Description
Brief by default prints only the message, and suppresses time, priority, and origin.

Arguments

show_time	in	Allowed: boolean Default: F
show_priority	in	Allowed: boolean Default: F
show_origin	in	Allowed: boolean Default: F

Returns
Bool

1.8.8 types - Tool

Package utility
Module tasking

Tool for defining types known to the tasking system
include "types.g"

Description

`types` is the means for defining the user interface (i.e., the functions and arguments) of other AIPS++tools.
An AIPS++tool or function usually takes various types of arguments. Glish itself is type-less and doesn't require that the type of a variable be defined explicitly or fixed, which has advantages and disadvantages. In the tasking system, it is important to impose some typing of variables so that, for example, the toolmanager knows whether a particular argument is a scalar or a measure or a file name. The `types` tool provides the mechanism for defining the type to be used in the `toolmanager`'s user interface.
This works as follows:

- The developer specifies the interface in a meta information file.

- In the AIPS++startup, the `types` tool reads these meta information files, and stores the results in a cached file.
- The functions of `types` may be used to recall information on the interface.
- User interfaces for tools and functions are constructed by the toolmanager using the meta information and passed to the autogui or autocli.
- The user interface does parsing as appropiate.
- If the tool is linked to C++, arguments are passed appropriately.

The net result for the developer is that a user interface for any tool or function may be generated easily merely by provision of a meta information file.
An example of a meta information file follows. This defines the interface for a tool called imager and hence should be located in a file called imager_meta.g.

```
pragma include once; # Include once only

include 'types.g'; # Required

#
# Name the include file that defines the tool
types.class('imager').includefile('imager.g');
#
# Constructor names are preceeded by ctor_. Then follows the
# list of arguments.  Each argument is defined with a
# function call of this form:
#
# type(arg, default=unset, dir='in', checkeval=T)
#
# where type is defined in type.g and is one of:
#
# file|table|string|integer|float|double|record|boolean|vector_string|complex
#
# arg is the name of the argument
# default sets default values (may be an executable string)
# dir sets the direction of the argument
# checkeval=T means that a string value is eval'ed and checked,
# otherwise no such check is done.
#
# The functions may be assigned to groups using the
# group('mygroup') function. A group() call resets the
# group to the default, which is 'basic'.
#
# Define two constructors, one with just a table name, and the other
```

```
# with more arguments
types.method('ctor_imager').table('filename').
    method('ctor_imagertester').string('filename', '3C273XC1.ms').
    integer('size', 256).string('cell', '0.7arcsec').string('stokes', 'IV').
    string('coordinates', 'b1950');

# Methods
# Open with just a table name
types.group('basic').method('open').table('thems');

# Plain close
types.method('close');

# Get the name back
types.method('name').string('return');

# Summary
types.method('summary');

# Set various image parameters. Note the use of the measures
# tool to set the phasecenter.
types.group('setup').method('setimage').integer('nx',128).integer('ny',128).
    string('cellx','1arcsec').string('celly','1arcsec').
    string('stokes', 'I').
    boolean('doshift', F).
    record('phasecenter', 'dm.direction(\'b1950\', \'0d\', \'0d\')',checkeval=F).
    string('shiftx','0arcsec').string('shifty','0arcsec').
    string('mode', 'mfs').integer('nchan', 1).integer('start', 1).
    integer('step', 1).string('mstart', '0km/s').string('mstep', '0km/s').
    integer('spwid', 1).integer('fieldid', 1);
```

