



RiverTools Reference Guide



March, 2001 Edition
Copyright © Rivix Limited Liability Company
All Rights Reserved

Restricted Rights Notice

The RiverTools™ and IDL® software program and the accompanying procedures, functions, and documentation described herein are sold under license agreement. Their use, duplication, and disclosure are subject to the restrictions stated in the license agreement. Research Systems, Inc., reserves the right to make changes to this document at any time and without notice.

Limitation of Warranty

Rivix LLC and Research Systems, Inc. makes no warranties, either express or implied, as to any matter not expressly set forth in the license agreement, including without limitation the condition of the software, merchantability, or fitness for any particular purpose.

Research Systems, Inc. shall not be liable for any direct, consequential, or other damages suffered by the Licensee or any others resulting from use of the RiverTools or IDL software package or its documentation.

Permission to Reproduce this Manual

If you are a licensed user of this product, Research Systems, Inc. grants you a limited, non-transferable license to reproduce this particular document provided such copies are for your use only and are not sold or distributed to third parties. All such copies must contain the title page and this notice page in their entirety.

Acknowledgments

IDL® is a registered trademark of Research Systems Inc., registered in the United States Patent and Trademark Office, for the computer program described herein.

RiverTools™ is a trademark of Rivix LLC.

Numerical Recipes™ is a trademark of Numerical Recipes Software. Numerical Recipes routines are used by permission.

GRG2™ is a trademark of Windward Technologies, Inc. The GRG2 software for nonlinear optimization is used by permission.

NCSA Hierarchical Data Format (HDF) Software Library and Utilities
Copyright © 1988-1998 The Board of Trustees of the University of Illinois
All rights reserved.

CDF Library
Copyright © 1999
National Space Science Data Center
NASA/Goddard Space Flight Center

NetCDF Library
Copyright © 1993-1996 University Corporation for Atmospheric Research/Unidata

HDF EOS Library
Copyright © 1996 Hughes and Applied Research Corporation

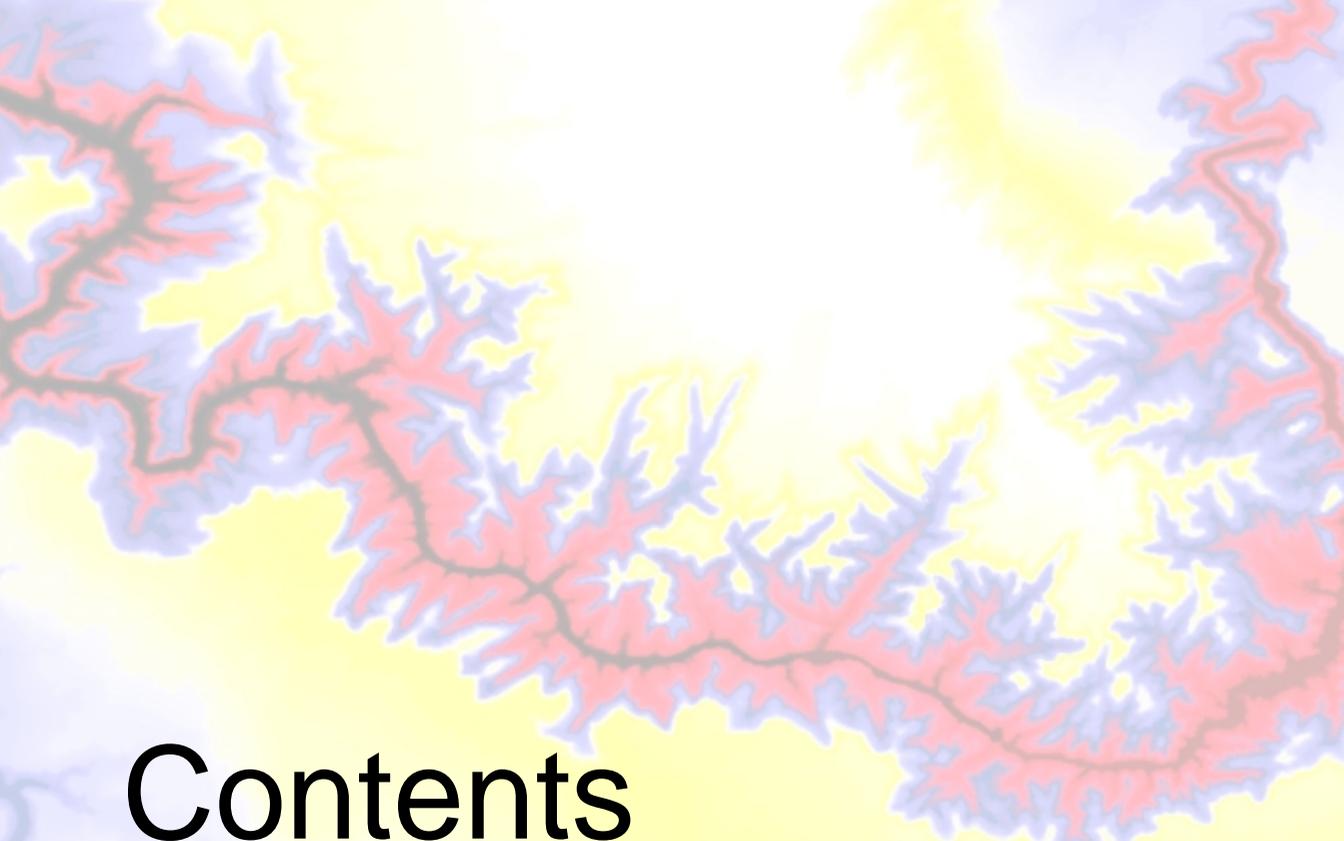
This software is based in part on the work of the Independent JPEG Group.

This product contains StoneTable™, by StoneTablet Publishing. All rights to StoneTable™ and its documentation are retained by StoneTablet Publishing, PO Box 12665, Portland OR 97212-0665. Copyright © 1992-1997 StoneTablet Publishing

WASTE text engine © 1993-1996 Marco Piovaneli

Portions of this software are copyrighted by INTERSOLV, Inc., 1991-1998.

Other trademarks and registered trademarks are the property of the respective trademark holders.



Contents

Reference:	
RiverTools Commands	7
Functional Listing of RiverTools Routines	9
RT_Add_Window	12
RT_Check_Grid_Type	13
RT_Check_Overwrite	15
RT_Check_Window	16
RT_Close_Plot	17
RT_Close_Window	18
RT_Color_Index	19
RT_Color_List	20
RT_Compress_File	21
RT_Copy_File	22
RT_Convert_Angular_Units	23
RT_Convert_DEM_To_Geo	24

RT_Convert_DEM_To_UTM	26
RT_Convert_DEM_To_Win	28
RT_Convert_DEM_Units	30
RT_Convert_Geo_To_DMS	31
RT_Convert_Geo_To_UTM	32
RT_Convert_Linear_Units	34
RT_Convert_Mask_To_RTM	35
RT_Convert_UTM_To_Geo	36
RT_Convert_Win_To_DEM	38
RT_Count_Lines	40
RT_Delete_File	42
RT_Ellipsoid_Info	43
RT_Ellipsoid_List	45
RT_Expand_Filenames	46
RT_File_OK	47
RT_Find_File	48
RT_Fit_Line	49
RT_Flow_Path_Coords	51
RT_Get_Array	53
RT_Get_Available_DEMs	54
RT_Get_Available_Grids	55
RT_Get_Available_Masks	56
RT_Get_Binomial_Cascade	57
RT_Get_Byte_Order	58
RT_Get_CDF	59
RT_Get_Convex_Hull	60
RT_Get_File_Where	61
RT_Get_Flow_Codes	62
RT_Get_Kid_ID	64
RT_Get_Parent_ID	65
RT_Get_Pixel_Coords	66
RT_Get_Pixel_Sizes	68
RT_Get_RTI_Filename	70
RT_Get_RTI_Record	71
RT_Get_RTM_Cell_Area	74
RT_Get_RTM_Info	75

RT_Get_RTV_Info	77
RT_Get_RTV_Record	79
RT_Get_SST_Info	82
RT_GUI_Replace_String	84
RT_GUI_View_DEM_Info	85
RT_GUI_View_Text_File	86
RT_Heap_Sort	88
RT_Init_RTG_File	89
RT_Init_RTM_File	90
RT_Make_Filled_DEM	91
RT_Make_Flow_Grid	92
RT_Make_River_Files	93
RT_Make_RTI_File	95
RT_Make_Theoretical_DEM	96
RT_Make_Treefile	98
RT_Not_Same_Order	99
RT_Open_Plot	100
RT_Parse_Filepath	102
RT_Plot_SST	104
RT_Print	107
RT_Read_DEM	108
RT_Read_DEM_Info	109
RT_Read_Function	111
RT_Read_Grid	112
RT_Read_Horton_Data	114
RT_Read_RTM	116
RT_Read_RTM_As_Grid	118
RT_Read_RTM_Cell	119
RT_Read_RTV_Data	121
RT_Read_Subgrid	123
RT_Read_Text_File	125
RT_Read_XYZ_Values	126
RT_Remove_Window	128
RT_Resize	129
RT_Save_Function	131
RT_Set_View	132

RT_Set_Window	134
RT_String	135
RT_Uncompress_File	136
RT_Write_Subgrid	137



Reference:

RiverTools Commands

Most modern software programs provide a convenient point-and-click graphical user interface (GUI) that makes it possible for users to draw on a large base of functionality and to achieve impressive results with a minimum of training. When it comes to rapidly performing common tasks and changing input parameters to subroutines, a graphical interface is much easier to use and far more efficient than a command-line interface. However, a command-line interface is more flexible when it comes to addressing unanticipated scenarios, large batch jobs, or unusual cases that a typical user may not encounter. This is especially true of scientific and Geographic Information System (GIS) software, which is why such programs usually offer a “scripting language” in addition to their graphical interface. Users who have purchased RiverTools+IDL can use IDL as a “scripting language” to extend RiverTools, even though IDL is much more powerful and complete than a typical scripting language. It is often the case that a task not available in the RiverTools graphical user interface can be accomplished by typing a few lines of IDL and RiverTools commands at the IDL command line.

