BROWSER_LABEL , what label to use for the column,

BROWSER_FORMAT , how to format the members of that column (uses C syntax, currently not implemented), and

BROWSER_HIDE , whether to display that column or not

Hide removes the column from the display canvas,

Format selects how to format the column, uses C syntax.

Rows and Individual Cells may be selected by pressing a mouse button on the text. Vectors may be displayed as "[1 2 3]" or "[3]integer" depending on how many elements of the vector are chosen to display. Selecting on the "[3]integer" form will cause an arraybrowser to be displayed with the values of the vector or array displayed in it.

See Also
table tool

1.7.23 tablecomparer - Tool

Package	**utility**
Module	**table**

Compare various aspects of two tables to see if they are equivalent

Description

A `tablecomparer` tool compares various aspects of two tables to see if they are equivalent. Currently, attributes that can be compared include column names, column keyword names, column values, and table keyword names. When checking the values of the column data, a tolerance can be specified on a per column basis and/or a per column type (i.e., float, double, complex, or dcomplex) basis. Currently only absolute tolerances can be specified. The user can also specify whether tests should be run on subtables as well as the main tables.

Constructors

tablecomparer	Construct a tablecomparer tool

Functions

checkcoldata	Compare column data
checkcolkeywordnames	Compare column keyword names
checkcolnames	Check column names

checkkeywordnames	Compare table keyword names
compare	Compare tables by running one or more tests
details	Get details regarding failed tests
select	Specify which columns to check (or not to check)
settolerance	Set tolerances when comparing column data
summary	Summarize the comparisons
tolerance	Get the tolerance of the specified column

tablecomparer - Constructor

Package	**utility**
Module	**table**
Tool	**tablecomparer**

Construct a tablecomparer tool

Synopsis
tablecomparer(test, model, label, verbose)

Description
Create a `tablecomparer` tool.

Arguments

test	Name of the test table Allowed:Any valid table name Default: None
model	Name of the model (fiducial) table to compare the test table against Allowed:Any valid table name Default: None
label	Label for the (top level) table Allowed:String Default: 'TABLE'
verbose	Controls the volume of output, 0 no log messages, higher numbers mean more Allowed:Int Default: 1

Returns
tablecomparer tool

Example

```
# create a tablecomparer tool, and specify you want a lot of output
- tc := tablecomparer('test.table','fiducial.table',verbose=2);
```

---

tablecomparer.checkcolnames - Function

Package	**utility**
Module	**table**
Tool	**tablecomparer**

Check column names

Synopsis
checkcolnames(checksubtables)

Description
Check column names to see if they are identical between the two tables. Note that all column names will be checked regardless of any column selection made previously with `select()`.

Arguments

checksubtables	Check column names of subtables as well? Allowed:Boolean Default: T

Returns
T if the column names between the two tables are the same, F otherwise

Example

```
# create a tablecomparer tool, and specify you want a lot of output
- tc := tablecomparer('test.table','fiducial.table',verbose=2);
- tc.checkcolnames()
NORMAL: Column name test passed on table TABLE
Opening  CAL_DESC
NORMAL: successful readonly open of default-locked table
```

```
test.table/CAL_DESC: 12 columns, 1 rows
NORMAL: successful readonly open of default-locked table
fiducial.table/CAL_DESC: 12 columns, 1 rows
NORMAL: Column name test passed on table CAL_DESC
Opening  CAL_HISTORY
NORMAL: successful readonly open of default-locked table
test.table/CAL_HISTORY: 4 columns, 0 rows
NORMAL: successful readonly open of default-locked table
fiducial.table/CAL_HISTORY: 4 columns, 0 rows
NORMAL: Column name test passed on table CAL_HISTORY
T
```

