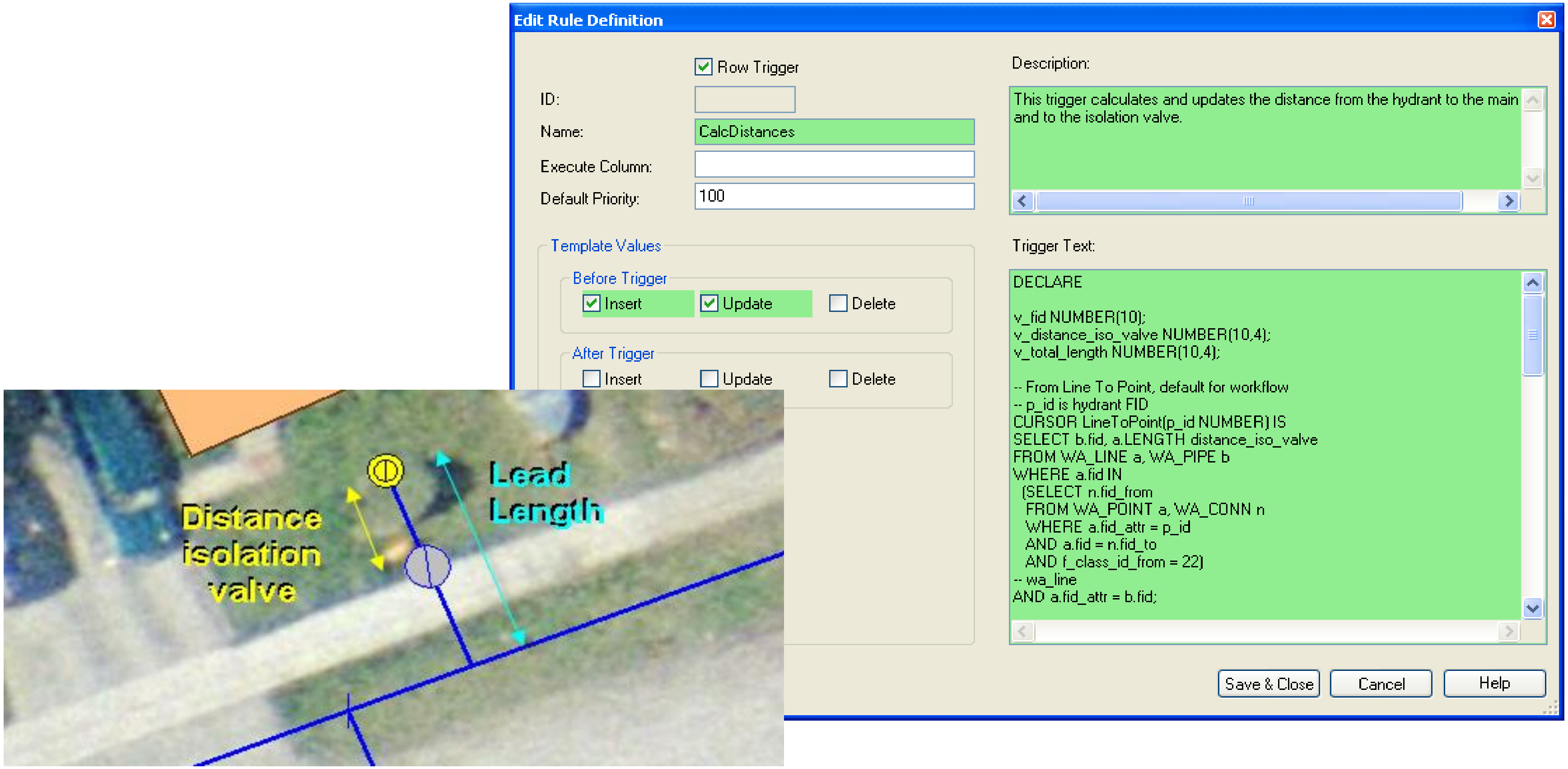
Name: Label

Available Definitions:

- □ Check Perimeter
- □ delete child features
- □ extract from collection (10g and high
- ☑ Label (LAE)
- ☑ Label (LAR)
- ☑ Label (LAS)
- ☑ Label (LBI)
- □ truncate area
- □ truncate exact_area
- □ Validate geometry

7.2.4 Server-Side Feature Rule Example

- You will create a new feature rule to fill out the LEAD_LENGTH, DISTANCE_ISO_VALVE attributes automatically whenever a hydrant is inserted or updated.



