

LTX Envision Test Looping Modes

Setup Loop - Pressing this button brings-up a dialog box, illustrated in Figure 9.3, Setup Loop Dialog, on page 9-102, which allows the user to specify which Loop mode to use. Loops are useful for test program debug, and may also be used in the actual program execution.

The following selections are available in this dialog:

Loop Condition - This selection is an editable field that contains the Loop **Expression**. The Loop expression is a boolean expression that determines if the test will loop when run. If the Loop expression resolves to FALSE the test terminates when the end is reached, otherwise it restarts when the end is reached. Pressing the [Halt Loop](#) button forces the loop value to FALSE, which in turn causes the test to stop looping. The Loop **Value** field shows the current Loop Condition status and it will either be TRUE or FALSE.

Loop Mode - There are three different loop modes, selected by clicking Mouse button 1 on the arrow icon until the desired mode is selected:

- **Objects** - This is the outer loop for the Test object. When the test is looped, the Entry objects are loaded, the Test Method arguments are resolved, the Test Method is executed, the Port Expressions are resolved, and the Exit objects are loaded.
- **Arguments** - This is the middle loop for the Test object. It is just like the **Objects** loop mode with the exception that the Entry and Exit objects are not loaded each time through the loop.
- **TestMethod** - This is the inner loop for the Test object. Each time through the loop, after the first test is executed, ONLY the Test Method is restarted. The arguments for the TestMethod are not resolved each time. This is the tightest loop mode and should result in minimal overhead, which should make it suitable for scope loops. This mode should preferably be run with **Loop Update Mode** set to **Off**.

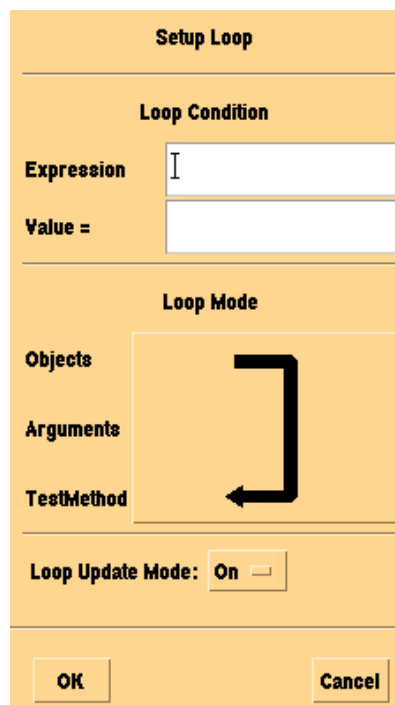


Figure 9.3: Setup Loop Dialog

LTX Envision Entry objects: **Levels**, **Pattern Sequence**, **Microflow**, or **Pin State** objects.

LTX Envision Exit objects: **Levels**, **Microflow**, or **Pin State** objects.

Pin State Object: An object which specifies the pin state and relay conditions for one or more pins, including power pins. The object can then be used as an entry or exit object for a Test or as part of a Suspended State Definition (when a pin is associated with a site on which testing has been suspended, in multi-site applications). The allowable states are:

Character	State	Description
.	PIN_STATE_NC	No change to the present setup. This is the default.
x	PIN_STATE_OFF	Force pin's driver state to off.
0	PIN_STATE_ON_LO	Force pin's driver state to on and drive low.
1	PIN_STATE_ON_HI	Force pin's driver state to on and drive high.
P	PIN_STATE_PAT	Force pin's driver state to follow the pattern system.
Z	PIN_STATE_DISC	Disconnect the pin from the DUT.

Table 31.1: Pin State Character Mnemonic Meaning

Applications include:

- Define the state of each device pin during a suspend state (when a site has been suspended during multi-site testing).
- As an Entry or Exit object in the Test tool, it can be used to preset (during Entry) or postset (during Exit) some special pin condition.

Levels and PatternSequence Objects: Equivalent to DCLevels, DCSets

Microflow Objects: Specialized flow objects and test methods in EnVision, which can be strung together as a subflow that can be inserted into the execution sequence of a TestMethod. The P1450.4 ability to create user-defined tests, together with a rich enough set of primitives from which to construct those user-defined tests, should be equivalent to Microflow objects.

```
Test <name> {
    Comment = "comment";
    Result = <expr>;
    LoopExpr = <expr>; (any valid expression)
    LoopNotify = <boolean>;
    LoopDepth = <name>; (where <name> presumably indicates which of the
choices shown above (object, argument, or testmethod) the loop
should encompass
    TestMethod = <name>;
    FocusCal = <name>;
    <statements>
}
```