Or, in the case where the test fails:

```
- t := tablecomparer('badtest.table','fiducial.table',verbose=2)
NORMAL: successful readonly open of default-locked table
badtest.table: 36 columns, 210 rows
NORMAL: successful readonly open of default-locked table
fiducial.table: 37 columns, 210 rows
- t.checkcolnames()
SEVERE: Column name test failed on table TABLE
SEVERE: TEST TABLE COLS: TIME TIME_EXTRA_PREC INTERVAL ANTENNA1 FEED1
FIELD_ID ARRAY_ID OBSERVATION_ID SCAN_NUMBER PROCESSOR_ID STATE_ID
PHASE_ID PULSAR_BIN PULSAR_GATE_ID FREQ_GROUP FREQ_GROUP_NAME
FIELD_NAME FIELD_CODE SOURCE_NAME SOURCE_CODE CALIBRATION_GROUP
REF_ANT REF_FEED REF_RECEPTOR REF_FREQUENCY MEAS_FREQ_REF
REF_DIRECTION MEAS_DIR_REF CAL_DESC_ID CAL_HISTORY_ID
TOTAL_SOLUTION_OK TOTAL_FIT TOTAL_FIT_WEIGHT SOLUTION_OK FIT
FIT_WEIGHT
SEVERE: MODEL TABLE COLS: TIME TIME_EXTRA_PREC INTERVAL ANTENNA1 FEED1
FIELD_ID ARRAY_ID OBSERVATION_ID SCAN_NUMBER PROCESSOR_ID STATE_ID
PHASE_ID PULSAR_BIN PULSAR_GATE_ID FREQ_GROUP FREQ_GROUP_NAME
FIELD_NAME FIELD_CODE SOURCE_NAME SOURCE_CODE CALIBRATION_GROUP GAIN
REF_ANT REF_FEED REF_RECEPTOR REF_FREQUENCY MEAS_FREQ_REF
REF_DIRECTION MEAS_DIR_REF CAL_DESC_ID CAL_HISTORY_ID
TOTAL_SOLUTION_OK TOTAL_FIT TOTAL_FIT_WEIGHT SOLUTION_OK FIT
FIT_WEIGHT
Opening  CAL_DESC
NORMAL: successful readonly open of default-locked table
badtest.table/CAL_DESC: 12 columns, 1 rows
NORMAL: successful readonly open of default-locked table
fiducial.table/CAL_DESC: 12 columns, 1 rows
NORMAL: Column name test passed on table CAL_DESC
Opening  CAL_HISTORY
NORMAL: successful readonly open of default-locked table
badtest.table/CAL_HISTORY: 4 columns, 0 rows
NORMAL: successful readonly open of default-locked table
```

```
fiducial.table/CAL_HISTORY: 4 columns, 0 rows
NORMAL: Column name test passed on table CAL_HISTORY
F

\vspace{.18in}
\hrule


   \subsubsection{tablecomparer.checkcolkeywordnames - Function\label{table:tablecomparer.checkcolkeywordnames.function}\index{table.tablecomparer.checkcolkeywordnames}\index{checkcolkeywordnames}}
   \subsectionM{tablecomparer.checkcolkeywordnames}{utility}{table}{tablecomparer}
   Compare column keyword names\\
   \flushleft\vspace{.20in}\textbf {Synopsis}\newline
   checkcolkeywordnames(checksubtables)

\flushleft\vspace{.20in}\textbf {Description}\newline
This function compares the keyword names of all columns not excluded
by the {\tt select()} function.  If all column keyword names match
between the two tables, it returns {\tt T}, if not, it returns {\tt F}.

\flushleft\vspace{.20in}\textbf {Arguments}\\
   \hfill \\
   \starthelptab
   \hline

   checksubtables & & \multicolumn{2}{p{3.36in}|}{Check column names of subtables as well?}\\
   & & Allowed: & Boolean\\
   & & Default: & T  \\ \hline
\end{tabular}
\flushleft\vspace{.20in}\textbf {Returns}\newline
   T if the tests pass, F otherwise\\
\flushleft\vspace{.20in}\textbf {Example}\newline
\begin{verbatim}
- tc := tablecomparer('test.table','fiducial.table',verbose=2)
NORMAL: successful readonly open of default-locked table test.table: 36 columns, 210 rows
NORMAL: successful readonly open of default-locked table fiducial.table: 37 columns, 210 rows
- tc.checkcolkeywordnames()
NORMAL: Column keyword name test passed on table TABLE column TIME
NORMAL: Column keyword name test passed on table TABLE column TIME_EXTRA_PREC
NORMAL: Column keyword name test passed on table TABLE column INTERVAL
NORMAL: Column keyword name test passed on table TABLE column ANTENNA1
NORMAL: Column keyword name test passed on table TABLE column FEED1
NORMAL: Column keyword name test passed on table TABLE column FIELD_ID
NORMAL: Column keyword name test passed on table TABLE column ARRAY_ID
NORMAL: Column keyword name test passed on table TABLE column OBSERVATION_ID
NORMAL: Column keyword name test passed on table TABLE column SCAN_NUMBER
NORMAL: Column keyword name test passed on table TABLE column PROCESSOR_ID
```

```
NORMAL: Column keyword name test passed on table TABLE column STATE_ID
NORMAL: Column keyword name test passed on table TABLE column PHASE_ID
NORMAL: Column keyword name test passed on table TABLE column PULSAR_BIN
NORMAL: Column keyword name test passed on table TABLE column PULSAR_GATE_ID
NORMAL: Column keyword name test passed on table TABLE column FREQ_GROUP
NORMAL: Column keyword name test passed on table TABLE column FREQ_GROUP_NAME
NORMAL: Column keyword name test passed on table TABLE column FIELD_NAME
NORMAL: Column keyword name test passed on table TABLE column FIELD_CODE
NORMAL: Column keyword name test passed on table TABLE column SOURCE_NAME
NORMAL: Column keyword name test passed on table TABLE column SOURCE_CODE
NORMAL: Column keyword name test passed on table TABLE column CALIBRATION_GROUP
NORMAL: Column keyword name test passed on table TABLE column REF_ANT
NORMAL: Column keyword name test passed on table TABLE column REF_FEED
NORMAL: Column keyword name test passed on table TABLE column REF_RECEPTOR
NORMAL: Column keyword name test passed on table TABLE column REF_FREQUENCY
NORMAL: Column keyword name test passed on table TABLE column MEAS_FREQ_REF
NORMAL: Column keyword name test passed on table TABLE column REF_DIRECTION
NORMAL: Column keyword name test passed on table TABLE column MEAS_DIR_REF
NORMAL: Column keyword name test passed on table TABLE column CAL_DESC_ID
NORMAL: Column keyword name test passed on table TABLE column CAL_HISTORY_ID
NORMAL: Column keyword name test passed on table TABLE column TOTAL_SOLUTION_OK
NORMAL: Column keyword name test passed on table TABLE column TOTAL_FIT
NORMAL: Column keyword name test passed on table TABLE column TOTAL_FIT_WEIGHT
NORMAL: Column keyword name test passed on table TABLE column SOLUTION_OK
NORMAL: Column keyword name test passed on table TABLE column FIT
NORMAL: Column keyword name test passed on table TABLE column FIT_WEIGHT
NORMAL: successful readonly open of default-locked table test.table/CAL_DESC: 12 columns, 1 rows
NORMAL: successful readonly open of default-locked table fiducial.table/CAL_DESC: 12 columns, 1 rows
NORMAL: Column keyword name test passed on table CAL_DESC column NUM_SPW
NORMAL: Column keyword name test passed on table CAL_DESC column NUM_CHAN
NORMAL: Column keyword name test passed on table CAL_DESC column NUM_RECEPTORS
NORMAL: Column keyword name test passed on table CAL_DESC column N_JONES
NORMAL: Column keyword name test passed on table CAL_DESC column SPECTRAL_WINDOW_ID
NORMAL: Column keyword name test passed on table CAL_DESC column CHAN_FREQ
NORMAL: Column keyword name test passed on table CAL_DESC column MEAS_FREQ_REF
NORMAL: Column keyword name test passed on table CAL_DESC column CHAN_WIDTH
NORMAL: Column keyword name test passed on table CAL_DESC column CHAN_RANGE
NORMAL: Column keyword name test passed on table CAL_DESC column POLARIZATION_TYPE
NORMAL: Column keyword name test passed on table CAL_DESC column JONES_TYPE
NORMAL: Column keyword name test passed on table CAL_DESC column MS_NAME
NORMAL: successful readonly open of default-locked table test.table/CAL_HISTORY: 4 columns, 0 rows
NORMAL: successful readonly open of default-locked table fiducial.table/CAL_HISTORY: 4 columns, 0 rows
NORMAL: Column keyword name test passed on table CAL_HISTORY column CAL_PARMS
NORMAL: Column keyword name test passed on table CAL_HISTORY column CAL_TABLES
NORMAL: Column keyword name test passed on table CAL_HISTORY column CAL_SELECT
NORMAL: Column keyword name test passed on table CAL_HISTORY column CAL_NOTES
```

T

---

tablecomparer.checkcoldata - Function

Package	**utility**
Module	**table**
Tool	**tablecomparer**

Compare column data

Synopsis
checkcoldata(checksubtables)

Description
This function compares the data of all columns not excluded by the `select()` function. Tolerances on a per column and/or per column type basis basis can be specified by running `setprecision()` before running this function. This function will check both regular and variable array column data, in the latter case doing a record by record test for equality. It assumes that unset values are equal. If all column data match (to within the specified precisions, if any) between the two tables, the function returns `T`, if not, it returns `F`.

