sditerator	The sditerator to averager Allowed:sditerator

Returns
boolean

---

sdaverager.average - Function

Package	**dish**
Module	**dish**
Tool	**sdaverager**

Get the current internal accumulation

Synopsis
average(data, tsys, crpix, crval, cdelt, cunit, ctype, veldef, restfrequency, exposure, duration)

Arguments

data	The average data array
tsys	The average system temperature
crpix	X axis reference pixel of the average
crval	X axis value at the reference pixel
cdelt	X axis increment per pixel
cunit	X axis units
ctype	X axis type
veldef	Velocity definition
restfrequency	The rest frequency
exposure	The effective integration time
duration	The total integration time

Returns
boolean

1.1.8 sditerator - Tool

Package	**dish**
Module	**dish**

A `tool` for interacting with single dish data.
include "sditerator.g"

Constructors

newsditerator	Make a new, empty sditerator.
sditerator	An sditerator from an existing file on disk.
sditfromsdit	An sditerator from another sditerator.

Functions

appendHistory	Add a history element to the global sditerator history.
appendrec	Put an sdrecord at the end of the sditerator.
correcteddata	Is the CORRECTED_DATA column being used?.
deepcopy	Make a true copy of this data set.
deleterec	Delete the sdrecord at the current location pointer.
done	Close this sditerator and disconnect it from the data set.
flush	Flush the contents to disk.
get	Get an sdrecord.
getdata	Get the data part of the sdrecord.
getdesc	Get the desc part of the field of the sdrecord.
getdescvector	Get a vector of values from the desc record.
getempty	Get an empty sdrecord.
getheader	Get the header part of the sdrecord.
getheadervector	Get a vector of values from the header record.
getother	Get the other part of the sdrecord.
getvectors	Get a vector of values for the fields in the template.
history	Get the global history associated with all sdrecords
id	Return the ID record of this sditerator.
iswritable	Can sdrecords be stored (written) to this sditerator?
length	Get the total number of sdrecords in this sditerator.
location	Get the value of the location pointer.
lock	Set the appropriate lock on all associated tables.
more	Are there more sdrecords after this location?
name	What is the name of the underlying data on disk.
next	Set the location pointer to the next record.
origin	Set the location pointer to the first record.
previous	Set the location pointer to the previous record.
put	Put an sdrecord at the current location pointer.
reselect	Re-apply the previous selection to the parent data set.
resync	Resynchronize with the data on disk.
sel2string	Convert a selection record to a string.
select	Make a new sditerator by applying a selection criteria record
setlocation	Set the location pointer to the indicated record.
stringfields	Returns a record indicating which fields in the sditerator are strings.
type	Return the type of the underlying data.

unlock	Release the lock on all associated tables.
usecorrecteddata	Use the CORRECTED_DATA column?.

Description
This `tool` is intended to be used with the dish `tool` . It provides access to single dish data sets. One can step (iterate) through a data set with this `tool` . Two types of single dish data sets are supported: MeasurementSet and a flat table modeled on the SDFITS convention. Data is copied to and from an SDRecord, which is a *Glish* record structure with a set of fixed fields plus a few optional fields which vary depending on the data source. Data can only be saved back to an sditerator using a flat table as the underlying data source. Selection is supported either through a constructor or through the select function.

The SDRecord structure
The SDRecord is the fundamental data atom of this `tool` as well as of dish. An SDRecord is a *Glish* record having the following fields:

data A *Glish* record containing the data and things which describe that data (axes, units, flags, weights, etc.). The following fields are found in **data**:

- **arr** The data array. Its dimensions are (nstokes, nchan) where nchan is the number of frequency channels and nstokes is the number of stokes pixels (polarizations). The type may be real or complex.
- **flag** A boolean array with the same shape as **arr**. A value of `T` (true) for any pixel here means the corresponding pixel in **arr** has been flagged as bad.
- **sigma** An array of floats having the same shape as **arr**.
- **weight** An array of floats having the same shape as **arr**.
- **desc** A *Glish* record describing **arr** (axes and units). The following fields are found in **desc**:
  - **chan_freq** A vector of quantities giving each frequency value along the first axis of **arr**. See the quanta `tool` for a description of how to extract the values from a quantity as well as how to work with quantities.
  - **refframe** A string indicate the frequency measures reference frame for the **chan_freq** values.
  - **reffrequency** A reference frequency for the frequency axis. This has the same units as the bf chan_freq values do.
  - **chan_width** The channel width at **reffrequency** for use in selection. This has the same units as the bf chan_freq values do.
  - **restfrequency** A rest frequency associated with this data. This has the same units as the bf chan_freq values do.

