Description
This tool function returns the string “os”. If you have a tool, but don’t know what kind it is, this function usually exists to tell you what kind of tool you have.

Example

```
  myos := os()
  print 'type=', myos.type()
# type=os
```

---

os.done - Function

Package	**utility**
Module	**misc**
Tool	**os**

end the os tool

Synopsis
done()

Returns
bool

1.6.3 aipsrc - Tool

Package	**utility**
Module	**misc**

tool to find aips resources from .aipsrc files
include ”aipsrc.g”

Constructors

aipsrc	Construct aipsrc tool

Functions

aipsarch	The AIPSARCH directory string
aipshome	The user's aips++ directory string
aipshost	The AIPSHOST directory string
aipsroot	The AIPSROOT directory string
aipssite	The AIPSSITE directory string
done	free resources used by tool
find	lookup a resource definition
findbool	lookup a boolean resource definition
findfloat	lookup a float value resource definition
findint	lookup a float value resource definition
findlist	lookup a value in a code list
id	identification of tool
init	re-read all the .aipsrc files
type	type of tool
tzoffset	The local time zone offset

Description

The aipsrc tool is used to interrogate the .aipsrc files (AIPS++resource files) to find values for a given keyword, and to connect with some of the more widely used keywords. More on aipsrc files can be found in the system manual, and in the aipsrcdata description.
Including the 'aipsrc.g' file creates a *defaultaipsrc* tool (also called **drc**.
Only in special cases it is necessary to create your own aipsrc tool.

Example

```
- include "aipsrc.g"
T
- drc.find(where, "help.directory")   # use default server
T
- arc := aipsrc()                     # create your own tool
- arc.find(where, "help.directory")
T
- print where
/aips++/daily/sun4sol_gnu/doc/aips2help
- arc.find(where, "does.not.exist")
F
- arc.tzoffset()
10
- arc.done()                          # delete your own tool
```

aipsrc - Constructor

Package	**utility**
Module	**misc**
Tool	**aipsrc**

Construct aipsrc tool

Synopsis
aipsrc(host, forceneweserver)

Arguments

host	in	host on which to run tool Allowed:valid host name string Default: ”
forceneweserver	in	force the use of a new server Allowed:bool Default: F

Returns
tool

Description
Create a quanta `tool`on the specified host (or by default the host you are running on).

---

aipsrc.find - Function

Package	**utility**
Module	**misc**
Tool	**aipsrc**

lookup a resource definition

Synopsis
find(value, keyword, def, usehome)

Description

find will return *T* if the specified keyword was found in an aipsrc file, *F* otherwise. If a find was made, its value will be returned. If a default value was specified it will be used if the aipsrc keyword was not present.

Arguments

value	The value of the resource (Out) Allowed:String
keyword	The name of the resource to look up (In) Allowed:String
def	The default value to use if no aipsrc resource found Allowed:String Default: F
usehome	Use the users own .aipsrc file? (In) Allowed:Bool Default: T

Returns
Bool – is the resource defined?

---

aipsrc.findbool - Function

Package	**utility**
Module	**misc**
Tool	**aipsrc**

lookup a boolean resource definition

Synopsis
findbool(value, keyword, def)

Description
findbool will return *T* if the specified keyword was found in an aipsrc file, *F* otherwise. If a find was made, its value is returned, otherwise the specified (or default) default value `def` will be used. The test for boolean is based on a first character being one of y Y t T 1 2 3 4 5 6 7 8 9 to return true, all others are interpreted as false.

Arguments

value	The value of the resource (Out) Allowed:Bool
keyword	The name of the resource to look up (In) Allowed:String
def	Default to use if keyword not found Allowed:Bool Default: T

Returns
Bool – is the resource defined?