☑ Row Trigger

ID:

Name: CalcDistances

Execute Column:

Default Priority: 100

Before Trigger: ☑ Insert ☑ Update □ Delete

After Trigger: □ Insert □ Update □ Delete

Description:

This trigger calculates and updates the distance from the hydrant to the main and to the isolation valve.

Trigger Text:

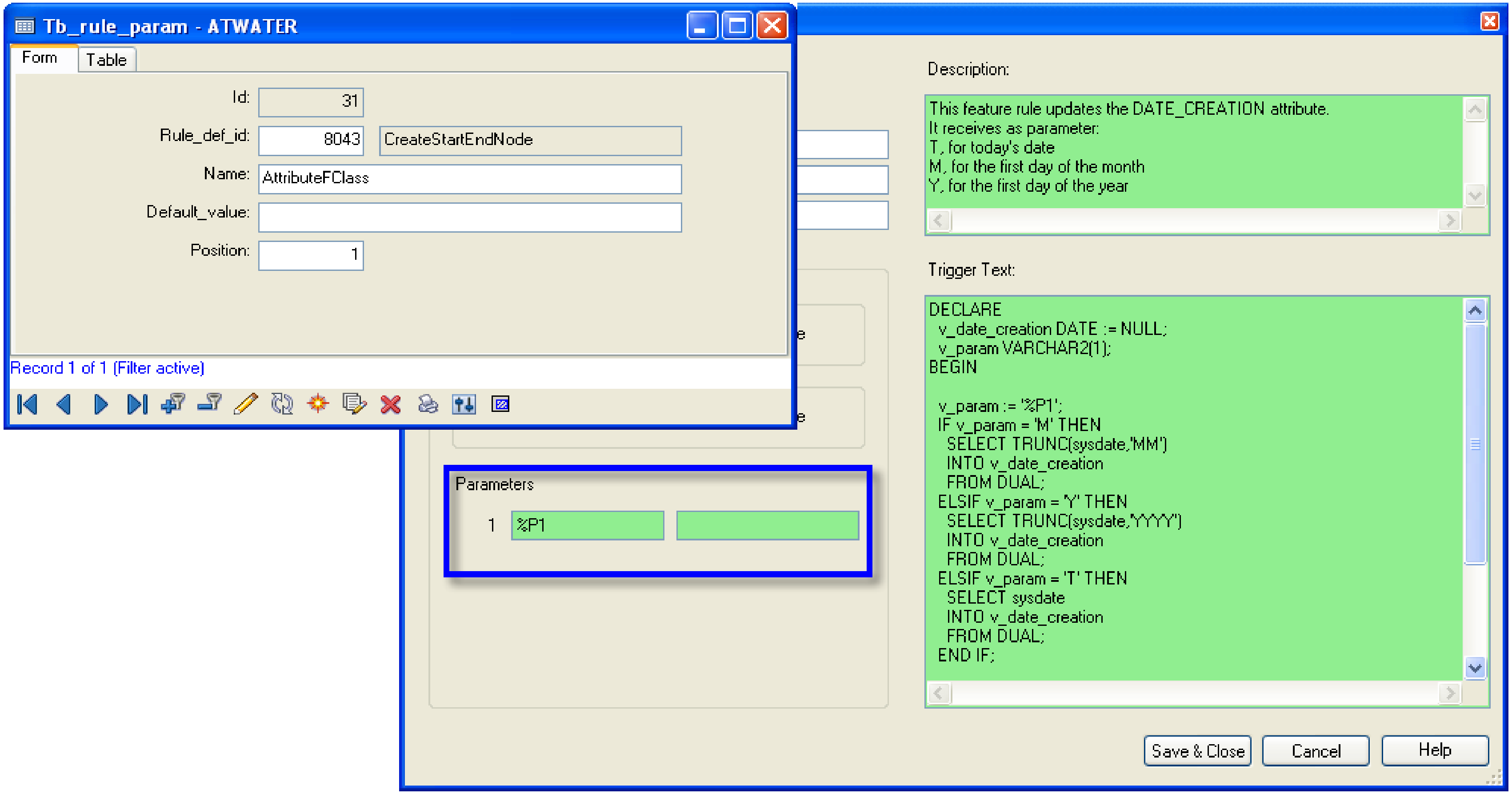
```
DECLARE

v_fid NUMBER(10);
v_distance_iso_valve NUMBER(10,4);
v_total_length NUMBER(10,4);

-- From Line To Point, default for workflow
-- p_id is hydrant FID
CURSOR LineToPoint(p_id NUMBER) IS
SELECT b.fid, a.LENGTH distance_iso_valve
FROM WA_LINE a, WA_PIPE b
WHERE a.fid IN
  (SELECT n.fid_from
   FROM WA_POINT a, WA_CONN n
   WHERE a.fid_attr = p_id
   AND a.fid = n.fid_to
   AND f_class_id_from = 22)
-- wa_line
AND a.fid_attr = b.fid;
```