corr_type A vector of strings giving the polarization type for each pixel of the stokes axis in **arr**.

units The units of **arr**.

header A *Glish* record containing associated header information common to all sditerators. The following fields are found in **header**:

time An epoch measure giving the midpoint of this integration. See the measures `tool` for a description of how to extract the values from a measure as well as how to work with measures.

scan_number An integer.

source_name

direction A direction measure showing where the telescope was pointed during this integration.

refdirection A direction measure for any reference (off) direction if position switched data has been calibrated.

veldef The velocity definition appropriate to this data.

transition A string describing the transition, if available.

exposure The effective integration time, in seconds.

duration The total time, including any blanking time, spent taking this data, in seconds.

observer

project

resolution The channel resolution at the **reffrequency** in Hz.

bandwidth The total bandwith in Hz.

tcal A vector of CAL temperatures - one for each pixel on the stokes axis, in K.

trx A vector of receiver temperatures - one for each pixel on the stokes axis, in K.

tsys A vector of system temperatures - one for each pixel on the stokes axis, in K.

telescope

telescope_position A position measure holding the location of the telescope.

pressure The atmospheric pressure in hPa.

dewpoint The dewpoint in K.

tambient The ambient temperature in K.

wind_dir The wind direction in radians.

wind_speed The wind speed in m/s.

azel A direction measure holding the azimuth and elevation for this integration (at **time**).

hist A vector of strings holding the history of operations on this data.

other A *Glish* record containing other information from the data set not found elsewhere in this structure. The contents of this field depend on the data source. If the source is a MeasurementSet, this record contains sub-records holding the appropriate values for the various sub-tables in that MeasurementSet. There will be one sub-record for each sub-table and the fields in each sub-record are the same as the column names in that sub-table. If the source is a flat table following the SDFITS convention, then any field not used in the **header** or **data** records is copied to an **sdfits** sub-record of the **other** record.

Example

```
sdit := sditerator('dishdemo2')
nrecords := sdit.length()
scans := array(0,nrecords)
sources := array("",nrecords)
count := 1
rec := sdit.get()
scans[count] := rec.header.scan_number
sources[count] := rec.header.source_name
while (sdit.more()) {
   sdit.next();
   rec := sdit.get()
   count +:= 1
   scans[count] := rec.header.scan_number
   sources[count] := rec.header.source_name
}
# or alternatively ...
vec := sdit.getheadervector("scan_number source_name")
scans := vec.scan_number
sources := vec.source_name
```

The first line makes an sditerator `tool` from a file. The next several lines show how the scan numbers and source names for each sdrecord in that iterator can be retrieved by iterating through the entire sditerator. The final three lines show how all of that could be done using the getheadervector function.

See Also

dish reference manual Using Dish

sditerator - Constructor

Package	**dish**
Module	**dish**
Tool	**sditerator**

An sditerator from an existing file on disk.

Synopsis

sditerator(filename, readonly, selection, lockoptions, correcteddata)

Description

This constructor is used to open an already existing single dish data set. The type of data set (flat-table or MeasurementSet) is automatically determined. MeasurementSets can only be opened read-only. The lockoptions are the same as those appropriate for the table `tool` . The selection record follows the same structure as that used by the select function. The CORRECTED_DATA column is only available for MeasurementSet-based sditerators. When CORRECTED_DATA is requested, the type of the **arr** field in the sdrecord will be complex.

Arguments

filename	in	Name of SDFITS file on disk Allowed:String
readonly	in	Open Read-only? Allowed:Bool Default: T
selection	in	Record describing any selections to do on opening Allowed:*Glish* record Default: [=]
lockoptions	in	locking to be used Allowed:String Default: auto
correcteddata	in	Use the CORRECTED_DATA column? Allowed:Bool Default: F

Returns

sditerator `tool`

Example

```
sdit := sditerator("a302_SP",correcteddata=T);
r := sdit.get();
```

The first line opens an existing data set, using the CORRECTED_DATA column to supply the values for the **arr** field in the returned sdrecord. The second line fetches the first sdrecord found in that sditerator. The type of r.data.arr is complex.