Arguments

checksubtables	Check column names of subtables as well? Allowed:Boolean Default: T

Returns
T if the tests pass, F otherwise

Example

```
- tc := tablecomparer('test.table','fiducial.table',verbose=2)
NORMAL: successful readonly open of default-locked table test.table: 37 columns, 210 rows
NORMAL: successful readonly open of default-locked table fiducial.table: 37 columns, 210 rows
- tc.checkcoldata()
NORMAL: Column data test passed on table TABLE column TIME
```

```
NORMAL: Column data test passed on table TABLE column TIME_EXTRA_PREC
NORMAL: Column data test passed on table TABLE column INTERVAL
NORMAL: Column data test passed on table TABLE column ANTENNA1
NORMAL: Column data test passed on table TABLE column FEED1
NORMAL: Column data test passed on table TABLE column FIELD_ID
NORMAL: Column data test passed on table TABLE column ARRAY_ID
NORMAL: Column data test passed on table TABLE column OBSERVATION_ID
NORMAL: Column data test passed on table TABLE column SCAN_NUMBER
NORMAL: Column data test passed on table TABLE column PROCESSOR_ID
NORMAL: Column data test passed on table TABLE column STATE_ID
NORMAL: Column data test passed on table TABLE column PHASE_ID
NORMAL: Column data test passed on table TABLE column PULSAR_BIN
NORMAL: Column data test passed on table TABLE column PULSAR_GATE_ID
NORMAL: Column data test passed on table TABLE column FREQ_GROUP
NORMAL: Column data test passed on table TABLE column FREQ_GROUP_NAME
NORMAL: Column data test passed on table TABLE column FIELD_NAME
NORMAL: Column data test passed on table TABLE column FIELD_CODE
NORMAL: Column data test passed on table TABLE column SOURCE_NAME
NORMAL: Column data test passed on table TABLE column SOURCE_CODE
NORMAL: Column data test passed on table TABLE column CALIBRATION_GROUP
NORMAL: Column data test passed on table TABLE column GAIN
NORMAL: Column data test passed on table TABLE column REF_ANT
NORMAL: Column data test passed on table TABLE column REF_FEED
NORMAL: Column data test passed on table TABLE column REF_RECEPTOR
NORMAL: Column data test passed on table TABLE column REF_FREQUENCY
NORMAL: Column data test passed on table TABLE column MEAS_FREQ_REF
NORMAL: Column data test passed on table TABLE column REF_DIRECTION
NORMAL: Column data test passed on table TABLE column MEAS_DIR_REF
NORMAL: Column data test passed on table TABLE column CAL_DESC_ID
NORMAL: Column data test passed on table TABLE column CAL_HISTORY_ID
NORMAL: Column data test passed on table TABLE column TOTAL_SOLUTION_OK
NORMAL: Column data test passed on table TABLE column TOTAL_FIT
NORMAL: Column data test passed on table TABLE column TOTAL_FIT_WEIGHT
NORMAL: Column data test passed on table TABLE column SOLUTION_OK
NORMAL: Column data test passed on table TABLE column FIT
NORMAL: Column data test passed on table TABLE column FIT_WEIGHT
NORMAL: successful readonly open of default-locked table test.table/CAL_DESC: 12 columns, 1 rows
NORMAL: successful readonly open of default-locked table fiducial.table/CAL_DESC: 12 columns, 1 rows
NORMAL: Column data test passed on table CAL_DESC column NUM_SPW
NORMAL: Column data test passed on table CAL_DESC column NUM_CHAN
NORMAL: Column data test passed on table CAL_DESC column NUM_RECEPTORS
NORMAL: Column data test passed on table CAL_DESC column N_JONES
NORMAL: Column data test passed on table CAL_DESC column SPECTRAL_WINDOW_ID
NORMAL: Column data test passed on table CAL_DESC column CHAN_FREQ
NORMAL: Column data test passed on table CAL_DESC column MEAS_FREQ_REF
NORMAL: Column data test passed on table CAL_DESC column CHAN_WIDTH
```

```
NORMAL: Column data test passed on table CAL_DESC column CHAN_RANGE
NORMAL: Column data test passed on table CAL_DESC column POLARIZATION_TYPE
NORMAL: Column data test passed on table CAL_DESC column JONES_TYPE
NORMAL: Column data test passed on table CAL_DESC column MS_NAME
NORMAL: successful readonly open of default-locked table test.table/CAL_HISTORY: 4 columns, 0 rows
NORMAL: successful readonly open of default-locked table fiducial.table/CAL_HISTORY: 4 columns, 0 rows
NORMAL: Column data test passed on table CAL_HISTORY column CAL_PARMS
NORMAL: Column data test passed on table CAL_HISTORY column CAL__TABLES
NORMAL: Column data test passed on table CAL_HISTORY column CAL_SELECT
NORMAL: Column data test passed on table CAL_HISTORY column CAL_NOTES
T
```

---

tablecomparer.checkkeywordnames - Function

Package	**utility**
Module	**table**
Tool	**tablecomparer**

Compare table keyword names

Synopsis
checkkeywordnames(checksubtables)

Description
This function compares table keyword names. If the keyword names in one table match those in the other, it returns `T`.

