Examples

```
String uniqueWaveName = UniqueName(baseWaveName, 1, 0)
String uniqueControlName = UniqueName("ctrl", 15, 0, "Panel0")
```

See Also

Object Names on page III-501, **Programming with Liberal Names** on page IV-168, **CreateDataObjectName**, **CheckName**, **CleanupName**

UnPadString

`UnPadString(`*`str`*`, `*`padValue`*`)`

The UnPadString function undoes the action of PadString. It returns a string identical to *str* except that trailing bytes of *padValue* are removed.

See Also

PadString

UnsetEnvironmentVariable

`UnsetEnvironmentVariable(`*`varName`*`)`

The UnsetEnvironmentVariable function deletes the variable named *varName* from the environment of Igor's process, if it exists

The function returns 0 if it succeeds or a nonzero value if it fails.

The UnsetEnvironmentVariable function was added in Igor Pro 7.00.

Parameters

varName	The name of an environment variable which does not need to actually exist. It must not be an empty string and may not contain an equals sign (=).

Details

The environment of Igor's process is composed of a set of key=value pairs that are known as environment variables.

The environment of Igor's process is composed of a set of key=value pairs that are known as environment variables. Any child process created by calling ExecuteScriptText inherits the environment variables of Igor's process.

UnsetEnvironmentVariable changes the environment variables present in Igor's process and any future process created by ExecuteScriptText but does not affect any other processes already created.

On Windows, environment variable names are case-insensitive. On other platforms, they are case-sensitive.

Examples

```
Variable result
result = SetEnvironmentVariable("SOME_VARIABLE", "15")       // Sets the variable
result = UnsetEnvironmentVariable("SOME_VARIABLE")           // Unsets the variable
```

See Also

GetEnvironmentVariable, **SetEnvironmentVariable**

Unwrap

`Unwrap `*`modulus`*`, `*`waveName`*` [, `*`waveName`*`]…`

The Unwrap operation scans through each named wave trying to undo the effect of a modulus operation.

Parameters

modulus is the value applied to the named waves through the **mod** function as if the command:

```
waveName = mod(waveName,modulus)
```

had been executed. It is this calculation which Unwrap attempts to undo.