---

newsditerator - Constructor

Package	**dish**
Module	**dish**
Tool	**sditerator**

Make a new, empty sditerator.

Synopsis
newsditerator(filename, type, lockoptions)

Description
Create a new, empty data set of the indicated type on disk and attach an sditerator to it. Currently, only flat-tables can be created. In the future, other types will be available.

Arguments

filename	in	Name of new data set on disk. Allowed:String
type	in	The type of data set. Allowed:String Default: 'Table'
lockoptions	in	locking to be used Allowed:String Default: auto

Returns
sditerator `tool`

Example

```
sdit := sditerator("a302_SP",correcteddata=T);
newit := newsditerator("a302_SP.copy");
for (i in 1:sdit.length()) {
   newit.appendrec(sdit.get());
   sdit.next();
}
```

This example shows how a new sditerator is created and sdrecords from a existing sditerator are copied to it. The first line opens the existing data set. The second line creates a new data set. The loop copies each record from the existing sditerator to the new one.

---

sditfromsdit - Constructor

Package	**dish**
Module	**dish**
Tool	**sditerator**

An sditerator from another sditerator.

Synopsis

sditfromsdit(sdit, selection)

Description

This constructor is equivalent to the select function.

Arguments

sdit	in	An sditerator Allowed:sditerator `tool`
selection	in	Record describing any selections to do on opening Allowed: *Glish* record Default: [=]

Returns

sditerator `tool`

Example

```
sdit := sditerator("a302_SP");
selrec := [=];
selrec.header := [=];
selrec.header.source_name := "My source";
sditSource := sditfromsdit(sdit,selrec)
```

The first line opens an existing data set. The rest make another sditerator by making and applying a selection record to the first iterator. In this example, sources matching the indicated name are selected. The resulting iterator will only have sdrecords which match that selection criteria.

---

sditerator.appendHistory - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Add a history element to the global sditerator history.

Synopsis
appendHistory(newHistory)

Description
This adds the argument to the global history associated with this sditerator. That global history is returned by the history function.

Arguments

newHistory	in	The history string to append. Allowed:String

Returns
Bool

---

sditerator.appendrec - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Put an sdrecord at the end of the sditerator.

Synopsis
appendrec(rec)

Description
Appends this sdrecord to the end of the sditerator. The length of the sditerator is increased by 1 as a result of this operation. The sditerator must be writable.

Arguments

rec	in	The sdrecord to append to this sditerator. Allowed:sdrecord

Returns
Bool

---

sditerator.correcteddata - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Is the CORRECTED_DATA column being used?.

Synopsis
correcteddata()

Description
Determine if the CORRECTED_DATA or FLOAT_DATA column is being used. When the return value is T, the CORRECTED_DATA are being copied to the **arr** field. Otherwise, the FLOAT_DATA values are being copied to the **arr** field. This only works with MeaurementSet based data sets.

Returns
Bool

---

sditerator.deepcopy - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Make a true copy of this data set.

Synopsis
deepcopy(newname)

Description
This functions copies the underlying table to a new table on disk. The select function only returns a reference to the selected portion of the data set (no copy is made). This function can be used to make a copy of that selected portion of the data set. An sditerator `tool` will then need to be constructed using this new data set.

Example

```
# sdit is an already existing data set
sdit.deepcopy("newCopy");
copyOfSdit := sditerator("newCopy");
```

Arguments

newname	in	The name of the disk file to hold the new copy Allowed:String

Returns
Bool

---

sditerator.deleterec - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Delete the sdrecord at the current location pointer.

Synopsis
deleterec()

Description
The sdrecord at the current location pointer is removed from the data set. The length of the sditerator is descreased by one. The sditerator must be writable.

Returns
Bool

---

sditerator.done - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Close this sditerator and disconnect it from the data set.

Synopsis
done()

Description
When invoked, any unwritten data is flushed to disk, the connection with the underlying data set is closed and the `tool` is left unusable.