Arguments

checksubtables	Check column names of subtables as well? Allowed:Boolean Default: T

Returns
T if the tests pass, F otherwise

Example

```
- tc := tablecomparer('test.table','fiducial.table',verbose=2)
NORMAL: successful readonly open of default-locked table test.table:
37 columns, 210 rows
NORMAL: successful readonly open of default-locked table fiducial.table: 37 columns, 210 rows
- tc.checkkeywordnames()
NORMAL: Keyword field names test passed for table TABLE
NORMAL: successful readonly open of default-locked table test.table/CAL_DESC: 12 columns, 1 rows
NORMAL: successful readonly open of default-locked table fiducial.table/CAL_DESC: 12 columns, 1 rows
NORMAL: Keyword field names test passed for table CAL_DESC
NORMAL: successful readonly open of default-locked table test.table/CAL_HISTORY: 4 columns, 0 rows
NORMAL: successful readonly open of default-locked table fiducial.table/CAL_HISTORY: 4 columns, 0 rows
NORMAL: Keyword field names test passed for table CAL_HISTORY
T
```

---

tablecomparer.settolerance - Function

Package	**utility**
Module	**table**
Tool	**tablecomparer**

Set tolerances when comparing column data

Synopsis
settolerance(float, double, complex, dcomplex, cols)

Description
This function sets tolerances that are used by `checkcoldata` when comparing (numerical) column data. Currently, only absolute (not fractional) tolerances are supported. Tolerances can be set according to the type of a column (`float, double, complex, dcomplex`) or they can be specified on a per column basis using the `cols` parameter. In the case where both the column and its type have been given a tolerance, the column specific tolerance is used. When in doubt, you can check what the tolerance of a specific column is by running the `tolerance()` function. In the case of complex numbers, the tolerance is a real number representing the radius about the value in question in the complex plane.
Here are a couple of examples. Suppose you have a table with a column of type double called `DCOL`. If you run `settolerance(double=1e-8)` then the acceptable tolerance when checking the values in this column will be 1e-8. The way to specify tolerances on a per column basis is to pass in a record with fields of the form `TABLELABEL.COLUMNNAME`, bearing in mind that the default value of `label` in the `tablecomparer()` constructor is `TABLE` and that

subtables of the main table are labeled by their names. So in the previous example, if you used the default value of `label` when constructing a tablecomparer tool, you could run

```
tc := tablecomparer('test.table','fiducial.table')
tol := [=];
tol.TABLE.DCOL := 1e-8;
tc.setprecision(cols := tol);
```

If you wanted to set the tolerance of `DCOL` in the main table to 1e-8 and the tolerance of `GCOL` of subtable `GTABLE` to be 1e-5, you could run

```
tc := tablecomparer('test.table','fiducial.table')
tol := [=];
tol.TABLE.DCOL := 1e-8;
tol.GTABLE.GCOL := 1e-5;
tc.setprecision(cols := tol);
```

Arguments

float	Default tolerance for all columns of type float Allowed:Float Default: 0
double	Default tolerance for all columns of type double Allowed:Double Default: 0
complex	Default tolerance for all columns of type complex Allowed:Float Default: 0
dcomplex	Default tolerance for all columns of type complex Allowed:Double Default: 0
cols	Column specific tolerances (see above for examples Allowed:Record containing real values Default: [=]