7.2.5 Use of Parameters

- For any feature rule (client or server-side), you can add parameters.
  - The parameters of a feature rule are stored in the system table TB_RULE_PARAM.



Id:	31	
Rule_def_id:	8043	CreateStartEndNode
Name:	AttributeFClass	
Default_value:		
Position:	1	

Record 1 of 1 (Filter active)

Parameters

1 %P1

Description:

This feature rule updates the DATE_CREATION attribute.
It receives as parameter:
T, for today's date
M, for the first day of the month
Y, for the first day of the year

Trigger Text:

```
DECLARE
  v_date_creation DATE := NULL;
  v_param VARCHAR2(1);
BEGIN

  v_param := '%P1';
  IF v_param = 'M' THEN
    SELECT TRUNC(sysdate,'MM')
    INTO v_date_creation
    FROM DUAL;
  ELSIF v_param = 'Y' THEN
    SELECT TRUNC(sysdate,'YYYY')
    INTO v_date_creation
    FROM DUAL;
  ELSIF v_param = 'T' THEN
    SELECT sysdate
    INTO v_date_creation
    FROM DUAL;
  END IF;
```

7.2.6 Triggers Naming Conventions

- The triggers are named in this way:

  <table_name>_XXX

- Where XXX indicates the type of trigger:
  - The first character indicates if it is a before (B) or after (A) trigger
  - The second character indicates the type of DML operation: (I) Insert, (U) Update, (D) Delete
  - The third character indicates the execution level: (S) Statement, (R) Row

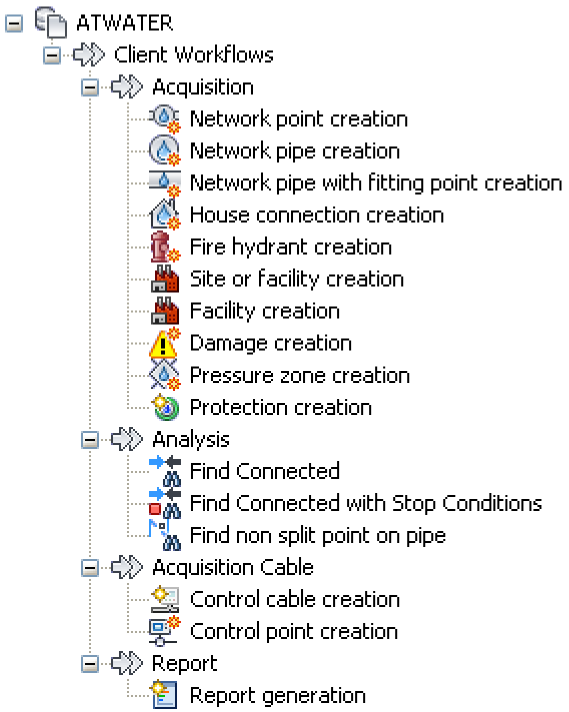
7.3 What Are Workflows?