Beneath the graphical user interface, RiverTools is built upon a rich infrastructure of routines that are written in Interactive Data Language (IDL) and these can be called

from the IDL command line, just like IDL's built-in routines. These routines can be viewed as an extension of the IDL language, and provide much greater flexibility for users who are familiar with IDL and want to get the most out of RiverTools. They can also be called by user-written IDL routines to make them work more seamlessly with RiverTools. Using these routines wherever possible in code development has many advantages, such as (1) greater assurance of compatibility with future RiverTools versions, (2) greater efficiency, since most RiverTools routines employ optimal algorithms or build upon the superior array processing capabilities of IDL, (3) no need to worry about the details of the C-like data structures used in RiverTools files, and (4) faster code development. As explained in the section “[Adding Extensions](#)” in the *User's Guide*, RiverTools also provides a simple mechanism for adding user routines to the User pull-down menu.

While there are several hundred low-level RiverTools routines, the intent of this chapter is to document the ones that are most useful for extending the capabilities of RiverTools. In version 2.4, all of the RiverTools commands have been renamed so as to have a “name space” that does not conflict with the names of IDL or ENVI procedures and functions. Each RiverTools command now begins with “RT_”. Commands that launch a dialog or graphical user interface (GUI) now begin with “RT_GUI” and are not documented here. As in version 2.0, the remaining portion of a command name follows an easy-to-remember “verb-noun” or “verb-adjective-noun” style, with separate words separated by underscores. A typical example is `RT_Read_Grid`.

In addition to command arguments, many of the RiverTools commands accept “keywords,” and these work exactly as they do for IDL commands, namely (1) they are usually optional, (2) the order in which they appear in the procedure call doesn't matter, (3) they may pass a value into or out of the routine, (4) they sometimes serve as boolean flags, in which case their name, preceded by the '/' character sets the flag, and (5) only the first few unique letters are required.

Please note that the routines described here will only work as described with RiverTools version 2.4 or higher.

Functional Listing of RiverTools Routines

Colors

[RT_Color_Index](#)

[RT_Color_List](#)

Coordinate Conversion

[RT_Convert_Angular_Units](#)

[RT_Convert_Linear_Units](#)

[RT_Convert_DEM_To_Geo](#)

[RT_Convert_UTM_To_Geo](#)

[RT_Convert_DEM_To_UTM](#)

[RT_Convert_Win_To_DEM](#)

[RT_Convert_DEM_To_Win](#)

[RT_Get_Pixel_Coords](#)

[RT_Convert_Geo_To_DMS](#)

[RT_Get_Pixel_Sizes](#)

[RT_Convert_Geo_To_UTM](#)

Ellipsoids

[RT_Ellipsoid_Info](#)

[RT_Ellipsoid_List](#)

File Handling

[RT_Check_Overwrite](#)

[RT_File_OK](#)

[RT_Compress_File](#)

[RT_Find_File](#)

[RT_Copy_File](#)

[RT_Parse_Filepath](#)

[RT_Delete_File](#)

[RT_Uncompress_File](#)

[RT_Expand_Filenames](#)

Network Extraction

RT_Flow_Path_Coords	RT_Make_Flow_Grid
RT_Get_Flow_Codes	RT_Make_River_Files
RT_Get_Kid_ID	RT_Make_Treefile
RT_Get_Parent_ID	RT_Read_Horton_Data
RT_Make_Filled_DEM	

RTG (and DEM) Files

RT_Check_Grid_Type	RT_Make_Theoretical_DEM
RT_Convert_DEM_Units	RT_Read_DEM
RT_Get_Available_DEMs	RT_Read_DEM_Info
RT_Get_Available_Grids	RT_Read_Grid
RT_Get_File_Where	RT_Read_RTM_As_Grid
RT_Init_RTG_File	RT_Read_Subgrid
RT_Make_Filled_DEM	RT_Write_Subgrid

RTI Files

RT_Get_RTI_Filename	RT_Make_RTI_File
RT_Get_RTI_Record	RT_Read_DEM_Info
RT_GUI_View_DEM_Info	

RTM Files

RT_Convert_Mask_To_RTM	RT_Init_RTM_File
RT_Get_Available_Masks	RT_Read_RTM
RT_Get_RTM_Cell_Area	RT_Read_RTM_As_Grid
RT_Get_RTM_Info	RT_Read_RTM_Cell

RTV Files

RT_Get_RTV_Info	RT_Read_RTV_Data
RT_Get_RTV_Record	RT_Read_RTV_Data

Special Plots

RT_Get_Binomial_Cascade	RT_Plot_SST
---	-----------------------------

Text Files

RT_Count_Lines	RT_Read_Text_File
RT_GUI_View_Text_File	RT_Read_XYZ_Values
RT_Read_Function	RT_Save_Function

Utilities

RT_Fit_Line	RT_Heap_Sort
RT_Get_Array	RT_Not_Same_Order
RT_Get_Byte_Order	RT_Print
RT_Get_CDF	RT_Resize
RT_Get_Convex_Hull	RT_Set_View
RT_Get_SST_Info	RT_String
RT_GUI_Replace_String	

Windows

RT_Add_Window	RT_Open_Plot
RT_Check_Window	RT_Set_Window
RT_Close_Plot	RT_Remove_Window
RT_Close_Window	

RT_Add_Window

The `RT_Add_Window` procedure adds a window to the roster of windows that is recognized and managed by RiverTools. It is meant for use by programmers who want to extend the RiverTools environment.

Syntax

`RT_Add_Window, BASE_ID, WINDOW_ID, TITLE`

Arguments

BASE_ID

The long (4-byte) integer ID of the window's top-level base.

WINDOW_ID

The long (4-byte) integer ID of the window's index.

TITLE

The title that appears in the title bar at the top of the window.

See Also

[RT_Check_Window](#), [RT_Close_Window](#), [RT_Open_Plot](#), [RT_Remove_Window](#), [RT_Set_Window](#)

RT_Check_Grid_Type

The RT_Check_Grid_Type procedure returns the data type of a RiverTools grid file (RTG) or checks that a candidate data type is consistent with the grid dimensions and file size.

Syntax

```
RT_Check_Grid_Type, RTG_File, OK [, TYPE=data_type] [, BPE=value]  
[, NCOLS=value] [, NROWS=value] [, FILESIZE=value] [, CHECKSIZE=value]  
[, /SILENT]
```

Arguments

RTG_File

The filename of the RiverTools grid file (RTG) to be examined.

OK

Returns 1b if the file was found and the data type was consistent with other information, and 0b otherwise.

Keywords

TYPE

The data type of the grid file, which may be 'BYTE', 'INTEGER', 'LONG', 'FLOAT', or 'DOUBLE'. If specified, it is viewed as a candidate data type.

BPE

The number of bytes per element (BPE) in the grid file.

NCOLS

The number of columns in the grid file. If unspecified, the number of columns in the currently selected DEM is used.

NROWS

The number of rows in the grid file. If unspecified, the number of rows in the currently selected DEM is used.

FILESIZE

The size, in bytes, of the input grid file.

CHECKSIZE

The expected size, in bytes, of the grid file if the specified data type is correct.

SILENT

Set this keyword to suppress warnings and other information.

RT_Check_Overwrite

The RT_Check_Overwrite procedure checks to see whether a file with the specified name already exists in the current directory, and if so, opens a dialog to ask whether the user wants to overwrite it. It also checks whether the user has permission to create a file in the current directory.

Syntax

RT_Check_Overwrite, *Filename*, *OK*

Arguments

Filename

The name of the file to be created and/or overwritten.

OK

Returns 1 if the user has permission to write to the current directory and the file either doesn't exist, or the user has indicated that it is OK to overwrite the existing file.

RT_Check_Window

The `RT_Check_Window` procedure is used to check whether a window is listed on the roster of windows managed by RiverTools. This procedure is meant for use by programmers who want to extend the RiverTools environment.

Syntax

`RT_Check_Window, Window_ID, OK`

Arguments

Window_ID

The window index or ID, a long (4-byte) integer, of the window that is to be removed from the roster. This index can be obtained with the [RT_Open_Plot](#) procedure when a window is first opened.

OK

Returns 1b if the window exists, and 0b otherwise.

See Also

[RT_Add_Window](#), [RT_Close_Window](#), [RT_Open_Plot](#), [RT_Remove_Window](#), [RT_Set_Window](#)

RT_Close_Plot

The RT_Close_Plot procedure closes the current output device and restores any device settings that may have been modified to their normal values. It is called every time a RiverTools plot has been completed. When plotting to a PostScript file or a printer, the plot is not complete until this procedure has been called.

Syntax

```
RT_Close_Plot [, Message] [, /SILENT]
```

Arguments

Message

A scalar string to be printed to the main RiverTools window when the plot is finished. The default is “Finished with plot.”

Keywords

SILENT

Set this keyword to suppress printing a message when finished.