Returns
T

Example

```
- tc := tablecomparer('test.table','fiducial.table',verbose=3)
- # set the default tolerance of all columns of type double to 1e-8
- # all float columns to 1e-4, and set the tolerance of the GAIN
```

```
- # column in the main table to 1e-5 and the CHA_WIDTH column in the
- # CAL_DESC table to 1e-4
- tol := [=];
- tol.TABLE.GAIN := 1e-5;
- tol.CAL_DESC.CHAN_WIDTH := 1e-4;
- tc.settolerance(float=1e-5, double=1e-8, cols=tol);
T
# compare column data using these tolerances
- tc.checkcoldata();
NORMAL: Checking column TIME
NORMAL: Using tolerance 1e-08 on table TABLE column TIME
NORMAL: Column data test passed on table TABLE column TIME
NORMAL: Checking column TIME_EXTRA_PREC
NORMAL: Using tolerance 1e-08 on table TABLE column TIME_EXTRA_PREC
NORMAL: Column data test passed on table TABLE column TIME_EXTRA_PREC
NORMAL: Checking column INTERVAL
NORMAL: Using tolerance 1e-08 on table TABLE column INTERVAL
NORMAL: Column data test passed on table TABLE column INTERVAL
NORMAL: Checking column ANTENNA1
NORMAL: Column data test passed on table TABLE column ANTENNA1
NORMAL: Checking column FEED1
NORMAL: Column data test passed on table TABLE column FEED1
NORMAL: Checking column FIELD_ID
NORMAL: Column data test passed on table TABLE column FIELD_ID
NORMAL: Checking column ARRAY_ID
NORMAL: Column data test passed on table TABLE column ARRAY_ID
NORMAL: Checking column OBSERVATION_ID
NORMAL: Column data test passed on table TABLE column OBSERVATION_ID
NORMAL: Checking column SCAN_NUMBER
NORMAL: Column data test passed on table TABLE column SCAN_NUMBER
NORMAL: Checking column PROCESSOR_ID
NORMAL: Column data test passed on table TABLE column PROCESSOR_ID
NORMAL: Checking column STATE_ID
NORMAL: Column data test passed on table TABLE column STATE_ID
NORMAL: Checking column PHASE_ID
NORMAL: Column data test passed on table TABLE column PHASE_ID
NORMAL: Checking column PULSAR_BIN
NORMAL: Column data test passed on table TABLE column PULSAR_BIN
NORMAL: Checking column PULSAR_GATE_ID
NORMAL: Column data test passed on table TABLE column PULSAR_GATE_ID
NORMAL: Checking column FREQ_GROUP
NORMAL: Column data test passed on table TABLE column FREQ_GROUP
NORMAL: Checking column FREQ_GROUP_NAME
NORMAL: Column data test passed on table TABLE column FREQ_GROUP_NAME
NORMAL: Checking column FIELD_NAME
NORMAL: Column data test passed on table TABLE column FIELD_NAME
```

```
NORMAL: Checking column FIELD_CODE
NORMAL: Column data test passed on table TABLE column FIELD_CODE
NORMAL: Checking column SOURCE_NAME
NORMAL: Column data test passed on table TABLE column SOURCE_NAME
NORMAL: Checking column SOURCE_CODE
NORMAL: Column data test passed on table TABLE column SOURCE_CODE
NORMAL: Checking column CALIBRATION_GROUP
NORMAL: Column data test passed on table TABLE column CALIBRATION_GROUP
NORMAL: Checking column GAIN
NORMAL: Using tolerance 1e-05 on table TABLE column GAIN
NORMAL: Column data test passed on table TABLE column GAIN
NORMAL: Checking column REF_ANT
NORMAL: Column data test passed on table TABLE column REF_ANT
NORMAL: Checking column REF_FEED
NORMAL: Column data test passed on table TABLE column REF_FEED
NORMAL: Checking column REF_RECEPTOR
NORMAL: Column data test passed on table TABLE column REF_RECEPTOR
NORMAL: Checking column REF_FREQUENCY
NORMAL: Using tolerance 1e-08 on table TABLE column REF_FREQUENCY
NORMAL: Column data test passed on table TABLE column REF_FREQUENCY
NORMAL: Checking column MEAS_FREQ_REF
NORMAL: Column data test passed on table TABLE column MEAS_FREQ_REF
NORMAL: Checking column REF_DIRECTION
NORMAL: Using tolerance 1e-08 on table TABLE column REF_DIRECTION
NORMAL: Column data test passed on table TABLE column REF_DIRECTION
NORMAL: Checking column MEAS_DIR_REF
NORMAL: Column data test passed on table TABLE column MEAS_DIR_REF
NORMAL: Checking column CAL_DESC_ID
NORMAL: Column data test passed on table TABLE column CAL_DESC_ID
NORMAL: Checking column CAL_HISTORY_ID
NORMAL: Column data test passed on table TABLE column CAL_HISTORY_ID
NORMAL: Checking column TOTAL_SOLUTION_OK
NORMAL: Column data test passed on table TABLE column TOTAL_SOLUTION_OK
NORMAL: Checking column TOTAL_FIT
NORMAL: Using tolerance 1e-05 on table TABLE column TOTAL_FIT
NORMAL: Column data test passed on table TABLE column TOTAL_FIT
NORMAL: Checking column TOTAL_FIT_WEIGHT
NORMAL: Using tolerance 1e-05 on table TABLE column TOTAL_FIT_WEIGHT
NORMAL: Column data test passed on table TABLE column TOTAL_FIT_WEIGHT
NORMAL: Checking column SOLUTION_OK
NORMAL: Column data test passed on table TABLE column SOLUTION_OK
NORMAL: Checking column FIT
NORMAL: Using tolerance 1e-05 on table TABLE column FIT
NORMAL: Column data test passed on table TABLE column FIT
NORMAL: Checking column FIT_WEIGHT
NORMAL: Using tolerance 1e-05 on table TABLE column FIT_WEIGHT
```

```
NORMAL: Column data test passed on table TABLE column FIT_WEIGHT
NORMAL: successful readonly open of default-locked table test.table/CAL_DESC: 12 columns, 1 rows
NORMAL: successful readonly open of default-locked table fiducial.table/CAL_DESC: 12 columns, 1 rows
NORMAL: Checking column NUM_SPW
NORMAL: Column data test passed on table CAL_DESC column NUM_SPW
NORMAL: Checking column NUM_CHAN
NORMAL: Column data test passed on table CAL_DESC column NUM_CHAN
NORMAL: Checking column NUM_RECEPTORS
NORMAL: Column data test passed on table CAL_DESC column NUM_RECEPTORS
NORMAL: Checking column N_JONES
NORMAL: Column data test passed on table CAL_DESC column N_JONES
NORMAL: Checking column SPECTRAL_WINDOW_ID
NORMAL: Column data test passed on table CAL_DESC column SPECTRAL_WINDOW_ID
NORMAL: Checking column CHAN_FREQ
NORMAL: Using tolerance 1e-08 on table CAL_DESC column CHAN_FREQ
NORMAL: Column data test passed on table CAL_DESC column CHAN_FREQ
NORMAL: Checking column MEAS_FREQ_REF
NORMAL: Column data test passed on table CAL_DESC column MEAS_FREQ_REF
NORMAL: Checking column CHAN_WIDTH
NORMAL: Using tolerance 0.0001 on table CAL_DESC column CHAN_WIDTH
NORMAL: Column data test passed on table CAL_DESC column CHAN_WIDTH
NORMAL: Checking column CHAN_RANGE
NORMAL: Column data test passed on table CAL_DESC column CHAN_RANGE
NORMAL: Checking column POLARIZATION_TYPE
NORMAL: Column data test passed on table CAL_DESC column
POLARIZATION_TYPE
NORMAL: Checking column JONES_TYPE
NORMAL: Column data test passed on table CAL_DESC column JONES_TYPE
NORMAL: Checking column MS_NAME
NORMAL: Column data test passed on table CAL_DESC column MS_NAME
NORMAL: successful readonly open of default-locked table test.table/CAL_HISTORY: 4 columns, 0 rows
NORMAL: successful readonly open of default-locked table fiducial.table/CAL_HISTORY: 4 columns, 0 rows
NORMAL: Checking column CAL_PARMS
NORMAL: Column data test passed on table CAL_HISTORY column CAL_PARMS
NORMAL: Checking column CAL_TABLES
NORMAL: Column data test passed on table CAL_HISTORY column CAL_TABLES
NORMAL: Checking column CAL_SELECT
NORMAL: Column data test passed on table CAL_HISTORY column CAL_SELECT
NORMAL: Checking column CAL_NOTES
NORMAL: Column data test passed on table CAL_HISTORY column CAL_NOTES
T
```

---

tablecomparer.select - Function

Package utility

Module **table**
Tool **tablecomparer**

Specify which columns to check (or not to check)

Synopsis

select(checkonlycols, nocheckcols)

