ifAssociativeArray

Implemented By

- roAssociativeArray
- roSGNode

Supported Methods

- AddReplace(key as String, value as Dynamic) as Void
- Lookup(key as String) as Dynamic
- DoesExist(key as String) as Boolean
- Delete(key as String) as Boolean
- Clear() as Void
- Keys() as Object
- Items() as Object
- SetModeCaseSensitive() as Void
- LookupCI(key as String) as Dynamic
- Append(aa as Object)
- Count() As Integer

Description of Methods

AddReplace(key as String, value as Dynamic) as Void

Add a new entry to the array associating the supplied value with the supplied key string. Only one value may be associated with a key. If the key is already associated with a value, the existing value is discarded.

Lookup(key as String) as Dynamic

Return the value in the array associated with the specified key. If there is no value associated with the key then type "invalid" is returned. Key comparison is case-insensitive, unless SetModeCaseSensitive() has been called.

DoesExist(key as String) as Boolean

Look for an entry in the array associated with the specified key. If there is no associated object then false is returned. If there is such an object then true is returned.

Delete(key as String) as Boolean

Look for an entry in the array associated with the specified key. If there is such an value then it is deleted and true is returned. If not then false is returned.

Clear() as Void

Remove all key/values from the associative array.

Keys() as Object

Returns an array containing the associative array keys in lexicographical order.

This function is available in firmware 7.0 or later.

Items() as Object

Returns an array containing the associative array key/value pairs in lexicographical order of key.
Each item is in the returned array is an associative array with 'key' and 'value' fields.

This function is available in firmware 7.5 or later.

Examples

```
aa = {one:1, two:2, three:3}
for each item in aa.Items()
print item.key, item.value
end for
REM prints "one  1", "three  3", "two  2"
```

SetModeCaseSensitive() as Void

Associative Array lookups are case insensitive by default. This call makes all subsequent actions case sensitive.

LookupCI(key as String) as Dynamic

Same as "Lookup" except key comparison is always case insensitive, regardless of case mode.

Append(aa as Object)

Append an AssociativeArray to this one. If any key in aa is already associated with a value in this AssociativeArray, the current value is discarded and is replaced with the value in aa.

Count() As Integer

Returns the number of keys in the associative array.