See Also

[RT_Open_Plot](#)

RT_Close_Window

The `RT_Close_Window` procedure is used to close a window that is managed by RiverTools. This procedure also calls the [RT_Remove_Window](#) procedure which removes the window from the roster of windows managed by RiverTools. This procedure is meant for use by programmers who want to extend the RiverTools environment.

Syntax

`RT_Close_Window, Base_ID`

Arguments

Base_ID

The top-level base ID, a long (4-byte) integer, of the window that is to be removed from the roster.

See Also

[RT_Add_Window](#), [RT_Check_Window](#), [RT_Open_Plot](#), [RT_Remove_Window](#), [RT_Set_Window](#)

RT_Color_Index

The `RT_Color_Index` function can be used to obtain a color index for many basic colors.

Syntax

Index = `RT_Color_Index(`*Color_Name*`)`

Arguments

Color_Name

A scalar string that is the name of a recognized color. Examples include 'black' and 'red'. The [RT_Color_List](#) function can be used to obtain a complete list of recognized basic color names.

See Also

[RT_Color_List](#)

RT_Color_List

The `RT_Color_List` function can be used to obtain a list of recognized basic color names.

Syntax

```
Color_List = RT_Color_List([/NO_SORT])
```

Keywords

NO_SORT

Set this keyword to prevent the color list from being sorted alphabetically. The color list is then sorted by color index.

See Also

[RT_Color_Index](#)

RT_Compress_File

The RT_Compress_File procedure compresses an input file with the well-known GZIP algorithm to create a compressed file. The original file is not deleted. This procedure could be used in a loop to compress a large number of files.

Syntax

RT_Compress_File, *In_File*, *Out_File*

Arguments

In_File

The input file to be compressed with GZIP.

Out_File

The compressed output file, which should have the filename extension “.gz”.

See Also

[RT_Uncompress_File](#), [RT_Delete_File](#), [RT_Copy_File](#)

RT_Copy_File

The RT_Copy_File procedure makes an exact copy of the specified RTG file.

Syntax

RT_Copy_File, *In_File*, *Out_File*, *Type*, *Ncols*, *Nrows*

Arguments

In_File

The name of the input RTG file.

Out_File

The name of the output RTG file.

Type

The data type of the input file.

Ncols

The number of columns in the grid file.

Nrows

The number of rows in the grid file.

See Also

[RT_Delete_File](#), [RT_Compress_File](#), [RT_Uncompress_File](#)

RT_Convert_Angular_Units

The `RT_Convert_Angular_Units` procedure converts position coordinates from one set of angular units to another.

Syntax

`RT_Convert_Angular_Units, Old_Units, Old_X, Old_Y, New_Units, New_X, New_Y`

Arguments

Old_Units

The original angular units as a scalar string, which can be: 'DEGREES', 'MINUTES', 'ARCMINUTES', 'SECONDS', 'ARCSECONDS', 'RADIANS', or 'DMS'. 'DMS' indicates that units are in Degrees-Minutes-Seconds format: DDD.MMSS.

Old_X

The original x-coordinates, as a scalar or 1D array.

Old_Y

The original y-coordinates, as a scalar or 1D array.

New_Units

The new units as a scalar string, which can be: 'DEGREES', 'MINUTES', 'ARCMINUTES', 'SECONDS', 'ARCSECONDS', 'RADIANS', or 'DMS'. 'DMS' indicates that units are in Degrees-Minutes-Seconds format: DDD.MMSS.

New_X

The new x-coordinates, as a scalar or 1D array.

New_Y

The new y-coordinates, as a scalar or 1D array.

See Also

[RT_Convert_Linear_Units](#), [RT_Convert_Geo_To_UTM](#),
[RT_Convert_Geo_To_DMS](#)

RT_Convert_DEM_To_Geo

The RT_Convert_DEM_To_Geo procedure uses information in a RiverTools info (RTI) file to convert column/row coordinates for pixels in a DEM or other RiverTools grid to Geographic or lon/lat coordinates.

Syntax

```
RT_Convert_DEM_To_Geo, Col, Row, Lon, Lat [, /CENTER]  
[, RTI_FILE=filename] [, /REPORT]
```

Arguments

Col

The column indices of pixels in a DEM or other RiverTools grid.

Row

The row indices of pixels in a DEM or other RiverTools grid.

Lon

The longitudes of the pixels.

Lat

The latitudes of the pixels.

Keywords

CENTER

Set this keyword to return longitude and latitude of a pixel's center as opposed to its southeast corner.

RTI_FILE

This keyword can be used to specify an RTI file from which georeferencing info should be obtained. If unspecified, the RTI file for the current DEM is used.

REPORT

Set this keyword to print both input and output coordinates to the main window. If the Col and Row arguments are arrays, then the average values are printed.

See Also

[RT_Convert_DEM_To_UTM](#)

RT_Convert_DEM_To_UTM

The RT_Convert_DEM_To_UTM procedure uses information in a RiverTools info (RTI) file to convert column/row coordinates for pixels in a DEM or other RiverTools grid to UTM (Universal Transverse Mercator) coordinates.

Syntax

```
RT_Convert_DEM_To_UTM, Col, Row, X_UTM, Y_UTM [, /CENTER]  
[, RTI_FILE=filename] [, /REPORT]
```

Arguments

Col

The column indices of pixels in a DEM or other RiverTools grid.

Row

The row indices of pixels in a DEM or other RiverTools grid.

X_UTM

The longitudes of the pixels.

Y_UTM

The latitudes of the pixels.

Keywords

CENTER

Set this keyword to return UTM coordinates of a pixel's center as opposed to its southeast corner.

RTI_FILE

This keyword can be used to specify an RTI file from which georeferencing info should be obtained. If unspecified, the RTI file for the current DEM is used.

REPORT

Set this keyword to print both input and output coordinates to the main window. If the Col and Row arguments are arrays, then the average values are printed.

See Also

[RT_Convert_DEM_To_Geo](#)

RT_Convert_DEM_To_Win

The RT_Convert_DEM_To_Win procedure converts between column/row coordinates in a DEM to window coordinates, assuming they differ by an integer scale factor.

Syntax

```
RT_Convert_DEM_To_Win, Col, Row, Xwin, Ywin [, SCALE=value] [, /NORMAL]  
[, RTI_FILE=filename]
```

Arguments

Col

The column indices of pixels in a DEM or other RiverTools grid.

Row

The row indices of pixels in a DEM or other RiverTools grid.

Xwin

The x-coordinate of a pixel in a graphics window, measured in pixels from the left edge.

Ywin

The y-coordinate of a pixel in a graphics window, measured in pixels from the top edge.

Keywords

SCALE

A scale factor which gives the ratio of window size to DEM size.

NORMAL

Set this keyword to return Xwin and Ywin as normalized coordinates that range between 0 and 1.

RTI_FILE

This keyword can be used to specify an RTI file from which georeferencing info should be obtained. If unspecified, the RTI file for the current DEM is used.

See Also

[RT_Convert_Win_To_DEM](#)

RT_Convert_DEM_Units

The RT_Convert_DEM_Units procedure multiplies the values in a 2D input array by a constant factor in order to convert the values to other units. The minimum and maximum elevations in the converted DEM are printed in the main log window.

Syntax

```
RT_Convert_DEM_Units, DEM [, Data_Type] [, FACTOR=value] [, /FT2M]  
[, /M2FT]
```

Arguments

DEM

The input and output DEM, as a 2D array. Values in the input DEM are overwritten with the converted values.

Data_Type

The output data type as a string which may be 'BYTE', 'INTEGER', 'LONG', 'FLOAT' or 'DOUBLE'. The default is 'DOUBLE'. Using an integer data type may lead to loss of precision.

Keywords

FACTOR

The conversion factor. If the FT2M or M2FT keywords are set, this keyword is ignored and the appropriate (exact) conversion factor is used. Otherwise, the default is 1.

FT2M

Set this keyword to convert from units of feet to meters using the exact conversion factor, (1.0 / 3.28083989501)

M2FT

Set this keyword to convert from units of meters to feet using the exact conversion factor, 3.28083989501.

RT_Convert_Geo_To_DMS

The `RT_Convert_Geo_To_DMS` procedure converts coordinates from Geographic lon/lat coordinates in decimal degrees to degrees, minutes, and seconds.

Syntax

`RT_Convert_Geo_To_DMS, Lon, Lat, Lon_Deg, Lon_Min, Lon_Sec, Lat_Deg, Lat_Min, Lat_Sec`

Arguments

Lon

The input longitude in decimal degrees.

Lat

The input latitude in decimal degrees.

Lon_Deg

The number of degrees longitude.

Lon_Min

The number of minutes of longitude, between 0 and 60.

Lon_Sec

The number of seconds of longitude, between 0 and 60.

Lat_Deg

The number of degrees latitude.

Lat_Min

The number of minutes of latitude, between 0 and 60.

Lat_Sec

The number of seconds of latitude, between 0 and 60.

See Also

[RT_Convert_Geo_To_UTM](#), [RT_Convert_UTM_To_Geo](#)

RT_Convert_Geo_To_UTM

The RT_Convert_Geo_To_UTM procedure converts coordinates from Geographic lon/lat (decimal degrees) to UTM (Universal Transverse Mercator). A UTM zone typically spans 6 degrees of longitude.

Syntax

```
RT_Convert_Geo_To_UTM, Lon, Lat, X, Y, Zone, North [, Scale, Zone_Letter]  
[, ELLIPSOID=value] [, /NO_FALSE] [, /REPORT]
```

Arguments

Lon

The input longitude, in decimal degrees.