Description

This function allows the user to specify which columns to check or not to check when running `checkcolkeywordnames()` and `checkcoldata()`. The `checkonlycols` parameter indicates that these functions should check only the columns specified, while the `nocheckcols` parameter indicates that all columns except those specified should be checked. If both are specified, it is the same as specifying only the `checkonlycols` parameter. Each parameter can be either a vector of column names or a record of column names. In the case of a vector, a column of that name in either the main table or any of its subtables matches (assuming the `check*` function is run with `checksubtables=T`. In the case of a record being used for these parameters, the field names of the record are the table labels, and each field can be assigned a vector of column names. So for examaple to check only the TIME and GAIN column of the main table and the SPECTRAL_WINDOW_ID column of the CAL_DESC subtable you would do (bearing in mind that the defalt label for the top-level table is `TABLE`):

```
tc := tablecomparer('test.table','fiducial.table')
occ := [=];
occ.MAIN := ['TIME','GAIN'];
occ.CAL_DESC := 'SPECTRAL_WINDOW_ID'
tc.select(checkonlycols=occ)
tc.checkcoldata();
```

Arguments

checkonlycols	Check only these columns Allowed: Array or record Default: []
nocheckcols	Do not check these columns (not used if checkonlycols is specified) Allowed: Array or record Default: []

Returns

T

Example

```
- tc := tablecomparer('test.table','fiducial.table', verbose=2)
# check only the GAIN column of the main table
- occ := [=]
- occ.TABLE := 'GAIN';
- tc.select(occ);
- tc.checkcoldata();
NORMAL: Skipping column TIME of table TABLE because of selection.
NORMAL: Skipping column TIME_EXTRA_PREC of table TABLE because of selection.
NORMAL: Skipping column INTERVAL of table TABLE because of selection.
NORMAL: Skipping column ANTENNA1 of table TABLE because of selection.
NORMAL: Skipping column FEED1 of table TABLE because of selection.
NORMAL: Skipping column FIELD_ID of table TABLE because of selection.
NORMAL: Skipping column ARRAY_ID of table TABLE because of selection.
NORMAL: Skipping column OBSERVATION_ID of table TABLE because of selection.
NORMAL: Skipping column SCAN_NUMBER of table TABLE because of selection.
NORMAL: Skipping column PROCESSOR_ID of table TABLE because of selection.
NORMAL: Skipping column STATE_ID of table TABLE because of selection.
NORMAL: Skipping column PHASE_ID of table TABLE because of selection.
NORMAL: Skipping column PULSAR_BIN of table TABLE because of selection.
NORMAL: Skipping column PULSAR_GATE_ID of table TABLE because of selection.
NORMAL: Skipping column FREQ_GROUP of table TABLE because of selection.
NORMAL: Skipping column FREQ_GROUP_NAME of table TABLE because of selection.
NORMAL: Skipping column FIELD_NAME of table TABLE because of selection.
NORMAL: Skipping column FIELD_CODE of table TABLE because of selection.
NORMAL: Skipping column SOURCE_NAME of table TABLE because of selection.
NORMAL: Skipping column SOURCE_CODE of table TABLE because of selection.
NORMAL: Skipping column CALIBRATION_GROUP of table TABLE because of selection.
NORMAL: Column data test passed on table TABLE column GAIN
NORMAL: Skipping column REF_ANT of table TABLE because of selection.
NORMAL: Skipping column REF_FEED of table TABLE because of selection.
NORMAL: Skipping column REF_RECEPTOR of table TABLE because of selection.
NORMAL: Skipping column REF_FREQUENCY of table TABLE because of selection.
NORMAL: Skipping column MEAS_FREQ_REF of table TABLE because of selection.
NORMAL: Skipping column REF_DIRECTION of table TABLE because of selection.
NORMAL: Skipping column MEAS_DIR_REF of table TABLE because of selection.
NORMAL: Skipping column CAL_DESC_ID of table TABLE because of selection.
NORMAL: Skipping column CAL_HISTORY_ID of table TABLE because of selection.
NORMAL: Skipping column TOTAL_SOLUTION_OK of table TABLE because of selection.
NORMAL: Skipping column TOTAL_FIT of table TABLE because of selection.
NORMAL: Skipping column TOTAL_FIT_WEIGHT of table TABLE because of selection.
NORMAL: Skipping column SOLUTION_OK of table TABLE because of selection.
NORMAL: Skipping column FIT of table TABLE because of selection.
NORMAL: Skipping column FIT_WEIGHT of table TABLE because of selection.
```

```
NORMAL: successful readonly open of default-locked table test.table/CAL_DESC: 12 columns, 1 rows
NORMAL: successful readonly open of default-locked table fiducial.table/CAL_DESC: 12 columns, 1 rows
NORMAL: Skipping column NUM_SPW of table CAL_DESC because of selection.
NORMAL: Skipping column NUM_CHAN of table CAL_DESC because of selection.
NORMAL: Skipping column NUM_RECEPTORS of table CAL_DESC because of selection.
NORMAL: Skipping column N_JONES of table CAL_DESC because of selection.
NORMAL: Skipping column SPECTRAL_WINDOW_ID of table CAL_DESC because of selection.
NORMAL: Skipping column CHAN_FREQ of table CAL_DESC because of selection.
NORMAL: Skipping column MEAS_FREQ_REF of table CAL_DESC because of selection.
NORMAL: Skipping column CHAN_WIDTH of table CAL_DESC because of selection.
NORMAL: Skipping column CHAN_RANGE of table CAL_DESC because of selection.
NORMAL: Skipping column POLARIZATION_TYPE of table CAL_DESC because of selection.
NORMAL: Skipping column JONES_TYPE of table CAL_DESC because of selection.
NORMAL: Skipping column MS_NAME of table CAL_DESC because of selection.
NORMAL: successful readonly open of default-locked table test.table/CAL_HISTORY: 4 columns, 0 rows
NORMAL: successful readonly open of default-locked table fiducial.table/CAL_HISTORY: 4 columns, 0 rows
NORMAL: Skipping column CAL_PARMS of table CAL_HISTORY because of selection.
NORMAL: Skipping column CAL_TABLES of table CAL_HISTORY because of selection.
NORMAL: Skipping column CAL_SELECT of table CAL_HISTORY because of selection.
NORMAL: Skipping column CAL_NOTES of table CAL_HISTORY because of selection.
T
```

---

tablecomparer.tolerance - Function

Package	**utility**
Module	**table**
Tool	**tablecomparer**

Get the tolerance of the specified column

Synopsis
tolerance(col, tlabel)

Description
This function returns the tolerance of the specified column in the specified table as it was set by `settolerance()` (0 if no set tolerance applies to the specified column).