Constructors

Functions

addmeta	Copy meta information to a class
all	Show all classes
antennas	Define a list of antennas
attachhelp	Attach help from the help system
baselines	Define a list of baselines
boolean	Define a boolean
check	Choose one or more from a list of strings
choice	Choose one from a list of strings
class	Define a class or type of tool
classes	Return names of the classes

complex	Define a complex
direction	Define a direction measure
directory	Define a directory
double	Define a double
epoch	Define an epoch measure
exists	Does the class, method, arg exist?
fields	Define a list of field ids
file	Define a file
float	Define a float
getincludefile	Get the include file for a class
group	Define the group name for a function
includefile	Specify the include file for a class
includemeta	Include all meta information files
integer	Define an integer
is_tool	Is this a tool?
list	Define a list of other types
measure	Define a measure
measurecodes	Define the reference code for a measure
meta	Return meta information for a type of tool
method	Define a method or function for a class
msselect	Define a taql string for a MeasurementSet
position	Define a position measure
quantity	Define a quantity
record	Define a record
region	Define a region
spectralwindows	Define a list of spectralwindows
string	Define a string
table	Define a table
taql	Define a Table Query Language string
untyped	Define an untyped variable
vector_boolean	Define a vector of booleans
vector_complex	Define a vector of complexs
vector_double	Define a vector of doubles
vector_float	Define a vector of floats
vector_integer	Define a vector of integer
vector_string	Define a vector of strings

types.exists - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Does the class, method, arg exist?

Synopsis
exists(class, method, arg)

Description
Does the class, method, arg exist? class may be composite.

Arguments

class	Class or type of tool Allowed:String
method	Method Allowed:String Default: unset
arg	Argument Allowed:String Default: unset

Example

```
- types.exists('imager','advise','cell')
T
- types.exists('imager.advise')
T
- types.exists('imager.integrate')
F
```

---

types.classes - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Return names of the classes

Synopsis
classes()

---

types.class - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a class or type of tool

Synopsis
class(class)

Arguments

class	Class or type of tool Allowed:String

---

types.group - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define the group name for a function

Synopsis
group(group)

Description
Functions may be assigned to names groups in order to bring some structure to the user interface. The default name is 'basic'.

Arguments

group	Group name Allowed:String Default: basic

---

types.includefile - Function

Package	**utility**

Module	**tasking**
Tool	**types**

Specify the include file for a class

Synopsis
includefile(file, class)

Arguments

file	File name Allowed:String
class	Class or type of tool Allowed:String Default: unset

---

types.getincludefile - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Get the include file for a class

Synopsis
getincludefile(class)

Arguments

class	Class or type of tool Allowed:String Default: unset

---

types.method - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a method or function for a class

Synopsis
method(method, title, label, category)

Arguments

method	Name of method Allowed:String
title	Title Allowed:String Default: unset
label	label Allowed:String Default: unset
category	Category Allowed:String Default: unset

---

types.list - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a list of other types

Synopsis
list(arg, types, options, name, names, default, dir, checkeval, help, allowunset)

Description
This widget allows the assembly of types containing zero or more composite types, where the composite type can be made of one or more other types. The user may then add or delete these composite types.
Any number of types may be included in a composite type. The corresponding arguments are given in either vector string or record form.
The test function typestest demonstrates a list type.

Arguments

arg	List of arguments Allowed:String
types	List of types Allowed:String
options	Record of options Allowed:Record
name	Name of the composite type Allowed:Vector String
names	Names of individual types Allowed:Vector String
default	Default values Allowed:Region Default: unset
dir	Direction of variable Allowed:String: 'in'—'inout'—'out' Default: 'in'
checkeval	Check eval'ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

Example

```
# The following defines a list containing two other types: string
# and direction.
    list('Sources', types=['string', 'direction'], names="Source Direction",
 name='Source', default=[Name='3C273', Direction=dm.direction()],
 help=['Source name', 'Source direction'],
 allowunset=T,
 options=unset);
```

---

types.integer - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define an integer

Synopsis
integer(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:Integer Default: 1
dir	Direction of variable Allowed:String: 'in'—'inout'—'out' Default: 'in'
checkeval	Check eval'ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.vector_integer - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a vector of integer

Synopsis
vector_integer(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:Vector of Int Default: []
dir	Direction of variable Allowed:String: 'in'—'inout'—'out' Default: 'in'
checkeval	Check eval'ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.float - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a float