Lat

The input latitude, in decimal degrees.

X

The output UTM x-coordinate.

Y

The output UTM y-coordinate.

Zone

The UTM zone number, between 1 and 60, that corresponds to a range of longitudes.

North

Returns 1 if latitude is greater than or equal to 0.

Scale

The scale on the central meridian of the zone.

Zone_Letter

A letter code as a scalar string that corresponds to a range of latitudes (8 degrees) in a given UTM zone.

Keywords

ELLIPSOID

A scalar string which specifies the reference ellipsoid to use when interpreting the input longitude and latitude. A list of the ellipsoids (and their corresponding strings) that are recognized by RiverTools can be returned with the [RT_Ellipsoid_List](#) function. If unspecified, the ellipsoid specified in the Planet Info section of the preferences is used (**File** → **Set Preferences**).

NO_FALSE

Unless this keyword is set, a false easting of 500000 is added to the x-coordinate.

REPORT

Set this keyword to print a summary report to the main window.

See Also

[RT_Convert_Geo_To_DMS](#), [RT_Convert_UTM_To_Geo](#)

RT_Convert_Linear_Units

The `RT_Convert_Linear_Units` procedure converts position coordinates from one set of linear units to another.

Syntax

`RT_Convert_Linear_Units, Old_Units, Old_X, Old_Y, New_Units, New_X, New_Y`

Arguments

Old_Units

The original linear units as a scalar string, which can be: 'METERS', 'M', 'FEET', 'FT', 'YARDS', 'KILOMETERS', 'KM', 'MILES', 'MI', or 'NAUTICAL_MILES'.

Old_X

The original x-coordinates, as a scalar or 1D array.

Old_Y

The original y-coordinates, as a scalar or 1D array.

New_Units

The new linear units as a scalar string, which can be: 'METERS', 'M', 'FEET', 'FT', 'YARDS', 'KILOMETERS', 'KM', 'MILES', 'MI', or 'NAUTICAL_MILES'.

New_X

The new x-coordinates, as a scalar or 1D array.

New_Y

The new y-coordinates, as a scalar or 1D array.

See Also

[RT_Convert_Angular_Units](#), [RT_Convert_UTM_To_Geo](#)

RT_Convert_Mask_To_RTM

The `RT_Convert_Mask_To_RTM` procedure converts a grid file that represents a mask to a more compact RiverTools mask (RTM) file. Values less than or equal to 0 in the grid file are assumed to represent “unmasked pixels”, and every other unique, integer value is assumed to represent the pixels in a distinct “mask cell.”

Syntax

`RT_Convert_Mask_To_RTM, RTG_File, RTM_File`

Arguments

RTG_File

The input RiverTools grid (RTG) file, assumed to contain long (4-byte) integers and to have the same dimensions as the current DEM.

RTM_File

The output RiverTools mask (RTM) file, which should end with the filename extension “.rtm”.

RT_Convert_UTM_To_Geo

The RT_Convert_UTM_To_Geo procedure converts coordinates from UTM (Universal Transverse Mercator) to Geographic lon/lat (decimal degrees).

Syntax

```
RT_Convert_UTM_To_Geo, X, Y, Zone, Lon, Lat [, ELLIPSOID=value]  
[, /NO_FALSE] [, /REPORT]
```

Arguments

X

The input UTM x-coordinate.

Y

The input UTM y-coordinate.

Zone

The input UTM zone number, between 1 and 60.

Lon

The output longitude, in decimal degrees.

Lat

The output latitude, in decimal degrees.

Keywords

ELLIPSOID

A scalar string which specifies the reference ellipsoid to use when interpreting the input longitude and latitude. A list of the ellipsoids (and their corresponding strings) that are recognized by RiverTools can be returned with the [RT_Ellipsoid_List](#) function. If unspecified, the ellipsoid specified in the Planet Info section of the preferences is used (**File** → **Set Preferences**).

NO_FALSE

Unless this keyword is set, a false easting of 500000 is subtracted from the input x-coordinate.

REPORT

Set this keyword to print a summary report to the main window.

See Also

[RT_Convert_Geo_To_UTM](#), [RT_Convert_Geo_To_DMS](#)

RT_Convert_Win_To_DEM

The RT_Convert_Win_To_DEM procedure converts window coordinates in an image of a DEM to column/row coordinates in a DEM, assuming they differ by an integer scale factor. If the window dimensions are smaller than the DEM dimensions, then there is a square block of DEM pixels that all have the same “window coordinates” and only a single representative column/row pair is returned.

Syntax

```
RT_Convert_Win_To_DEM, Xwin, Ywin, Col, Row [, SCALE=value] [, /NORMAL]
[, RTI_FILE=filename]
```

Arguments

Xwin

The x-coordinate of a pixel in a graphics window, measured in pixels from the left edge.

Ywin

The y-coordinate of a pixel in a graphics window, measured in pixels from the top edge.

Col

The column indices of pixels in a DEM or other RiverTools grid.

Row

The row indices of pixels in a DEM or other RiverTools grid.

Keywords

SCALE

A scale factor which gives the ratio of window size to DEM size.

NORMAL

Set this keyword to return Xwin and Ywin as normalized coordinates that range between 0 and 1.

RTI_FILE

This keyword can be used to specify an RTI file from which georeferencing info should be obtained. If unspecified, the RTI file for the current DEM is used.

See Also

[RT_Convert_DEM_To_Win](#)

RT_Count_Lines

The RT_Count_Lines procedure counts all of the lines in a specified set of text files. The files must all exist in the same directory or folder.

Syntax

```
RT_Count_Lines, Nlines, [, Files] [, /ALL_PRO_FILES] [, /ALL_TEXT_FILES]  
[, DIRECTORY=directory_name] [, NCHARS=variable]  
[, NULL_NLINES=variable] [, /REPORT] [, /SILENT]
```

Arguments

Nlines

The total number of lines in the specified set of text files.

Files

A string array or scalar which contains a list of filenames. If the DIRECTORY keyword is set, they are assumed to exist in the specified directory. This argument is not required if the ALL_PRO_FILES or ALL_TEXT_FILES keyword is set.

Keywords

ALL_PRO_FILES

Set this keyword to count lines in all files with the extension “.pro” in the current or specified directory.

ALL_TEXT_FILES

Set this keyword to count lines in all files with the extension “.txt” in the current or specified directory.

DIRECTORY

Set this keyword to the directory which contains the text files. If unspecified, then the first file in Files is parsed for a directory and if still unspecified then the current directory is used.

NCHARS

Set this keyword to a named variable to return the total number of characters in the set of text files.

NULL_NLINES

Set this keyword to a named variable to return the total number of null lines in the set of text files. A null line contains only a newline character.

REPORT

Set this keyword to print a summary report in the main window.

SILENT

Set this keyword to suppress all information messages.

See Also

[RT_Read_Text_File](#), [RT_GUI_View_Text_File](#)

RT_Delete_File

The RT_Delete_File procedure deletes the specified file. This procedure should be used with care, because no warning messages are issued and once deleted there is no way to retrieve the file.

Syntax

RT_Delete_File, *Filename*

Arguments

Filename

The name of the file to be deleted. If the filename does not contain a path, then the file is assumed to exist in the current directory.

See Also

[RT_Copy_File](#), [RT_Find_File](#)

RT_Ellipsoid_Info

The RT_Ellipsoid_Info function returns a structure with the following fields:

- a = semi-major axis
- b=semi-minor axis
- f=flattening ratio (a-b)/b
- e=eccentricity
- g= inverse flattening ratio (1/f)
- used_by = a string with information about where the ellipsoid is used
- comment = any comments regarding the ellipsoid.

Ellipsoids are usually defined in terms of their semi-major axis, a, and their inverse flattening ratio, g. The other properties are computed from these.

Syntax

```
Info = RT_Ellipsoid_Info(Ellipsoid_Name [, /METERS] [, /REPORT])
```

Arguments

Ellipsoid_Name

A scalar string which specifies the name of an ellipsoid. A list of recognized ellipsoid names can be obtained with the [RT_Ellipsoid_List](#) function. Ellipsoid names typically consist of the name of a person or organization, which may be followed by a space and then a four-digit year, such as “CLARKE 1866”, “KRASSOVSKY” and “WGS 1984”. “AUTHALIC SPHERE” indicates a sphere with radius 6371 km.

Keywords

METERS

Set this keyword to return the semi-major and semi-minor axes in units of meters instead of kilometers.

REPORT

Set this keyword to print a summary report in the main window.

See Also

[RT_Ellipsoid_List](#)

RT_Ellipsoid_List

The `RT_Ellipsoid_List` function returns a list of the ellipsoids that are recognized by RiverTools. Over 50 ellipsoids are recognized by RiverTools 2.4. Ellipsoid names typically consist of the name of a person or organization, which is often followed by a space and then a four-digit year, such as “CLARKE 1866”, “KRASSOVSKY” and “WGS 1984”. This function has no arguments or keywords.

Syntax

```
List = RT_Ellipsoid_List()
```

See Also

[RT_Ellipsoid_Info](#)

RT_Expand_Filenames

The RT_Expand_Filenames procedure uses the In_String and Out_String arguments, which may contain one asterisk each (a “wildcard”), to construct a set of In_Files and a set of Out_Files. This procedure is useful when batch processing a set of input files to create a corresponding set of output files.

Syntax

```
RT_Expand_Filenames, In_String, In_Files, Out_String, Out_Files [, In_File_Count]
[, OK]
```

Arguments

In_String

An input string containing a single asterisk, e.g. “*.dem”.