Arguments

col	The name of the column to get the tolerance of Allowed:A valid column name Default: None
tlabel	The label of the table or subtable in which this column is Allowed:A valid table label Default: '' (the top level table)

Returns
T

Example

```
- tc := tablecomparer('test.table','fiducial.table')
- tol := [=];
- tol.TABLE.DCOL := 1e-8;
- tol.GTABLE.GCOL := 1e-5;
- tc.setprecision(double=1e-9, cols=tol);
- tc.tolerance('DCOL')
1e-8
- tc.tolerance('GCOL','GTABLE')
1e-05
- # CCOL is a column of type double
- tc.tolerance('CCOL')
1e-9
- # HCOL is a column of type float
- tc.tolerance('HCOL')
0
```

---

tablecomparer.compare - Function

Package	**utility**
Module	**table**
Tool	**tablecomparer**

Compare tables by running one or more tests

Synopsis
compare(checksubtables, checkcolnames, checkcolkeywordnames, checkcoldata, checktablekeywordnames)

Description
This function runs one or more `check*` tests on the table. If all tests pass, it returns `T`, it returns `F` otherwise.

Arguments

checksubtables	Check subtables as well? Allowed:Boolean Default: T
checkcolnames	Run the column names test? Allowed:Boolean Default: T
checkcolkeywordnames	Run the column keyword names test? Allowed:Boolean Default: T
checkcoldata	Run the column data test? Allowed:Boolean Default: T
checktablekeywordnames	Run the table keyword names test? Allowed:Boolean Default: T

Returns
T if all specified tests pass, F otherwise

Example

```
- tc := tablecomparer('test.table','fiducial.table', verbose=2)
- # run all available tests
- tc.compare();
<lots of messages>
T
```

---

tablecomparer.details - Function

Package	**utility**
Module	**table**
Tool	**tablecomparer**

Get details regarding failed tests

Synopsis
details()

Description
This function returns a record containing the details of failed tests. Currently, this record contains information only column data tests which fail. In the case of homogenous array data, this record contains `test`, `model`, and `diff` fields. These fields themselves are records containing fields of the form `LABEL.COLUMN` which hold the actual column data of of the test and model and the difference of these columns, respectively. In the case of heterogeneous (variable) array columns, the diff record is absent. Thus, the `test.TABLE.DATA` field contains the DATA column of the main table of the test dataset on which the DATA column test failed.

Arguments

Returns
Record

Example

```
- tc := tablecomparer('test.table','fiducial.table', verbose=2)
- tc.compare();
- # the DATA column test of the IMAGE subtable failed. Stuff that
- # column into a glish array for inspection
- d := tc.details();
- tdata := d.test.IMAGE.DATA
```

---

tablecomparer.summary - Function

Package	**utility**
Module	**table**
Tool	**tablecomparer**

Summarize the comparisons

Synopsis
summary(dolog)

Description
This function returns a summary of the results of comparisons that have been run and optionally outputs them to the logger. A value of `T` means that a test passed, a value of `F` means that it failed.

