Index Range Notation

Index range, or "dot-dot", notation is one of the most important features in VPascal. It gives you the ability to write extremely compact and intuitive code by providing sophisticated array indexing capabilities. Index range notation was introduced in V++ 4.0.

Array Indexing
Index range, or "dot-dot" notation is used in VPascal modules to access rectangular portions of an image or sequence. Recall that to access a single pixel in an image requires code like this:

```
Image[ 100, 55 ] := 123 ;
```

This assigns the value 123 to the pixel at coordinate (100,55). Index-range notation extends this idea by specifying a range of indexes for the pixel's coordinates. For example:

```
Image[ 100..200, 55 ] := 123 ;
```

This statement assigns the number 123 to pixels whose x-locations are 100 to 200, and whose y-location is 55. In other words, it writes the number 123 into all pixels in the section of row 55 between columns 100 and 200.

To refer to an entire dimension, simply leave out the beginning and ending values (called terminals) of the range. For example, the following statement assigns 0 to pixels with x-positions 100 to 200, and all y-positions:

```
Image[ 100..200, .. ] := 0 ;
```

The lower terminal or upper terminal may be omitted, in which case the most extreme value is used. For example:

```
Image[ 50.., ..70 ] := 255 ;
```

This statement assigns 255 to all pixels with x-positions 50 and above, and all y-positions from 0 up to 70.

Index-range notation can be used in an expression, in which case the appropriate portion of the image is extracted. For example:

```
Image[ .., 0..3 ] := Image[ .., 4..7 ] ;
```

This statement replaces the first four lines in the image with the four lines immediately following.

Accessing Pixels the "Old Way"
In older versions of V++, the way to access a range of pixels was by using one of the "region" routines. For example, to copy a rectangular region from one image to another required code like this:

```
Temp := GetRegion( ImageA, 100, 200, 199, 299 ) ;
PutRegion( ImageB, Temp, 50, 60 ) ;
```

This can now be more concisely accomplished using this syntax:

```
ImageB[ 50..149, 60..159 ] := ImageA[ 100..199, 200..299 ] ;
```

PO Box 35-715 • Browns Bay • Auckland 1330 • New Zealand
Tel +64 (9) 478 5779 • Fax +64 (9) 479 4750
info@digitaloptics.co.nz • www.digitaloptics.co.nz

All of the old style ways to access pixels and regions are still available, but index-range notation is much simpler, and more powerful. The new syntax eliminates the need to use the following 20 routines:

```
GetRow, GetColumn, GetRegion, GetXYRegion, GetYZRegion, GetXZRegion,
GetXYZRegion, GetXLine, GetYLine, GetZLine, PutRow, PutColumn, PutRegion,
PutXYRegion, PutYZRegion, PutXZRegion, PutXYZRegion, PutXLine, PutYLine,
PutZLine
```

However, VPascal modules using these functions will continue to work normally.

Compatibility Issues
Index-range notation assumes that the pixel coordinates are always presented in the order x-y-z. Any trailing coordinates that are omitted are assumed to be "..".

For example,

```
Image[ 100..200 ] := 0 ;
```

is identical to

```
Image[ 100..200, .. ] := 0 ;
```

as well as

```
Image[ 100..200, .., .. ] := 0 ;
```

The only compatibility issue arising from this is that in previous versions of VPascal a single index into an array referred to a frame number.

For example:

```
{ Old-style extraction of frame 12 }
Image := Sequence[ 12 ] ;
```

is now interpreted as

```
Image := Sequence[ 12, .., .. ] ;
```

VPascal modules that rely on the previous interpretation will need minor modifications. To achieve the desired effect of extracting frame number 12 from a sequence, do this:

```
{ The new way to extract frame 12 }
Image := Sequence[ .., .., 12 ] ;
```

Also note that references to the elements of a row vector can still be made with only one index but references to column vector elements require two indexes. Old VPascal code that indexes a column vector with a single index will need to be updated.