In_Files

The set of files in the current directory that match In_String.

Out_String

An output string containing a single asterisk, e.g. “KY_*.rtg”.

Out_Files

A returned set of filenames, such that there is one for each file in In_Files and the asterisk means the same thing as it does for the corresponding entry in the string array In_Files.

In_File_Count

The number of files in the current directory that match In_String.

OK

Returns 1 if no problems are encountered, and 0 otherwise.

RT_File_OK

The RT_File_OK function returns a 1 if the specified file exists and 0 otherwise. If the filename corresponds to one of the main files in the currently selected RiverTools data set (such as the DEM file, RTI file or flow grid file), a dialog is displayed with information about the file and why it is required.

Syntax

```
OK = RT_File_OK(Filename [, /SILENT])
```

Arguments

Filename

The name of the file to be checked. If the filename does not include a path, then the current directory is searched.

Keywords

SILENT

Set this keyword to suppress all information messages.

See Also

[RT_Find_File](#)

RT_Find_File

The RT_Find_File procedure searches for the specified file in the current directory, and then in IDL's system variable, !PATH. Differences between operating systems are taken into account.

Syntax

```
RT_Find_File, Filename, Filepath, Directory [, /REPORT]
```

Arguments

Filename

The name of the file for which to search.

Filepath

If the file is found, the fully-qualified “filepath” is returned.

Directory

If the file is found, the directory in which the file was found is returned.

Keywords

REPORT

Set this keyword to print a summary report in the main window.

Example

```
RT_Find_File, 'people.dat', /REPORT
```

See Also

[RT_File_OK](#), [RT_Parse_Filepath](#)

For more information, see FILEPATH and FINDFILE in the *IDL Reference Guide*.

RT_Fit_Line

The RT_Fit_Line procedure uses least-squares regression to fit a line to a set of input data.

Syntax

```
RT_Fit_Line, X, Y [, Slope] [, R] [, /INSET] [, /LINE] [, /PLOT]  
[, WT_TYPE=value] [, WEIGHTS=array]
```

Arguments

X

A 1D array of data point x values.

Y

A 1D array of data point y values.

Slope

The slope of the best-fit line.

R

The correlation coefficient of the best-fit line.

Keywords

INSET

Set this keyword to include Slope and R in the plot as an inset.

LINE

Set this keyword to plot the best-fit line in addition to the data points.

PLOT

Set this keyword to create a plot of the data points.

WT_TYPE

A value of 0 indicates that each data point should be weighted by the weights specified with the WEIGHTS keyword. A value of 1 indicates the reverse of the

weights given by the WEIGHTS keyword. A value of 2 indicates that each data point should be given equal weight.

WEIGHTS

A 1D array containing the same number of elements as X and Y and indicating how much weight to give to each data point.

Example

```
X = indgen(10)
Y = X + randomu(seed,10)
RT_Fit_Line, X, Y, /PLOT, /LINE
```

RT_Flow_Path_Coords

The RT_Flow_Path_Coords function uses a D8 flow grid to compute the coordinates (and long-integer IDs) of pixels that lie on the flow path which starts with the pixel whose ID is Start_ID. The returned array has dimensions (2, nIDs), where nIDs is the number of pixels on the flow path. The x and y-coordinates of the pixels are given by $x=coords(0,*)$ and $y=coords(1,*)$. Georeferencing information in the RTI file for the current data set is used to compute the Geographic or UTM coordinates of pixels on the flow path.

Syntax

```
Coords = RT_Flow_Path_Coords(Start_ID [, Stop_ID] [, FILE=filename]
[, MAP=function] [, NIDS=value] [, PATH_IDS=value] [, UNIT=value]
[, /VERBOSE])
```

Arguments

Start_ID

A long-integer ID of a pixel in the current DEM, which marks the upstream end of the flow path.

Stop_ID

A long-integer ID of a pixel in the current DEM, which marks the downstream end (if known) of the flow path. This argument is optional and has no effect unless Stop_ID is found among the PATH_IDS.

Keywords

FILE

A RiverTools D8 flow grid file (*_flow.rtg). If unspecified, the D8 flow grid for the current data set is used.

MAP

A “mapping function” that can be used to quickly determine the IDs of downstream “parent” pixels for a set of pixels, from their own IDs and flow codes. That is, if dID is the flow code of the pixel given by ID, then the ID of the pixel it flows to is:
 $pID = ID + map(dID)$.

NIDS

The number of pixels in the flow path.

PATH_IDS

The long-integer IDs of the pixels on the flow path.

UNIT

A file unit number for a D8 flow grid that is already open for reading. This keyword is useful when computing the flow paths for a large number of Start_IDs, because it prevents opening and closing the D8 flow grid file each time.

VERBOSE

Set this keyword to enable information messages.

RT_Get_Array

The RT_Get_Array procedure returns a one or two-dimensional array with the specified dimensions and data type. This procedure is similar to IDL's MAKE_ARRAY procedure, but differs in the manner in which the data type is specified.

Syntax

```
RT_Get_Array, Array, Ncols [, Nrows] [, TYPE=value] [, /NOZERO]
```

Arguments

Array

The new array to be returned.

Ncols

The number of columns in the new array.

Nrows

The number of rows in the new array. If unspecified, a value of 1 is used.

Keywords

TYPE

The data type of the new array, as a string, which may be: 'BYTE', 'INTEGER', 'LONG', 'FLOAT', or 'DOUBLE'. If unspecified, the data type of the current DEM is used.

NOZERO

Set this keyword to prevent the array from being initialized to contain all zeros. If all values in the array will be set or overwritten later, then this keyword can be set and will allow the array to be created more quickly.

See Also

MAKE_ARRAY in the *IDL Reference Guide*.

RT_Get_Available_DEMs

The `RT_Get_Available_DEMs` procedure returns a sorted list of all files in the working directory that appear to be valid DEMs (based on their file extension) with the same dimensions as the current DEM. This procedure is used by the **Display** → **Density Plot** dialog.

Syntax

```
RT_Get_Available_DEMs, DEM_Files [, /FULL_PATH]
```

Arguments

DEM_Files

The returned string array containing a list of binary grid files in the working directory.

Keywords

FULL_PATH

Set this keyword to return the fully-qualified filepath for each file.

See Also

[RT_Get_Available_Grids](#), [RT_Get_Available_Masks](#)

RT_Get_Available_Grids

The `RT_Get_Available_Grids` procedure returns a sorted list of all files in the working directory that appear to be valid binary grids with the same dimensions as the current DEM. This procedure is used by the **Display** → **Density Plot** dialog.

Syntax

```
RT_Get_Available_Grids, Grid_Files [, /FULL_PATH]
```

Arguments

Grid_Files

The returned string array containing a list of binary grid files in the working directory.

Keywords

FULL_PATH

Set this keyword to return the fully-qualified filepath for each file.

See Also

[RT_Get_Available_Masks](#), [RT_Get_Available_DEMs](#)

RT_Get_Available_Masks

The `RT_Get_Available_Masks` procedure returns a sorted list of all files in the working directory that appear to be valid RiverTools mask (RTM) files that index the current DEM. This procedure is used by the **Display** → **Masked Region** dialog.

Syntax

```
RT_Get_Available_Masks, RTM_Files [, /FULL_PATH]
```

Arguments

RTM_Files

The returned string array containing a list of RiverTools mask (RTM) files in the working directory.

Keywords

FULL_PATH

Set this keyword to return the fully-qualified filepath for each file.

See Also

[RT_Get_Available_DEMs](#), [RT_Get_Available_Grids](#)

RT_Get_Binomial_Cascade

The RT_Get_Binomial_Cascade procedure contains a list of values corresponding to the values at level n in a binomial cascade with parameter p . A binomial cascade is the simplest example of a multifractal measure, and demonstrates a mass distribution that is self-similar at all scales. The width function associated with certain self-similar tree graphs is identical to a binomial (or multinomial) cascade.

Another interesting aspect of binomial cascades is that they can be viewed as “maximally unsorted” lists of numbers. The QuickSort sorting algorithm attains its worst case performance (n^2) when applied to a binomial cascade.

Syntax

```
RT_Get_Binomial_Cascade [, P] [, Nlevs] [, LIST=value] [, LENGTH=value]  
[, /NO_PLOT] [, /REPORT]
```

Arguments

P

A value between 0 and 1 that defines the cascade. If unspecified, $P=0.3$.

Nlevs

The number of levels to generate in the cascade before stopping. If unspecified, $Nlevs = 6$.

Keywords

LIST

The list of values in the cascade.

LENGTH

The number of elements in LIST, which equals 2^{Nlevs} .

NO_PLOT

Set this keyword to suppress a plot of the cascade values.

REPORT

Set this keyword to print a summary report in the main window.

RT_Get_Byte_Order

The `RT_Get_Byte_Order` procedure returns the byte order of the user's computer as a string.

Syntax

`RT_Get_Byte_Order, Byte_Order`

Arguments

Byte_Order

The byte order of the current computer, as a string. The string will be 'LSB' (Least Significant Byte first) for little-endian and 'MSB' (Most Significant Byte first) for big-endian.

See Also

[RT_Not_Same_Order](#)

RT_Get_CDF

The RT_Get_CDF procedure computes the empirical cumulative distribution function (ECDF) for a set of measurements or data values.

Syntax

```
RT_Get_CDF, Values, X,Y[, Mean] [, Subs] [, /PLOT] [, /NORMALIZE]
```

Arguments

Values

The input data values as a 1D array.

