

Chapter IV-9 — Dependencies

```
DoUpdate           // Make Igor recalculate formulae
Print recalculateThis // Prints updated value
End
```

Running this function prints the following to the history area:

```
•TestRecalculation()
1
2
```

The call to DoUpdate is needed because Igor recalculates dependency formulas only when no user-defined functions are running or when DoUpdate is called.

This function uses SetFormula to create the dependency because the := operator is not allowed in user-defined functions.

Wave Dependencies

The assignment statement:

```
dependentWaveName := formula
```

creates a dependency and links the dependency formula to the dependent wave. Whenever any change is made to any object in the formula, the contents of the dependent wave are updated.

The command

```
SetFormula dependentWaveName, "formula"
```

establishes the same dependency.

From the command line, you can use either a dependency assignment statement or SetFormula to establish a dependency. In a user-defined function, you must use SetFormula.

Cascading Dependencies

“Cascading dependencies” refers to the situation that arises when an object depends on a second object, which in turn depends on a third object, etc. When an object changes, all objects that directly depend on that object are updated, *and* objects that depend directly on those updated objects are updated until no more updates are needed.

You can use the Object Status dialog to investigate cascading dependencies.

Deleting a Dependency

A dependency is deleted when the dependent object is assigned a value using the = operator:

```
recalculateThis := dependsOnThis // Creates a dependency
recalculateThis = 0             // Deletes the dependency
```

This method of deleting a dependency does not work in user-defined functions. You must use the SetFormula operation.

For example:

```
Execute "recalculateThis = 0"
```

will delete the dependency even in a user-defined function.

You can also delete this dependency using the SetFormula operation.

```
SetFormula recalculateThis, ""
```