Returns
Bool

---

sditerator.get - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Get an sdrecord.

Synopsis

get()

Description
Get the sdrecord at the location of the current location pointer.

Returns
sdrecord

---

sditerator.getempty - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Get an empty sdrecord.

Synopsis
getempty(rec, nchan, nstokes)

Description
Get an empty sdrecord with default values for all of the fields and the shapes of the array fields consistent with the indicated shape.

Arguments

rec	out	the requested empty sdrecord Allowed:sdrecord
nchan	in	The number of channels Allowed:Integer
nstokes	in	The number of pixels along the stokes axis Allowed:Integer

Returns
Bool

---

sditerator.getdata - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Get the data part of the sdrecord.

Synopsis
getdata()

Description
Get the **data** part of the sdrecord at the location of the current location pointer. If all you want is information from the **data** record, this may be faster than getting the full record and picking out that information.

Returns
sdrecord

---

sditerator.getdesc - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Get the desc part of the field of the sdrecord.

Synopsis
getdesc()

Description
Get the **desc** part of the **data** field of the sdrecord at the location of the current location pointer. If all you want is information from the **desc** record, this may be faster than getting the full record and picking out that information.

Returns
sdrecord

---

sditerator.getdescvector - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Get a vector of values from the desc record.

Synopsis
getdescvector(fieldNames)

Description
This allows you to get all of the values for any scalar fields in the **desc** portion of an sdrecord in the sditerator with a single function call. This uses the getvectors function but is simpler to use if all of the fields are from the desc record. Here, you only give a vector of field names to extract from the sditerator. The returned value is a record where the field names are the requested fields.

Arguments

fieldNames	in	A vector of desc fields to fetch. Allowed:string vector

Returns
record

---

sditerator.getheader - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Get the header part of the sdrecord.

Synopsis
getheader()

Description
Get the **header** part of the sdrecord at the location of the current location pointer. If all you want is information from the **header** record, this may be faster than getting the full record and picking out that information.

Returns
sdrecord

sditerator.getheadervector - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Get a vector of values from the header record.

Synopsis
getheadervector(fieldNames)

Description
This allows you to get all of the values for any scalar fields in the **header** portion of an sdrecord in the sditerator with a single function call. This uses the getvectors function but is simpler to use if all of fields are from the **header** record. Here, you only give a vector of field names to extract from the sditerator. The returned value is a record where the field names are the requested fields.

Example

```
vec := sdit.getheadervectors("scan_number source_name");
scans := vec.scan_number;
sources := vec.source_name;
```

Arguments

fieldNames	in	A vector of header fields to fetch. Allowed:string vector

Returns
record

sditerator.getother - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Get the other part of the sdrecord.

Synopsis
getother()

Description
Get the **other** part of the sdrecord at the location of the current location pointer. If all you want is information from the **other** record, this may be faster than getting the full record and picking out that information.

Returns
sdrecord

---

sditerator.getvectors - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Get a vector of values for the fields in the template.

Synopsis
getvectors(template)

Description
This allows you to get all of the values for any scalar fields in an sditerator with a single function call. The template has the same structure as an sdrecord except that you only supply those fields that you want extracted. The returned value has the same structure as the template, but the field values are filled in with a vector containing each value from each sdrecord in the sditerator, in order. This function is tuned for a few fields so that they can be retrieved relatively quickly if they are the only fields in the template. These tuned fields are those used in the dish selection GUI: **source_name**, **scan_number**, **time**, **chan_width**, and **restfrequency**.

Example

```
# tediously extract the individual values
```

```
sdit.origin();
nrecords := sdit.length();
scans := array(0,nrecords);
sources := array("",nrecords);
restfqs := array(0.0,nrecords);
count := 1;
rec := sdit.get();
scans[count] := rec.header.scan_number;
sources[count] := rec.header.source_name;
restfqs[count] := rec.data.desc.restfrequency;
while (sdit.more()) {
   sdit.next();
   rec := sdit.get()
   count +:= 1
   scans[count] := rec.header.scan_number;
   sources[count] := rec.header.source_name;
   restfqs[count] := rec.data.desc.restfrequency;
}

# or alternatively using getvectors
template := [data=[desc=[restfrequency=0.0]],
             header=[scan_number=0,source_name=""]];
vec := sdit.getvectors(template);
scans := vec.header.scan_number;
sources := vec.header.source_name;
restfqs := vec.data.desc.restfrequency;
```

Arguments

template	in	The template indicating which fields to fetch. Allowed:record

Returns
record

---

sditerator.history - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Get the global history associated with all sdrecords

Synopsis
history()

Description
This history describes the creation of this sditerator and any selection and other global operations done on this sditerator. This history is appended to the sdrecord **hist** field when each sdrecord is fetched.