X

The output x-coordinates in the ECDF.

Y

The output y-coordinates in the ECDF.

Mean

The mean or sample average of the data.

Subs

The sorted array subscripts of data; s=sort(data).

Keywords

PLOT

Set this keyword to PLOT the ECDF in an IDL window.

NORMALIZE

Set this keyword to divide the data by its mean before computing the ECDF.

RT_Get_Convex_Hull

The `RT_Get_Convex_Hull` procedure uses the “gift-wrapping” algorithm to compute the 2D convex hull of a set of points whose coordinates are given by the 1D arrays `X` and `Y`. The number of points must be greater than or equal to two. Hull points are returned in counter-clockwise order. This procedure runs in $O(n h)$ time, where n is the number of points and h is the number of points on the convex hull. The case of co-linear points/pixels on the hull is also handled correctly.

Syntax

```
RT_Get_Convex_Hull, Hull, X, Y [, /SILENT]
```

Arguments

Hull

A 2D array with dimensions $(2,n+1)$, where n is the number of points. The coordinates of the points on the hull are given by `x_hull=hull(0,*)` and `y_hull=hull(1,*)`.

X

A 1D array of x-coordinates for a set of points.

Y

A 1D array of y-coordinates for a set of points.

Keywords

SILENT

Set this keyword to suppress all information messages.

Example

Generate a random set of 20 points, compute the convex hull and plot it.

```
IDL> x= randomu(1, 20)
IDL> y= randomu(2, 20)
IDL> RT_Get_Convex_Hull, hull, x, y
IDL> plot, x, y, psym=1
IDL> oplot, hull(0,*), hull(1,*), psym=-1
```

RT_Get_File_Where

The `RT_Get_File_Where` procedure returns the subscripts in an array that match a given expression. It is similar to the IDL function `WHERE`, except that it can be applied to large binary files that contain arrays without reading the entire file into RAM. This can be useful for processing files that are larger than the available RAM.

Syntax

```
RT_Get_File_Where, Filename, Type, Expression, Indices [, Count] [, /SILENT]
```

Arguments

Filename

The input filename in single quotes.

Type

A data type string, which can be: 'BYTE', 'INTEGER', 'LONG', 'FLOAT', or 'DOUBLE'.

Expression

A search expression as a string, such as 'LT 0' or 'EQ -1L'.

Indices

The indices to the array of the specified type, contained in `Filename`, that match the search expression.

Count

The number of elements in `Indices`.

Keywords

SILENT

Set this keyword to suppress all information messages.

RT_Get_Flow_Codes

The `RT_Get_Flow_Codes` procedure returns the “flow codes” that correspond to the eight possible flow directions in a D8 flow grid.

Syntax

```
RT_Get_Flow_Codes, Codes [, Map] [, Incs] [, ANGLE_MAP=variable]
[, NCOLS=value] [, OPPS=value]
```

Arguments

Codes

An eight-element integer array containing the D8 flow codes, in clockwise order, starting with the code for Northeast.

Map

A “mapping function” that can be used to quickly determine the IDs of downstream “parent” pixels for a set of pixels, from their own IDs and flow codes. That is, if `dID` is the flow code of the pixel given by `ID`, then the ID of the pixel it flows to is: `pID = ID + map(dID)`.

Incs

Increments, that when added to the long-integer ID of given pixel, give the long-integer IDs of that pixel’s eight neighbor pixels, in clockwise order starting from Northeast.

Keywords

ANGLE_MAP

Set this keyword to a named variable to return a “map” that can be used to quickly convert an array of D8 flow codes to their equivalent continuous flow angles. Flow angles are measured in radians counter-clockwise from due east, and range from 0 to 2 Pi.

NCOLS

The number of columns in the D8 flow grid, which is used to compute `Incs`. If unspecified, the number of columns in the current DEM is used.

OPPS

The flow codes that are the “opposites” of the D8 flow codes. This is the same as a clockwise listing of the flow codes beginning with Southwest.

RT_Get_Kid_ID

The `RT_Get_Kid_ID` procedure computes the long-integer ID of a pixel from its flow code and the ID of its parent pixel. A pixel's parent pixel is the downstream pixel that it sends flow to based on D8 flow directions. The arguments may be arrays.

Syntax

`RT_Get_Kid_ID, ID, pID, Code`

Arguments

ID

The long-integer ID of a pixel in the current DEM to be returned.

pID

The long-integer ID of a pixel in the current DEM that is the parent pixel for the pixel whose ID is to be returned.

Code

The one-byte RiverTools flow code of the pixel whose ID is to be returned.

See Also

[RT_Get_Flow_Codes](#), [RT_Get_Parent_ID](#)

RT_Get_Parent_ID

The RT_Get_Parent_ID procedure computes the long-integer ID of the pixel which is the parent to the pixel whose long-integer ID and flow code are specified. A pixel's parent pixel is the downstream pixel that it sends flow to based on D8 flow directions. The arguments may be arrays.

Syntax

RT_Get_Parent_ID, *pID*, *ID*, *Code*

Arguments

pID

The long-integer ID of the parent pixel in the current DEM to be returned.

ID

The long-integer ID of a pixel in the current DEM whose parent pixel ID is to be returned.

Code

The one-byte RiverTools flow code of the pixel in the current DEM whose parent pixel ID is to be returned.

See Also

[RT_Get_Flow_Codes](#), [RT_Get_Kid_ID](#)

RT_Get_Pixel_Coords

The RT_Get_Pixel_Coords procedure computes the Geographic or UTM coordinates of a set of pixels from their DEM column and row coordinates.

Syntax

```
RT_Get_Pixel_Coords, Col, Row, X, Y [, Xstr] [, Ystr] [, /DMS] [, /GEO] [, /UTM]  
[, RTI_FILE=filename]
```

Arguments

Col

The column-coordinates of a set of pixels in a DEM.

Row

The row-coordinates of a set of pixels in a DEM.

X

The returned x-coordinates of the set of pixels.

Y

The returned y-coordinates of the set of pixels.

Xstr

The x-coordinates of the pixel(s) as a string or string array.

Ystr

The y-coordinates of the pixel(s) as a string or string array.

Keywords

DMS

Set this keyword to make Xstr and Ystr report the coordinates in a Degrees-Minutes-Seconds format.

GEO

Set this keyword to return the Geographic or lon/lat coordinates of the pixel(s).

RTI_FILE

Set this keyword to the name of a RiverTools information (RTI) file that describes the DEM which contains the input pixel. If unspecified, the RTI file for the current DEM is used.

UTM

Set this keyword to return the UTM coordinates of the pixel(s).

See Also

[RT_Get_Pixel_Sizes](#)

RT_Get_Pixel_Sizes

The RT_Get_Pixel_Sizes procedure returns four 1D arrays that gives the “sizes” of all pixels in the DEM as a function of which row they are in. For DEMs whose pixel geometry and bounding box are given by Geographic coordinates, the dependence on latitude and the ellipsoid specified in **File** → **Set Preferences** is taken into account. This routine is used throughout RiverTools in order to accurately compute lengths, slopes and basin areas.

Tip

Pixel sizes are also reported by the Info menu of the Value Zoom dialog.

Syntax

```
RT_Get_Pixel_Sizes, dx, dy [, dd] [, da] [, /METERS] [, /REPORT]  
[, RTI_FILE=filename]
```

Arguments

dx

A 1D array of pixel x-sizes, in kilometers unless the METERS keyword is set.

dy

A 1D array of pixel y-sizes, in kilometers unless the METERS keyword is set.

dd

A 1D array of pixel diagonal size, in kilometers unless the METERS keyword is set.

da

A 1D array of pixel areas, in square kilometers unless the METERS keyword is set.

Keywords

METERS

Set this keyword to report pixel sizes in units of meters instead of kilometers.

REPORT

Set this keyword to print a summary report in the main window.

RTI_FILE

Set this keyword to the name of a RiverTools information (RTI) file. If unspecified, then the RTI file for the current DEM is used.

RT_Get_RTI_Filename

The `RT_Get_RTI_Filename` procedure is used to determine the name of the RTI file that describes the DEM contained in `DEM_File`. If the `DEM_File` argument contains full path information, this path is inherited by `RTI_File`.

Syntax

```
RT_Get_RTI_Filename, DEM_File, RTI_File
```

Arguments

DEM_File

The filename of a binary grid file that contains a DEM.

RTI_File

The returned filename of the RTI file that contains georeferencing information for the specified DEM grid file.

Example

```
RT_Get_RTI_Filename, 'Beaver_DEM.rtg', RTI_file  
print, RTI_file
```

This prints:

```
Beaver.rti
```

See Also

[RT_Get_RTI_Record](#), [RT_Make_RTI_File](#)

RT_Get_RTI_Record

The RT_Get_RTI_Record procedure returns an empty structure that can be filled with georeferencing information and then used to create a RiverTools information (RTI) file.