Synopsis
float(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:Float Default: 0.0
dir	Direction of variable Allowed:String: 'in'—'inout'—'out' Default: 'in'
checkeval	Check eval'ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.vector_float - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a vector of floats

Synopsis
vector_float(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:Vector of floats Default: []
dir	Direction of variable Allowed:String: 'in'—'inout'—'out' Default: 'in'
checkeval	Check eval'ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.double - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a double

Synopsis
double(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:Double Default: 0.0
dir	Direction of variable Allowed:String: 'in'—'inout'—'out' Default: 'in'
checkeval	Check eval'ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.vector_double - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a vector of doubles

Synopsis
vector_double(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:Vector of doubles Default: []
dir	Direction of variable Allowed:String: 'in'—'inout'—'out' Default: 'in'
checkeval	Check eval'ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.complex - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a complex

Synopsis
complex(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:Complex Default: 0+0i
dir	Direction of variable Allowed:String: 'in'—'inout'—'out' Default: 'in'
checkeval	Check eval'ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.vector_complex - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a vector of complexs

Synopsis
vector_complex(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:Vector of complexs Default: []
dir	Direction of variable Allowed:String: 'in'—'inout'—'out' Default: 'in'
checkeval	Check eval'ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.string - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a string

Synopsis
string(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:String Default: ”
dir	Direction of variable Allowed:String: ’in’—’inout’—’out’ Default: ’in’
checkeval	Check eval’ability Allowed:Boolean Default: F
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.vector_string - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a vector of strings

Synopsis
vector_string(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:Vector of strings Default: [”]
dir	Direction of variable Allowed:String: ’in’—’inout’—’out’ Default: ’in’
checkeval	Check eval’ability Allowed:Boolean Default: F
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.measure - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a measure

Synopsis
measure(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:Measure Default: 'dm.direction('b1950', '0deg', '0deg')'
dir	Direction of variable Allowed:String: 'in'—'inout'—'out' Default: 'in'
checkeval	Check eval'ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.measurecodes - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define the reference code for a measure

Synopsis

measurecodes(arg, default, dir, options, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:Measure Default: 'dm.direction('b1950', '0deg', '0deg')'
dir	Direction of variable Allowed:String: 'in'—'inout'—'out' Default: 'in'
options	List of options Allowed:Vector of strings Default: unset
checkeval	Check eval'ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.epoch - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define an epoch measure

Synopsis
epoch(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:Measure Default: dm.epoch('utc', 'today')
dir	Direction of variable Allowed:String: 'in'—'inout'—'out' Default: 'in'
checkeval	Check eval'ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.position - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a position measure

Synopsis
position(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:Measure Default: dm.observatory('vla')
dir	Direction of variable Allowed:String: 'in'—'inout'—'out' Default: 'in'
checkeval	Check eval'ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.direction - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a direction measure

Synopsis
direction(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:Measure Default: dm.direction('b1950', '0deg', '0deg')
dir	Direction of variable Allowed:String: 'in'—'inout'—'out' Default: 'in'
checkeval	Check eval'ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.quantity - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a quantity

Synopsis
quantity(arg, default, dir, checkeval, help, options, allowunset)

Description
The type of the quantity is fixed on construction. It is determined by the type of the default, or if that is unset, from the options string.