Returns
Bool

---

sditerator.id - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Return the ID record of this sditerator.

Synopsis
id()

Description
Used by the sdaverager tool to determine which sditerator is being used.

Returns
Record

---

sditerator.iswritable - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Can sdrecords be stored (written) to this sditerator?

Synopsis

iswritable()

Description
This returns the value of the readonly argument used at construction for this sditerator. Currently, only flat-table based sditerators can be written to.

Returns
Bool

---

sditerator.length - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Get the total number of sdrecords in this sditerator.

Synopsis
length()

Description
Returns the total number of sdrecords that are found in this sditerator. Note that for the flat-table data set, this may require examining each row of the table to see if there is more than one sdrecord stored in that row hence this may be an expensive operation for that type of table. Using more and next is always a faster way to step through (iterate) an sditerator (rather than using length and setlocation in a for loop).
ahreturnsInteger

---

sditerator.location - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Get the value of the location pointer.

Synopsis

location()

Returns
Integer

---

sditerator.more - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Are there more sdrecords after this location?

Synopsis
more()

Description
This function returns T if the location pointer is pointing at the last sdrecord in this sditerator.

Returns
Bool

---

sditerator.name - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

What is the name of the underlying data on disk.

Synopsis
name()

Returns
String

---

sditerator.next - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Set the location pointer to the next record.

Synopsis
next()

Description
This function advances the location pointer by one unless the pointer is already at the end. A subsequent use of the get function would return the sdrecord at that location.

Returns
Bool

---

sditerator.unlock - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Release the lock on all associated tables.

Synopsis
unlock()

Description
This function releases any table locks that may exist on the tables that are associated with this sditerator. Normally this isn't necessary but if you are trying to work directly with the associated tables through a table tool using unlock and lock should make the interaction happen faster since the the auto-locking mechanism won't need to periodically check to see if a lock needs to be released.

Returns

Bool

---

sditerator.lock - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Set the appropriate lock on all associated tables.

Synopsis
lock(nattempts)

Description
This function set table locks on the tables that are associated with this sditerator. Normally this isn't necessary but if you are trying to work directly with the associated tables through a table tool using unlock and lock should make the interaction happen faster since the the auto-locking mechanism won't need to periodically check to see if a lock needs to be set. The nattempts argument is used to indicate how many additional attempts to acquire the locks the tool should make before giving up. Each additional attempt happens after about a 1 second pause. If nattempts is 0, only the initial attempt will be made. If the locks can't be set, the return value will be false (F).

Arguments

nattempts	in	The number of times to attempt to set the lock. Allowed:Integer

Returns
Bool

---

sditerator.origin - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Set the location pointer to the first record.

Synopsis
origin()

Description
This function resets the location pointer to the first record in the sditerator. A subsequent use of the get function would return the sdrecord at that location.

Returns
Bool

---

sditerator.previous - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Set the location pointer to the previous record.

Synopsis
previous()

Description
This function decrements the location pointer by one unless the pointer is already at the origin. A subsequent use of the get function would return the sdrecord at that location.

Returns
Bool

---

sditerator.put - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Put an sdrecord at the current location pointer.

Synopsis
put(rec)

Description
Replaces the sdrecord at the current location pointer with this sdrecord. The sditerator must be writable.

Arguments

rec	in	The replacement sdrecord Allowed:sdrecord

Returns
Bool

---

sditerator.reselect - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Re-apply the previous selection to the parent data set.

Synopsis
reselect()

Description
This function re-applies the previous selection record to the parent data set of this iterator. If there is no previous selection, this function does nothing. This function is useful if the data set has changed since the last selection.