Syntax

RT_Get_RTI_Record, *Info*

Arguments

Info

A structure with the following fields:

Field	Description
DEM_File	The DEM grid file that this info describes.
Info_File	The name of the RiverTools information (RTI) file.
Data_Source	The source of the DEM data, if known.
Ncols	The number of columns in the DEM.
Nrows	The number of rows in the DEM.
Data_Type	The data type of the DEM as a string array, which can be: <ul style="list-style-type: none"> • 'BYTE' • 'INTEGER' • 'LONG' • 'FLOAT' • 'DOUBLE'
Byte_Order	The byte order of the DEM as a string array, which can be: <ul style="list-style-type: none"> • 'LSB' • 'MSB' <p>Note - LSB = little-endian = host order, MSB=big-endian = networkorder.</p>

Table 1: Info Fields for RT_Get_RTI_Record

Field	Description
Pixel_Geom	A pixel geometry code for the DEM, which is 0b for fixed-angle (Geographic, lon/lat) and 1b for fixed-length (UTM) pixel geometry.
Xres	The x-size of a pixel, which must be in arcseconds for fixed-angle pixel geometry and in meters for fixed-length pixel geometry.
Yres	The y-size of a pixel, which must be in arcseconds for fixed-angle pixel geometry and in meters for fixed-length pixel geometry.
Zres	The vertical resolution of the DEM. This is typically 1 meter or 1 foot when the data type is 'INTEGER' and is currently ignored for other data types.
Z_Units	The elevation units of the DEM as a string, which can be: 'METERS' or 'FEET'.
Y_North_Edge	The coordinates of the north edge of the northern most pixel in the DEM.
Y_South_Edge	The coordinates of the south edge of the southern most pixel in the DEM.
X_West_Edge	The coordinates of the west edge of the western most pixel in the DEM.
X_East_Edge	The coordinates of the east edge of the eastern most pixel in the DEM.
Box_Units	The planimetric or bounding box units, which must be 'DEGREES' for DEMs with fixed-angle pixels and 'METERS' for DEMs with fixed-length pixels.
Emin	The minimum elevation in the DEM.
Emax	The maximum elevation in the DEM.
UTM_Zone	The UTM zone as string that represents an integer between 1 and 60, if applicable.

Table 1: Info Fields for RT_Get_RTI_Record (Continued)

See Also

[RT_Read_DEM_Info](#)

RT_Get_RTM_Cell_Area

The `RT_Get_RTM_Cell_Area` procedure computes the total area spanned by any set of pixels in the current DEM. For DEMs with Geographic (latitude / longitude) coordinates, the variation of pixel dimensions with latitude is taken into account.

Syntax

```
RT_Get_RTM_Cell_Area, IDs, Area [, da]
```

Arguments

IDs

A 1D array of long-integer IDs for a set of pixels in the current DEM.

Area

The total area, in square kilometers, spanned by the set of pixels.

da

A 1D array of pixel areas returned from the `RT_Get_Pixel_Sizes` procedure. If not provided, it is recomputed each time this procedure is called. It should be provided when this procedure is to be called repeatedly, as in a loop.

Example

```
IDL> RT_Read_DEM, DEM
IDL> IDs= where(DEM lt 300, n)
IDL> if (n ne 0) then RT_Get_RTM_Cell_Area, IDs, Area
IDL> print, Area
```

See Also

[RT_Get_Pixel_Sizes](#)

RT_Get_RTM_Info

The `RT_Get_RTM_Info` returns a structure that contains descriptive information about the specified RiverTools mask (RTM) file. An RTM file contains long-integer IDs that identify pixels in an associated DEM. Pixels can be grouped into “mask cells”, such as the pixels in a particular basin, by using `-1L` as a delimiter between sets of pixel IDs. RTM files can also be merged with the **Prepare** → **Merge Files** dialog.

Syntax

`RT_Get_RTM_Info, RTM_File, RTM_Info [, OK=variable]`

Arguments

RTM_File

The filename of a RiverTools mask (RTM) file.

RTM_Info

A structure that contains the following fields:

Field	Description
Maskfile	The name of the RTM file
Filesize	The size of the RTM file, in bytes.
Nlongs	The total number of long integers contained in the file. Some of these will be pixel IDs and some will be delimiters between mask cells.
Numcells	The number of “mask cells” contained in the RTM file.
Subs	The subscripts that correspond to delimiters between mask cells.
Cellsizes	An array that gives the size, in pixels, of all mask cells in the RTM file.

Table 2: RTM_Info Fields

Note

Several additional fields are reserved for internal use and are subject to change.

Keywords**OK**

This keyword returns 1 if the RTM_Info was read from RTM_File without errors and 0 otherwise.

See Also

[RT_Get_RTM_Cell_Area](#)

RT_Get_RTV_Info

The `RT_Get_RTV_Info` returns a structure that contains descriptive information about the specified RiverTools Vector (RTV) file. An RTV file contains an array of records that contain topological information and attributes for channel segments in one or more river networks in the same DEM. These channel segments can be links (`*_links.rtv`), Horton-Strahler streams (`*_streams.rtv`) or “sublinks” that connect adjacent pixels (`*_tree.rtv`).

Syntax

```
RT_Get_RTV_Info, RTV_File, RTV_Info [, /NOCLOSE] [, /REPORT] [, /UPDATE]
```

Arguments

RTV_File

The filename of a RiverTools Vector (RTV) file. This name must end with one of the standard filename extensions “`_tree.rtv`”, “`_links.rtv`”, or “`_streams.rtv`”.

RTV_Info

A structure that contains the following fields:

Field	Description
Filename	The name of the RTV file
Filesize	The size of the RTV file, in bytes.
Filetype	The type of RTV file, as a scalar string which may be ‘TREE’, ‘LINK’, or ‘STREAM’.
Unit	The logical unit number (LUN) for the <code>RTV_File</code> which can be used for further file I/O if the <code>NOCLOSE</code> and/or <code>UPDATE</code> keywords are set.
Blocksize	RTV files are typically processed in blocks whose size is set to be less than the <i>Max RAM block size</i> preference. This field contains the size, in bytes, of a block of RTV records based on the current preference setting.

Table 3: *RTV_Info* Fields

Field	Description
Numblocks	The number of blocks of size <i>Blocksize</i> that are contained in the RTV file. This does not include a possible remainder block. See the descriptions of the <i>Remainder</i> and <i>Kmax</i> fields.
Recsinblock	The number of RTV records (of type ‘TREE’, ‘LINK’, or ‘STREAM’) in a block of size <i>Blocksize</i> .
BPR	The number of Bytes Per Record (of type ‘TREE’, ‘LINK’, or ‘STREAM’).
Remainder	The number of records in a “remainder block” that may be left over after dividing the file into <i>Numblocks</i> blocks of size <i>Blocksize</i> . This number may be zero.
Kmax	The total number of “blocks” in the file, including a possible remainder block.

Table 3: RTV_Info Fields (Continued)

Keywords

NOCLOSE

Set this keyword to leave the file, RTV_File, open for further I/O.

REPORT

Set this keyword to print the information in the RTV_Info record as a short report in the RiverTools output log window.

UPDATE

Set this keyword to leave the file RTV_File open and in “update” mode, as opposed to “read-only” mode. This keyword has no effect unless the NOCLOSE keyword is also set.

See Also

[RT_Get_RTV_Record](#), [RT_Read_RTV_Data](#)

RT_Get_RTV_Record

The `RT_Get_RTV_Record` procedure returns an empty structure that corresponds to a single array element in a RiverTools vector (RTV) file.

Syntax

`RT_Get_RTV_Record, Filetype, Record`

Arguments

Filetype

The RTV filetype as a string, which can be: 'TREE', 'LINK' or 'STREAM'. The possible extensions for an RTV file are '_tree.rtv', '_links.rtv', '_streams.rtv'.

Record

The empty record to be returned, which contains information for a “channel segment.” The channel segments in treefiles, linkfiles and streamfiles are called sublinks, links and streams, respectively. IDs refer to pixels in the DEM, and are assigned in calendar fashion. Additional attributes of channel segments can be computed from these “primitive” attributes. Treefile, linkfile and streamfile records all have the following fields:

Field	Description
PID	The long-integer ID of the pixel at the downstream end of the channel segment.
ID	The long-integer ID of the pixel at the upstream end of the channel segment.
Outdir	The flow code of the pixel whose ID is PID.
Pdist	A special indexing code for the channel segment.
Length	The length of the channel segment in kilometers.
Area	The contributing area upstream of the pixel whose ID is ID, in square km.

Table 4: Record Fields

Field	Description
NkOrder	An encoded form of both the “number of kids” of the pixel whose ID is ID and the Horton-Strahler order.
Drop	The elevation drop between the pixels whose IDs are PID and ID.

Table 4: Record Fields (Continued)

Linkfile and streamfile records have the following additional fields:

Field	Description
Mag	The Shreve magnitude (or number of leaves/sources) of the river tree that is rooted at the downstream node of the channel segment.
LCLen	The length, in kilometers, of the longest channel that drains through the downstream node of the channel segment. This is also the greatest flow distance to any channel source in the river tree that is rooted at the downstream node.
Diam	The network diameter of the river tree that is rooted at the downstream end of the channel segment. Network diameter is greatest flow distance, measured in links, to any channel source in the river tree that is rooted at the downstream node.
Tlen	The total length, in kilometers, of all channels in the river tree that is rooted at the downstream end of the channel segment.
Relief	The elevation difference, in kilometers, between the highest pixel in the subbasin that drains through the downstream node of the channel segment and the elevation of this downstream node.

Table 5: Addition Record Fields for Linkfile and Streamfile

Streamfiles have the following additional fields:

Field	Description
Nlinks	The number of links in the Horton-Strahler stream.
T	A three-element byte array that gives the number of side tributaries of 1, 2, and 3 orders lower than the streams own Horton-Strahler order. Values must be between 0 and 255.