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:Quantity Default: '0deg'
dir	Direction of variable Allowed:String: 'in'—'inout'—'out' Default: 'in'
checkeval	Check eval'ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
options	Default type Allowed:Vector of strings Default: angle—time—flux—unnormalizedfluxdensity—vel—long—lat—len—temp
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.choice - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Choose one from a list of strings

Synopsis
choice(arg, default, options, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:String Default: ”
options	List of options Allowed:Vector of strings Default: unset
dir	Direction of variable Allowed:String: ’in’—’inout’—’out’ Default: ’in’
checkeval	Check eval’ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.check - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Choose one or more from a list of strings

Synopsis
check(arg, default, options, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:String Default: ”
options	List of options Allowed:Vector of strings Default: unset
dir	Direction of variable Allowed:String: ’in’—’inout’—’out’ Default: ’in’
checkeval	Check eval’ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.boolean - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a boolean

Synopsis
boolean(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:Boolean Default: T
dir	Direction of variable Allowed:String: 'in'—'inout'—'out' Default: 'in'
checkeval	Check eval'ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.vector_boolean - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a vector of booleans

Synopsis
vector_boolean(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:Vector of booleans Default: []
dir	Direction of variable Allowed:String: 'in'—'inout'—'out' Default: 'in'
checkeval	Check eval'ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.record - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a record

Synopsis
record(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:Record Default: [=]
dir	Direction of variable Allowed:String: 'in'—'inout'—'out' Default: 'in'
checkeval	Check eval'ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.table - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a table

Synopsis
table(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:String Default: ”
dir	Direction of variable Allowed:String: ’in’—’inout’—’out’ Default: ’in’
checkeval	Check eval’ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.taql - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a Table Query Language string

Synopsis
taql(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:String Default: ”
dir	Direction of variable Allowed:String: ’in’—’inout’—’out’ Default: ’in’
checkeval	Check eval’ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.msselect - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a taql string for a MeasurementSet

Synopsis
msselect(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:String Default: ”
dir	Direction of variable Allowed:String: ’in’—’inout’—’out’ Default: ’in’
checkeval	Check eval’ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.file - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a file

Synopsis
file(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:String Default: ”
dir	Direction of variable Allowed:String: ’in’—’inout’—’out’ Default: ’in’
checkeval	Check eval’ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.directory - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a directory

Synopsis
directory(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:String Default: ”
dir	Direction of variable Allowed:String: ’in’—’inout’—’out’ Default: ’in’
checkeval	Check eval’ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.region - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a region

Synopsis
region(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:Region Default: unset
dir	Direction of variable Allowed:String: 'in'—'inout'—'out' Default: 'in'
checkeval	Check eval'ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.antennas - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a list of antennas

Synopsis

antennas(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:Region Default: unset
dir	Direction of variable Allowed:String: 'in'—'inout'—'out' Default: 'in'
checkeval	Check eval'ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.baselines - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a list of baselines

Synopsis
baselines(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:Region Default: unset
dir	Direction of variable Allowed:String: 'in'—'inout'—'out' Default: 'in'
checkeval	Check eval'ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.fields - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a list of field ids

Synopsis

fields(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:Region Default: unset
dir	Direction of variable Allowed:String: 'in'—'inout'—'out' Default: 'in'
checkeval	Check eval'ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.spectralwindows - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define a list of spectralwindows

Synopsis

spectralwindows(arg, default, dir, checkeval, help, allowunset)

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:Region Default: unset
dir	Direction of variable Allowed:String: 'in'—'inout'—'out' Default: 'in'
checkeval	Check eval'ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.untyped - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Define an untyped variable

Synopsis
untyped(arg, default, dir, checkeval, help, options, allowunset)

Description
This is a catch-all for variables that may be of many different types (such as return from the table tool. Don't use it in place of defining the correct type. The behavior is that:

- In determining the type of a given input string or insert argument, various conversions are tried, and the first successful one retained.
- The search list of conversions can be set using the options string.
- The default search list is "measure array coordinates region record string untyped"
- Valid search list items are the entry types that the Entryparser `tool` recognizes.

Arguments

arg	List of arguments Allowed:String
default	Default value Allowed:Any type Default: ”
dir	Direction of variable Allowed:String: ’in’—’inout’—’out’ Default: ’in’
checkeval	Check eval’ability Allowed:Boolean Default: T
help	Help string: if unset, help file is used Allowed:String Default: unset
options	Search order string Allowed:Vector of strings Default: unset
allowunset	Allow unset values? Allowed:Boolean Default: F

---

types.meta - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Return meta information for a type of tool

Synopsis
meta(class, ctors, addhelp)

Description
The type information read from the meta-information file is returned in a record.