---

aipsrc.findfloat - Function

Package	**utility**
Module	**misc**
Tool	**aipsrc**

lookup a float value resource definition

Synopsis
findfloat(value, keyword, def, undef, unres)

Description
findfloat will return T if the specified keyword was found in an aipsrc file, F otherwise. If a find was made, its value is returned, otherwise the specified (or default) default value `def` will be used. The value defined in the keyword is converting according to standard rules, i.e. the field upto a non-conforming character will be returned.
The value can be defined with optional units in the aipsrc file. By specifying in the findint a default unit string that has to be used in the absence of given units, and specifying units the result will be converted to before the value is returned. E.g.

```
drc.findfloat(x, 'test.float' , 5, 'h', 'd')
```

will assume that if the keyword is not present, the value of 5 is returned in x. If the keyword is given, but has no units specified, it will be assumed that the user has given the value in 'h'. The value returned will be converted to 'd' (i.e. twenty-four times smaller). If the user has specified units, the value returned will be converted to these ('d' in this case).

Example
With the following aipsrc keywords:

```
test.float: 5
test.float2: 5:30
test.float3: 5a
```

The following will be the result:

```
- arc:=aipsrc()
- arc.findfloat(x, 'test.float')
T
- x
5
- arc.findfloat(x, 'test.float1')
F
- x
0
- arc.findfloat(x, 'test.float1', 7)
F
- x
7
- arc.findfloat(x, 'test.float', undef='h', unres='d')
T
- x
0.208333333
- arc.findfloat(x, 'test.float2', undef='h', unres='d')
T
- x
0.229166667
- arc.findfloat(x, 'test.float3', undef='h', unres='d')
T
- x
1826.25
- arc.done()            # remove tool
```

Arguments

value	The value of the resource (Out) Allowed:Double
keyword	The name of the resource to look up (In) Allowed:String
def	Default to use if keyword not found Allowed:Numeric Default: 0
undef	Default units to use if none given in aipsrc file Allowed:valid unit string Default: F
unres	The units the result will be converted to before giving the value from aipsrc file Allowed:valid unit string Default: F

Returns
Bool – is the resource defined?

---

aipsrc.findint - Function

Package	**utility**
Module	**misc**
Tool	**aipsrc**

lookup a float value resource definition

Synopsis
findint(value, keyword, def, undef, unres)

Description
findint will return *T* if the specified keyword was found in an aipsrc file, *F* otherwise. If a find was made, its value is returned, otherwise the specified (or default) default value `def` will be used. The value defined in the keyword is read as a float, and converted according to standard rules.
The value can be defined with optional units in the aipsrc file. By specifying in the findint a default unit string that has to be used in the absence of given units, and specifying units the result will be converted to before the value is returned. E.g.

```
drc.findint(x, 'test.int' , 5, 'km', 'm')
```

will assume that if the keyword is not present, the value of 5 is returned in x. If the keyword is given, but has no units specified, it will be assumed that the user has given the value in 'km'. The value returned will be converted to 'm' (i.e. thousand times bigger). If the user has specified units, the value returned will be converted to these ('m' in this case).

Example
With the following aipsrc keywords:

```
test.int: 2.9
test.int2: 2.9hm
```

The following will be the result:

```
- arc:=aipsrc()
- arc.findint(x, 'test.int')
T
- x
2
- arc.findint(x, 'test.int1')
F
- x
0
- arc.findint(x, 'test.int1', 7)
F
- x
7
- arc.findint(x, 'test.int', 7, 'm', 'km')
T
- x
0
- arc.findint(x, 'test.int', 7, 'km', 'm')
T
- x
2900
- arc.findint(x, 'test.int1', 7, 'km', 'm')
F
- x
7
- arc.findint(x, 'test.int2', 7, 'km', 'm')
T
- x
290
- arc.findint(x, 'test.int2', 7, '', '')
T
- x
290
- arc.findint(x, 'test.int2', 7)
```

```
T
- x
2
- arc.done()            # remove tool
```

Arguments

value	The value of the resource (Out) Allowed:Int
keyword	The name of the resource to look up (In) Allowed:String
def	Default value to use if keyword not found Allowed:Numeric Default: 0
undef	Default units to use if none given in aipsrc file Allowed:valid unit string Default: F
unres	The units the result will be converted to before giving the value from aipsrc file Allowed:valid unit string Default: F

Returns
Bool – is the resource defined?