- A workflow is an explicit prescription of how a process is to be structured and executed.
  - The value of a workflow is that it can increase the quality and repeatability of a task by controlling how the task is done.

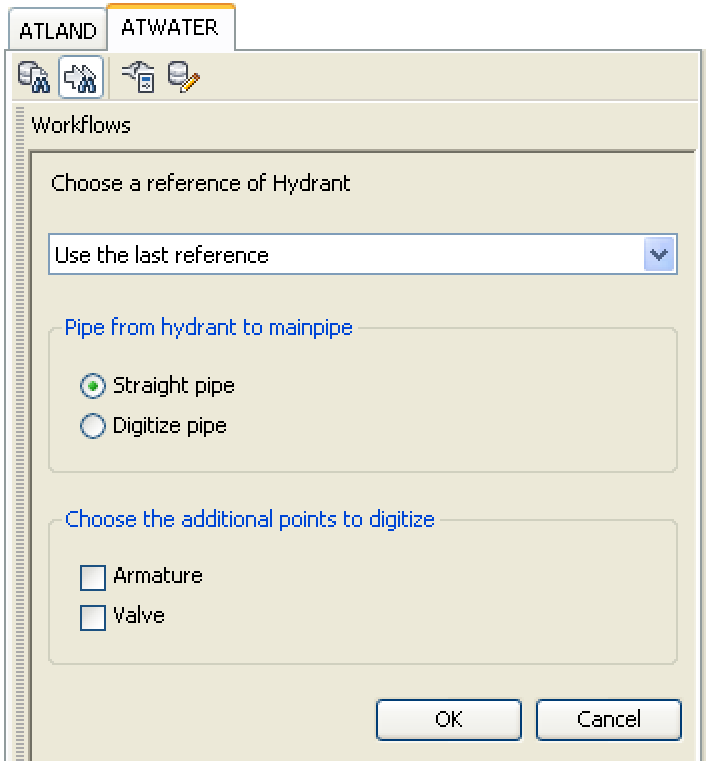
- Topobase provides workflows to guide users through tasks like:
  - Acquisition
  - Analysis
  - Maintenance
  - Reports

- You can also create custom workflows

7.3.1 Using an Existing Workflow



- ATWATER
  - Client Workflows
    - Acquisition
      - Network point creation
      - Network pipe creation
      - Network pipe with fitting point creation
      - House connection creation
      - Fire hydrant creation
      - Site or facility creation
      - Facility creation
      - Damage creation
      - Pressure zone creation
      - Protection creation
    - Analysis
      - Find Connected
      - Find Connected with Stop Conditions
      - Find non split point on pipe
    - Acquisition Cable
      - Control cable creation
      - Control point creation
    - Report
      - Report generation