Arguments

class	Class or type of tool Allowed:String
ctors	Show constructors Allowed:Boolean Default: F
addhelp	Add the help information? Allowed:Boolean Default: T

Example

```
- types.meta('imager', T, F)
[imager=[data=[toolname=[type=string, default=myimager,
parameters=[ptype=string, allowunset=F, default=myimager,
value=myimager, dir=in, dlformat=toolname, listname=toolname],
help=[text=Glish name for the constructed tool]],
filename=[type=table, default=, parameters=[ptype=table,
popt=Measurement Set, allowunset=F, default=, value=, dir=in,
dlformat=filename, listname=filename]]], gui=autogui, cli=autocli,
group=basic, title=Construct an imager tool],
imagertester=[data=[toolname=[type=string, default=myimager,
parameters=[ptype=string, allowunset=F, default=myimager,
value=myimager, dir=in, dlformat=toolname, listname=toolname],
help=[text=Glish name for the constructed tool]],
filename=[type=table, default=3C273XC1.ms, parameters=[ptype=table,
popt=Measurement Set, allowunset=F, default=3C273XC1.ms,
value=3C273XC1.ms, dir=in, dlformat=filename, listname=filename],
help=[text=Name of MS to create]], size=[type=integer, default=256,
parameters=[ptype=scalar, allowunset=F, default=256, value=256,
dir=in, dlformat=size, listname=size], help=[text=Number of pixels on
a side]], cell=[type=string, default=0.7arcsec,
parameters=[ptype=quantity, popt=[i_am_unset=i_am_unset],
allowunset=F, default=0.7arcsec, value=0.7arcsec, dir=in,
dlformat=cell, listname=cell]], stokes=[type=string, default=IV,
parameters=[ptype=choice, popt=I IV IQUV, allowunset=F, default=IV,
value=IV, dir=in, dlformat=stokes, listname=stokes], help=[text=Stokes
parameters to image (e.g. ' IQUV' )]], coordinates=[type=string,
default=b1950, parameters=[ptype=choice, popt=b1950 gal, allowunset=F,
default=b1950, value=b1950, dir=in, dlformat=coordinates,
listname=coordinates], help=[text=Coordinate system to use e.g. '
j2000,' gal' ]]], gui=autogui, cli=autocli, group=basic,
title=Construct an imager tool for testing]]
- field_names(types.meta('imager', F, F).advise)
data gui cli group title
```

Returns
Record

---

types.addmeta - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Copy meta information to a class

Synopsis
addmeta(class, meta)

Arguments

class	Class or type of tool Allowed:String
meta	Meta-information Allowed:Record

Returns
Boolean

---

types.all - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Show all classes

Synopsis
all(class, meta)

Arguments

class	Class or type of tool Allowed:String
meta	Meta-information Allowed:Record

Returns
String

---

types.is_tool - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Is this a tool?

Synopsis
is_tool(obj, type)

Description
Is this a tool? It must be a record of public functions only. If the type is specified, it will be tested. Then consistency with the meta-information is checked: all functions in the meta information must appear in the object (but not the other way around).

Arguments

obj	Possible object Allowed:Variable
type	Type of variable Allowed:String Default: 'unset'

---

types.attachhelp - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Attach help from the help system

Synopsis
attachhelp(type)

Description
Help information may be defined in the meta-information file or may be copied from that in the .help files.

Arguments

type	Type of tool Allowed:String

---

types.includemeta - Function

Package	**utility**
Module	**tasking**
Tool	**types**

Include all meta information files

Synopsis
includemeta(includepath)

Description
Force inclusion of the meta information files so that the global types function is up to date.

Arguments

includepath	Path for include files Allowed:String Default: unset

---