---

aipsrc.findlist - Function

Package	**utility**
Module	**misc**
Tool	**aipsrc**

lookup a value in a code list

Synopsis
findlist(value, keyword, vlist, def)

Description
findlist will return *T* if the specified keyword was found in an aipsrc file and the value conformed to one in the list given, *F* otherwise. If a find was made, its value is compared with the codes given in a list (comparison is case insensitive, and only enough characters have to be in the value to give a

unique match). The position in the list will be returned. If the keyword was not found, or if no match was found, a default value (if given) will be matched and its position returned. If the default is not given, or also does not match the code list, the returned value will be beyond the given list.

Example
With the following aipsrc keywords:

```
test.list: ScR
test.list1: test
```

The following will be the result:

```
- y := "none screen gui"
- drc.findlist(x, 'test.list', y, 'gui')
T
- x
2
- drc.findlist(x, 'test.list1', y, 'gui')
F
- x
3
- drc.findlist(x, 'test.list1', y)
F
- x
4
```

Arguments

value	The value of the resource (Out) Allowed:Int
keyword	The name of the resource to look up (In) Allowed:String
vlist	List of code values that will be recognised Allowed:String array
def	Default value to use if keyword not present, or value does not match code value list Allowed:String Default: "

Returns
Bool – is the resource defined and has correct code?

---

aipsrc.init - Function

Package	**utility**
Module	**misc**
Tool	**aipsrc**

re-read all the .aipsrc files

Synopsis
init()

---

aipsrc.aipsroot - Function

Package	**utility**
Module	**misc**
Tool	**aipsrc**

The AIPSROOT directory string

Synopsis
aipsroot()

Returns
String

---

aipsrc.aipsarch - Function

Package	**utility**
Module	**misc**
Tool	**aipsrc**

The AIPSARCH directory string

Synopsis
aipsarch()

Returns
String

---

aipsrc.aipssite - Function

Package	**utility**
Module	**misc**
Tool	**aipsrc**

The AIPSSITE directory string

Synopsis
aipssite()

Returns
String

---

aipsrc.aipshost - Function

Package	**utility**
Module	**misc**
Tool	**aipsrc**

The AIPSHOST directory string

Synopsis
aipshost()

Returns
String

---

aipsrc.aipshome - Function

Package	**utility**
Module	**misc**
Tool	**aipsrc**

The user's aips++ directory string

Synopsis
aipshome()

Description
aipshome will return the user's aips++ base directory. This will either be

```
~/aips++
```

or the value of the aipsrc keyword 'user.aipsdir' if it has been defined.

Returns
String

---

aipsrc.tzoffset - Function

Package	**utility**
Module	**misc**
Tool	**aipsrc**

The local time zone offset

Synopsis
tzoffset()

Description
tzoffset will return the local time zone offset in hours (East, i.e. add it to UTC).

Returns
Double – offset in hours

---

aipsrc.id - Function

Package	**utility**
Module	**misc**
Tool	**aipsrc**

identification of tool

Synopsis
id()

Arguments

Returns
object id

Description
id will return the object identification. Its use is meant for the toolmanager.

Example

```
drc.id()
[sequence=1, pid=5765, time=1020744553, host=cetus, agentid=2]
```

---

aipsrc.type - Function

Package	**utility**
Module	**misc**
Tool	**aipsrc**

type of tool

Synopsis
type()

Arguments

Returns
string

Description
type will return the tool name.

Example

```
- drc.type()
aipsrc
```

---

aipsrc.done - Function

Package	**utility**
Module	**misc**
Tool	**aipsrc**

free resources used by tool

Synopsis
done(kill)

Arguments

kill	in	force kill of the default tool Allowed:bool Default: F

Returns
bool or fail

Description

If you have created your own aipsrc `tool` (rather than using the default *drc* one), and you no longer need it, its resources can be destroyed by the `done` function. Note that the default `tool` can only be destriyed by setting the `kill` argument explicitly to T.

Example