Choose a reference of Hydrant

Use the last reference

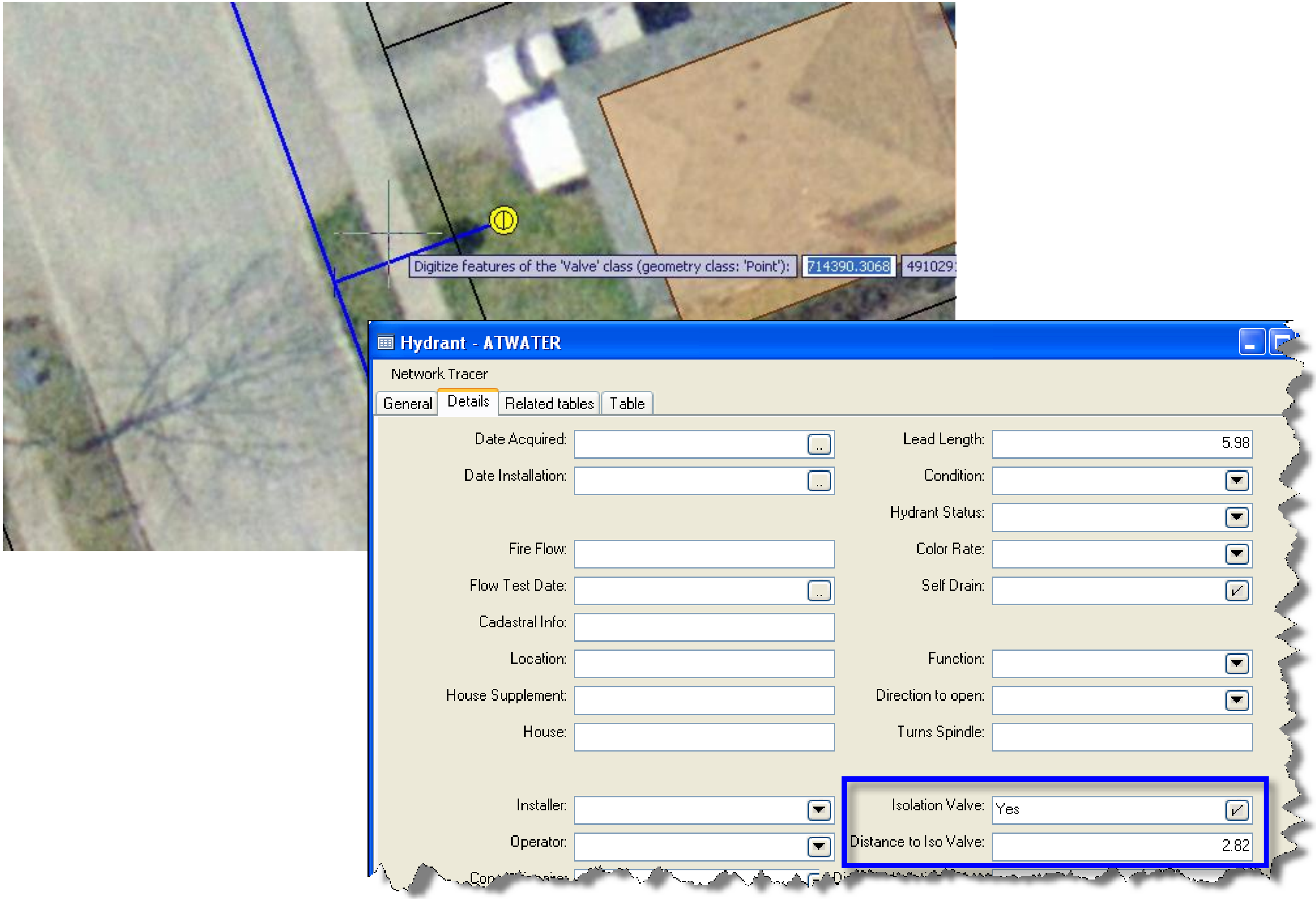
Pipe from hydrant to mainpipe

● Straight pipe
○ Digitize pipe

Choose the additional points to digitize

□ Armature
□ Valve

7.3.1 Using an Existing Workflow +



Digitize features of the 'Valve' class (geometry class: 'Point'): 714390.3068 491029

Hydrant - ATWATER

Date Acquired:

Date Installation:

Fire Flow:

Flow Test Date:

Cadastral Info:

Location:

House Supplement:

House:

Installer:

Operator:

Lead Length: 5.98

Condition:

Hydrant Status:

Color Rate:

Self Drain:

Function:

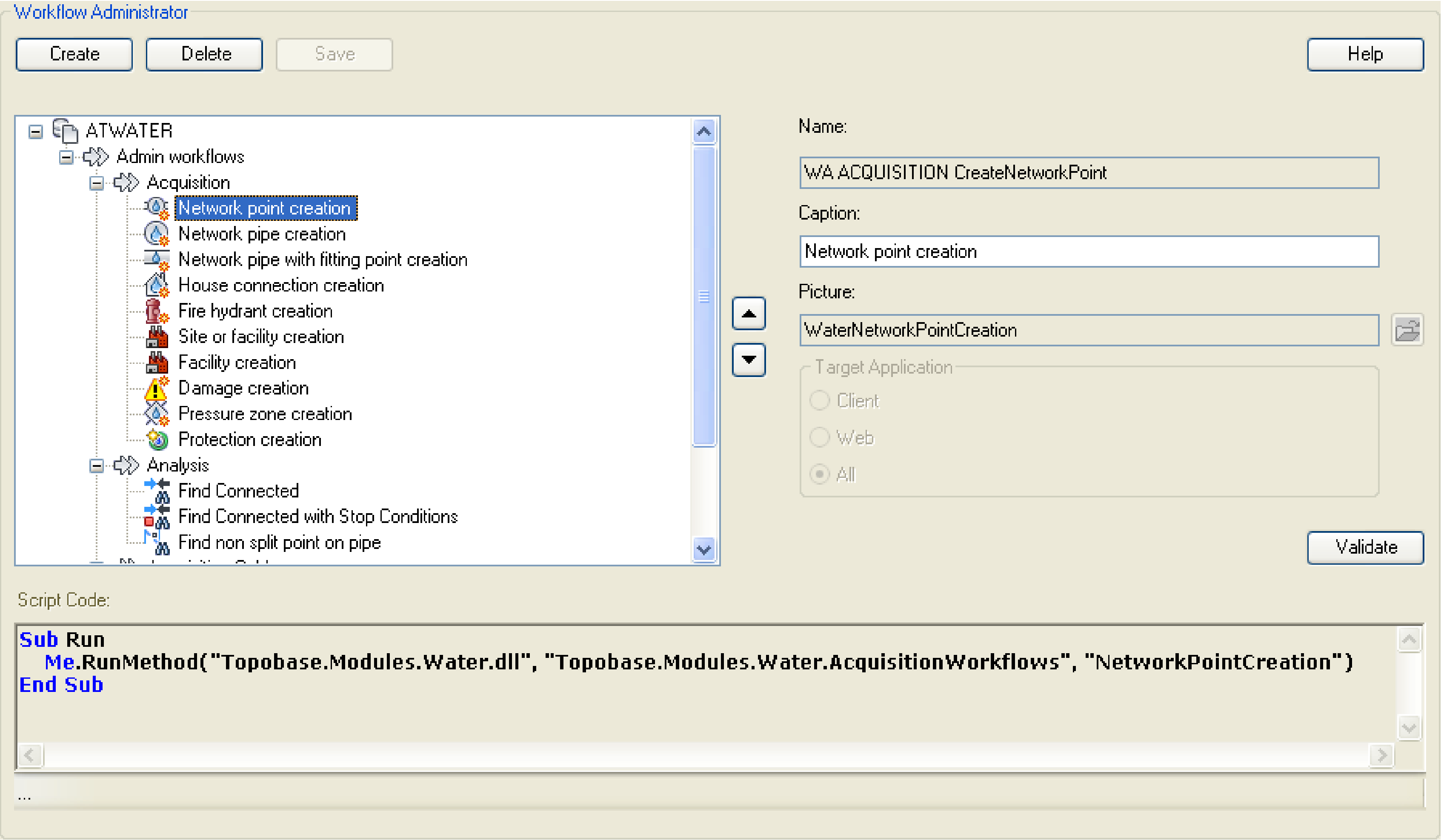
Direction to open:

Turns Spindle:

Isolation Valve: Yes

Distance to Iso Valve: 2.82

7.3.2 Browsing Predefined Workflows



Name: WA ACQUISITION CreateNetworkPoint

Caption: Network point creation

Picture: WaterNetworkPointCreation

Target Application

○ Client

○ Web

● All

Script Code:

```
Sub Run
   Me.RunMethod("Topobase.Modules.Water.dll", "Topobase.Modules.Water.AcquisitionWorkflows", "NetworkPointCreation")
End Sub
```

7.5 Chapter Summary

- You should now be able to:

  - Understand the purpose of feature rules
  - Create a server-side feature rule
  - Understand the value of using workflows
  - Start a workflow and follow workflow guidelines

Autodesk®