Returns
Bool

---

sditerator.resync - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Resynchronize with the data on disk.

Synopsis
resync()

Description
This function resynchronizes the internals of the sditerator to the data on disk. The default locking mode does that automatically and so this normally will not need to be used. Some locking modes do not do that. In that case, another process (e.g. an on-line filler) can write to the data on disk. In that case, the sditerator will not see those changes until this function is called.

Returns
Bool

---

sditerator.flush - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Flush the contents to disk.

Synopsis
flush()

Description
This function flushes the internals of the sditerator to the data on disk. The default locking mode does that automatically and so this normally will not need to be used. Some locking modes do not do that. In that case, another process (e.g. an the table client) may read data written through the sditerator. In that case, the data in the sditerator may not be available to that other process until it has been explicitly flushed to disk.

Returns
Bool

---

sditerator.select - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Make a new sditerator by applying a selection criteria record

Synopsis
select(selection)

Description
A new sditerator is constructed by applying the indicated selection criteria record to this sditerator. The returned sditerator is an sditerator which references this sditerator. Only sdrecords which match the criteria are found in the returned sditerator. The returned sditerator may be empty. The structure of the selection criteria record follows the structure of an sdrecord. Only fields which are to be selected on need be included in the selection criteria record. Only scalar fields (strings and numeric fields) in the sdrecord can be part of the criteria. Fields which are measures or quanta records in the sdrecord can be selected if they are single valued. The values used in the selection record should be in the same units as the first sdrecord in this sditerator. For string fields, the selection criteria value for that field should be a vector of values to match during the selection. A simple wildcard can be indicated using an asterisk. For numeric fields, the selection criteria should be a 2-dimensional array of values having dimensions (2, nranges). In other words, this is a set of range pairs. Values in this sditerator for that field which fall within that range or are equal to the range endpoints are considered to match this criteria.

Arguments

selection	in	Selection criteria record Allowed: *Glish* record Default: [=]

Returns
sditerator `tool`

Example

```
sdit := sditerator('a302_SP');
sel := [header=[scan_number=array(0,2,3),source_name="UGC1*"]];
sel.header.scan_number[,1] := [-1, 10];
sel.header.scan_number[,2] := [15,15];
sel.header.scan_number[,3] := [50,9999999];
s2 := sdit.select(sel);
```

In this example, scan numbers which are less than or equal to 10 or equal to 15 or greater than or equal to 50 are selected, so long as the source name matches the one given in the example (a wild card is used here).

---

sditerator.sel2string - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Convert a selection record to a string.

Synopsis
sel2string(selrec)

Description
This function converts a selection record to a string so that it can be stored in the history record associate with that iterator. This is used by the constructors and will likely not be used by the average user.

Arguments

selrec	in	Selection criteria record Allowed: *Glish* record

Returns
String

---

sditerator.setlocation - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Set the location pointer to the indicated record.

Synopsis
setlocation(location)

Description
This function sets the location pointer to the indicated record number. Valid record numbers run from 1 to the length of the sditerator. If the location argument is outside this range, the location pointer is set to the appropriate end point. A subsequent use of the getfunction would return the sdrecord at that location.

Arguments

location	in	The desired record number. Allowed:Integer

Returns
Bool

---

sditerator.stringfields - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Returns a record indicating which fields in the sditerator are strings.

Synopsis
stringfields()

Description
This is used in various places as a way of ensuring that the dish selection mechanism does the appropriate thing.

Returns
Record

---

sditerator.type - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Return the type of the underlying data.

Synopsis
type()

Description
Returns either "MeasurementSet" or "Table" to indicate the type of data underlying this sditerator.

Returns
String

---

sditerator.usecorrecteddata - Function

Package	**dish**
Module	**dish**
Tool	**sditerator**

Use the CORRECTED_DATA column?.

Synopsis
usecorrecteddata(correcteddata)

Description
Toggle the use of the CORRECTED_DATA or FLOAT_DATA column. When T, the CORRECTED_DATA are copied to the **arr** field. Otherwise, the FLOAT_DATA values are copied to the **arr** field. This only works with MeaurementSet-based data sets.

Arguments

correcteddata	in	Use the CORRECTED_DATA column? Allowed:Bool