Arguments

dolog	Write summary to logger? Allowed:Boolean Default: F

Returns
Record

Example

```
- tc := tablecomparer('test.table','fiducial.table', verbose=2)
- tc.compare();
- results := tc.summary(T);
NORMAL: TABLE.COLNAMES: T
NORMAL: TABLE.COLKEYWORDNAMES.TIME: T
NORMAL: TABLE.COLKEYWORDNAMES.TIME_EXTRA_PREC: T
NORMAL: TABLE.COLKEYWORDNAMES.INTERVAL: T
NORMAL: TABLE.COLKEYWORDNAMES.ANTENNA1: T
NORMAL: TABLE.COLKEYWORDNAMES.FEED1: T
NORMAL: TABLE.COLKEYWORDNAMES.FIELD_ID: T
NORMAL: TABLE.COLKEYWORDNAMES.ARRAY_ID: T
NORMAL: TABLE.COLKEYWORDNAMES.OBSERVATION_ID: T
NORMAL: TABLE.COLKEYWORDNAMES.SCAN_NUMBER: T
NORMAL: TABLE.COLKEYWORDNAMES.PROCESSOR_ID: T
NORMAL: TABLE.COLKEYWORDNAMES.STATE_ID: T
NORMAL: TABLE.COLKEYWORDNAMES.PHASE_ID: T
NORMAL: TABLE.COLKEYWORDNAMES.PULSAR_BIN: T
NORMAL: TABLE.COLKEYWORDNAMES.PULSAR_GATE_ID: T
NORMAL: TABLE.COLKEYWORDNAMES.FREQ_GROUP: T
NORMAL: TABLE.COLKEYWORDNAMES.FREQ_GROUP_NAME: T
NORMAL: TABLE.COLKEYWORDNAMES.FIELD_NAME: T
NORMAL: TABLE.COLKEYWORDNAMES.FIELD_CODE: T
NORMAL: TABLE.COLKEYWORDNAMES.SOURCE_NAME: T
```

```
NORMAL: TABLE.COLKEYWORDNAMES.SOURCE_CODE: T
NORMAL: TABLE.COLKEYWORDNAMES.CALIBRATION_GROUP: T
NORMAL: TABLE.COLKEYWORDNAMES.GAIN: T
NORMAL: TABLE.COLKEYWORDNAMES.REF_ANT: T
NORMAL: TABLE.COLKEYWORDNAMES.REF_FEED: T
NORMAL: TABLE.COLKEYWORDNAMES.REF_RECEPTOR: T
NORMAL: TABLE.COLKEYWORDNAMES.REF_FREQUENCY: T
NORMAL: TABLE.COLKEYWORDNAMES.MEAS_FREQ_REF: T
NORMAL: TABLE.COLKEYWORDNAMES.REF_DIRECTION: T
NORMAL: TABLE.COLKEYWORDNAMES.MEAS_DIR_REF: T
NORMAL: TABLE.COLKEYWORDNAMES.CAL_DESC_ID: T
NORMAL: TABLE.COLKEYWORDNAMES.CAL_HISTORY_ID: T
NORMAL: TABLE.COLKEYWORDNAMES.TOTAL_SOLUTION_OK: T
NORMAL: TABLE.COLKEYWORDNAMES.TOTAL_FIT: T
NORMAL: TABLE.COLKEYWORDNAMES.TOTAL_FIT_WEIGHT: T
NORMAL: TABLE.COLKEYWORDNAMES.SOLUTION_OK: T
NORMAL: TABLE.COLKEYWORDNAMES.FIT: T
NORMAL: TABLE.COLKEYWORDNAMES.FIT_WEIGHT: T
NORMAL: TABLE.COLVALUES.TIME: T
NORMAL: TABLE.COLVALUES.TIME_EXTRA_PREC: T
NORMAL: TABLE.COLVALUES.INTERVAL: T
NORMAL: TABLE.COLVALUES.ANTENNA1: T
NORMAL: TABLE.COLVALUES.FEED1: T
NORMAL: TABLE.COLVALUES.FIELD_ID: T
NORMAL: TABLE.COLVALUES.ARRAY_ID: T
NORMAL: TABLE.COLVALUES.OBSERVATION_ID: T
NORMAL: TABLE.COLVALUES.SCAN_NUMBER: T
NORMAL: TABLE.COLVALUES.PROCESSOR_ID: T
NORMAL: TABLE.COLVALUES.STATE_ID: T
NORMAL: TABLE.COLVALUES.PHASE_ID: T
NORMAL: TABLE.COLVALUES.PULSAR_BIN: T
NORMAL: TABLE.COLVALUES.PULSAR_GATE_ID: T
NORMAL: TABLE.COLVALUES.FREQ_GROUP: T
NORMAL: TABLE.COLVALUES.FREQ_GROUP_NAME: T
NORMAL: TABLE.COLVALUES.FIELD_NAME: T
NORMAL: TABLE.COLVALUES.FIELD_CODE: T
NORMAL: TABLE.COLVALUES.SOURCE_NAME: T
NORMAL: TABLE.COLVALUES.SOURCE_CODE: T
NORMAL: TABLE.COLVALUES.CALIBRATION_GROUP: T
NORMAL: TABLE.COLVALUES.GAIN: T
NORMAL: TABLE.COLVALUES.REF_ANT: T
NORMAL: TABLE.COLVALUES.REF_FEED: T
NORMAL: TABLE.COLVALUES.REF_RECEPTOR: T
NORMAL: TABLE.COLVALUES.REF_FREQUENCY: T
NORMAL: TABLE.COLVALUES.MEAS_FREQ_REF: T
NORMAL: TABLE.COLVALUES.REF_DIRECTION: T
```

```
NORMAL: TABLE.COLVALUES.MEAS_DIR_REF: T
NORMAL: TABLE.COLVALUES.CAL_DESC_ID: T
NORMAL: TABLE.COLVALUES.CAL_HISTORY_ID: T
NORMAL: TABLE.COLVALUES.TOTAL_SOLUTION_OK: T
NORMAL: TABLE.COLVALUES.TOTAL_FIT: T
NORMAL: TABLE.COLVALUES.TOTAL_FIT_WEIGHT: T
NORMAL: TABLE.COLVALUES.SOLUTION_OK: T
NORMAL: TABLE.COLVALUES.FIT: T
NORMAL: TABLE.COLVALUES.FIT_WEIGHT: T
NORMAL: TABLE.KEYWORDFIELDNAMES: T
NORMAL: CAL_DESC.COLNAMES: T
NORMAL: CAL_DESC.COLKEYWORDNAMES.NUM_SPW: T
NORMAL: CAL_DESC.COLKEYWORDNAMES.NUM_CHAN: T
NORMAL: CAL_DESC.COLKEYWORDNAMES.NUM_RECEPTORS: T
NORMAL: CAL_DESC.COLKEYWORDNAMES.N_JONES: T
NORMAL: CAL_DESC.COLKEYWORDNAMES.SPECTRAL_WINDOW_ID: T
NORMAL: CAL_DESC.COLKEYWORDNAMES.CHAN_FREQ: T
NORMAL: CAL_DESC.COLKEYWORDNAMES.MEAS_FREQ_REF: T
NORMAL: CAL_DESC.COLKEYWORDNAMES.CHAN_WIDTH: T
NORMAL: CAL_DESC.COLKEYWORDNAMES.CHAN_RANGE: T
NORMAL: CAL_DESC.COLKEYWORDNAMES.POLARIZATION_TYPE: T
NORMAL: CAL_DESC.COLKEYWORDNAMES.JONES_TYPE: T
NORMAL: CAL_DESC.COLKEYWORDNAMES.MS_NAME: T
NORMAL: CAL_DESC.COLVALUES.NUM_SPW: T
NORMAL: CAL_DESC.COLVALUES.NUM_CHAN: T
NORMAL: CAL_DESC.COLVALUES.NUM_RECEPTORS: T
NORMAL: CAL_DESC.COLVALUES.N_JONES: T
NORMAL: CAL_DESC.COLVALUES.SPECTRAL_WINDOW_ID: T
NORMAL: CAL_DESC.COLVALUES.CHAN_FREQ: T
NORMAL: CAL_DESC.COLVALUES.MEAS_FREQ_REF: T
NORMAL: CAL_DESC.COLVALUES.CHAN_WIDTH: T
NORMAL: CAL_DESC.COLVALUES.CHAN_RANGE: T
NORMAL: CAL_DESC.COLVALUES.POLARIZATION_TYPE: T
NORMAL: CAL_DESC.COLVALUES.JONES_TYPE: T
NORMAL: CAL_DESC.COLVALUES.MS_NAME: T
NORMAL: CAL_DESC.KEYWORDFIELDNAMES: T
NORMAL: CAL_HISTORY.COLNAMES: T
NORMAL: CAL_HISTORY.COLKEYWORDNAMES.CAL_PARMS: T
NORMAL: CAL_HISTORY.COLKEYWORDNAMES.CAL_TABLES: T
NORMAL: CAL_HISTORY.COLKEYWORDNAMES.CAL_SELECT: T
NORMAL: CAL_HISTORY.COLKEYWORDNAMES.CAL_NOTES: T
NORMAL: CAL_HISTORY.COLVALUES.CAL_PARMS: T
NORMAL: CAL_HISTORY.COLVALUES.CAL_TABLES: T
NORMAL: CAL_HISTORY.COLVALUES.CAL_SELECT: T
NORMAL: CAL_HISTORY.COLVALUES.CAL_NOTES: T
NORMAL: CAL_HISTORY.KEYWORDFIELDNAMES: T
```