Table 6: Additional Fields for Streamfiles

See Also

[RT_Get_RTV_Record](#), [RT_Get_RTV_Info](#)

RT_Get_SST_Info

The RT_Get_SST_Info procedure returns information about a self-similar tree (SST) or “fractal tree” graph, given its generators. Several well-known SSTs can be selected by setting a keyword. A summary report with information about the specified SST is printed to the main window, such as the stream numbers, number of links per stream and magnitude, as functions of Horton-Strahler order. See the references in the Recommended Reading section of Chapter 1 for more information about the self-similar tree model of river networks.

Syntax

```
RT_Get_SST_Info [, GENERATORS=array] [, BN=value] [, OMAX=value]
[, OUTFILE=filename] [, /SHREVE | /TOKUNAGA | /PERFECT | /HORTON |
/MANDELBROT | /LINEAR | /PEANO | /RRSST]
```

Keywords

GENERATORS

A 1D array of SST generators. The k th element in this array, starting at 1, gives the number of Horton-Strahler streams of order $(w-k)$ that are side tributaries to any stream of order (w) . This keyword need not be specified if any of the following keywords are set: HORTON, LINEAR, MANDELBROT, PEANO, PERFECT, RRSST, SHREVE or TOKUNAGA.

BN

The branching number of the self-similar tree. The default value is 2.

OMAX

The maximum Horton-Strahler order to consider. If the GENERATORS keyword is set, the number of elements in the array is used. Otherwise, the default value is 8.

OUTFILE

Set this keyword to the name of a new text file to which the report should be saved.

HORTON

Set this keyword to use the generators for a Hortonian tree, which has $T(1)=c$ and $T(k)=0$ for all k greater than 1. The user is prompted for the value of c .

LINEAR

Set this keyword to use “linear” generators of the form $T(k)=a * k$. The user is prompted for the value of a.

MANDELBROT

Set this keyword to use the generators for the “Mandelbrot-Vicsek” tree, which has $T(1)=0$ and $T(k)=2^{(k-2)}$.

PEANO

Set this keyword to use the generators for the Peano tree, which has $T(1)=0$, $T(k)=2^{(k-1)}$ and a branching number of 3.

PERFECT

Set this keyword to use the generators for the degenerate case of a “perfect” tree, which has $T(k)=0$ for all values of k.

RRSST

Set this keyword to use the generators for a “recursive replacement” tree, which has $T(1)=(b-1)*(n-1)$ and $T(k)=(b-1) * (n^2) * (n+1)^{(k-2)}$, where b is the branching number and n is the number of junctions in the replacement pattern.

SHREVE

Set this keyword to use the generators for the “average Shreve tree”, which has $T(k)=2^{(k-1)}$.

TOKUNAGA

Set this keyword to use the generators for a “Tokunaga tree,” which has $T(k)=a * c^{(k-1)}$. The user is prompted for the values of the two parameters a and c. Tokunaga trees have been shown to be good models for the average side tributary structure of real river networks.

See Also

[RT_Plot_SST](#)

RT_GUI_Replace_String

The RT_GUI_Replace_String procedure launches a special “search and replace” dialog that can make replacements on an entire set of text files that reside in a single directory. Press the Help button in the dialog for more information.

Since the search and replace operation can sometimes result in changes that were unanticipated by the user, this dialog has several “safety” features, such as the ability to create a report of all changes that were made and the ability to retain the original set of files while making changes only in copies of the original files. However, this is a powerful and potentially dangerous tool that should be used with care. It is not yet available in the main graphical user interface.

Syntax

```
RT_GUI_Replace_String [, Leader]
```

Arguments

Leader

An optional long-integer ID of another top-level base that will serve as the “group leader” for this new dialog. If the group leader is destroyed, then this dialog will also be destroyed.

RT_GUI_View_DEM_Info

The RT_GUI_View_DEM_Info procedure opens a dialog that allows the georeferencing information in a RiverTools information (RTI) file to be viewed and modified. This dialog is available via **File** → **View DEM Info** from within RiverTools, but it is sometimes useful to call it from other program modules.

Syntax

```
RT_GUI_View_DEM_Info [, Leader] [, /MODAL] [, RTI_FILE=filename]
```

Arguments

Leader

The long-integer ID of another top-level base that will serve as the “group leader” for this new dialog. If the group leader is destroyed, then this dialog will also be destroyed.

Keywords

RTI_FILE

Set this keyword to the RiverTools information (RTI) file that should be read and displayed. If unspecified, the RTI file for the current DEM is used.

MODAL

Set this keyword to make the dialog modal, so that all other program execution is halted until the dialog is dismissed.

See Also

[RT_Make_RTI_File](#), [RT_Get_RTI_Record](#)

RT_GUI_View_Text_File

The `RT_GUI_View_Text_File` procedure opens a dialog that allows a text file to be viewed and searched. This dialog is available via **File** → **View Text File** from within RiverTools, but it is often useful to call it from other program modules.

Syntax

```
RT_GUI_View_Text_File, Textfile [, FILTER=string] [, FONT=string]  
[, FW_FONT=string] [, LEADER=value] [, LINES=value]  
[, PER_PAGE_LINES=value] [, /PICKFILE] [, TITLE=string] [, WIDTH=value]
```

Arguments

Textfile

The name of the text file to be viewed.

Keywords

FILTER

If the `PICKFILE` keyword is set, then this keyword can be set to a string with a filename filter, such as `*.txt`, so that only matching files can be selected with the native file selection dialog.

FONT

A string that specifies the font to be used. Available fonts can be obtained as a string array with the IDL command: `DEVICE, GET_FONTNAMES=fontnames`

FW_FONT

Set this keyword to use an available fixed-width font, such as Courier. This is useful when displaying tabular data, because every character in a fixed-width font has the same on screen width.

LEADER

The long-integer ID of another top-level base that will serve as the “group leader” for this new dialog. If the group leader is destroyed, then this dialog will also be destroyed.

LINES

The number of visible lines of text in the dialog. The default is 20.

PER_PAGE_LINES

The number of lines of text to display on each page of text. The default is 42.

PICKFILE

Set this keyword to indicate that the native file selection dialog should be used to obtain the Textfile argument.

TITLE

A string which provides a title for the dialog's title bar. The default includes the name of the textfile being viewed.

WIDTH

The width of the dialog, in text characters. The default is 71.

See Also

[RT_Read_Text_File](#)

RT_Heap_Sort

The `RT_Heap_Sort` function sorts an array of numbers via the well-known “heap sort” algorithm. For large arrays this is sometimes faster (and sometimes much faster) than the QuickSort algorithm used by the IDL function called `SORT`, because the worst case speed of QuickSort goes as $O(n^2)$, while the worst case speed for a heap sort is $O(n * \log(n))$, where n is the number of elements in the array. The sorted array can be obtained as: `New_Array = Array(RT_Heap_Sort(Array))`.

Syntax

```
Sorted_Indices = RT_Heap_Sort(Array)
```

Arguments

Array

The array of values to be sorted.

Example

This example may take several minutes or more, but it demonstrates the point. The number of elements in a binomial cascade grows exponentially with n and is 2^n at level n . As the value of n is gradually increased from 16 to 20 or more, the `SORT` function eventually takes much longer than the `RT_Heap_Sort` function. The IDL function called `SYSTIME` can be used to run time trials.

```
IDL> n = 16
IDL> RT_Get_Binomial_Cascade, 0.3, n, LIST=list, /NO_PLOT
IDL> print, n_elements(list)
IDL> s1 = sort(list)
IDL> s2 = RT_Heap_Sort(list)
```

See Also

For more information, see `SORT` and `SYSTIME`. in the *IDL Reference Guide*.

RT_Init_RTG_File

The `RT_Init_RTG_File` procedure creates a new RiverTools grid (RTG) file and initializes all grid elements to either zero or a user-specified value.

Syntax

```
RT_Init_RTG_File, RTG_File [, Type] [, START_VALUE=value]
```

Arguments

RTG_File

The name of the RTG_File to be created and initialized. No warnings are issued if the file already exists, but the [RT_Check_Overwrite](#) procedure can be used.

Type

The data type of the new RTG file as a string, which may be: 'BYTE', 'INTEGER', 'LONG', 'FLOAT' or 'DOUBLE'.

Keywords

START_VALUE

An initial value for the RTG file. The default is zero.

Example

```
IDL> RTG_File = 'New_Grid.rtg'  
IDL> RT_Check_Overwrite, RTG_File, OK  
IDL> if (OK) then RT_Init_RTG_File, RTG_File, 'FLOAT', START=5
```

See Also

[RT_Get_Array](#)

RT_Init_RTM_File

The `RT_Init_RTM_File` procedure can be used to create and initialize a new RiverTools mask (RTM) file.

Syntax

```
RT_Init_RTM_File, RTM_File, UNIT, Cellsizes, Ncells [/CLOSE] [, /REPORT]
```

Arguments

RTM_File

The filename of the new RTM file to be created and initialized, in single quotes.

UNIT

This argument returns the file unit associated with the newly created and still open RTM file. The file is not closed on exit unless the `CLOSE` keyword is set.

Cellsizes

A long-integer array that indicates the number of pixels in each RTM mask cell that is to be written to the file. The number of elements in the array must match the number of mask cells, `Ncells`, to be written to the file.

Ncells

The number of RTM mask cells to be written to the file.

Keywords

CLOSE

Unless this keyword is set, the new RTM file is left open so that mask cell data can be written to it.

REPORT

Set this keyword to print a summary report in the main window.

See Also

[RT_Read_RTM](#)

RT_Make_Filled_DEM

The RT_Make_Filled_DEM procedure is used to create an RTG file containing a depressionless DEM (in which all depressions have been filled) from an input RTG file containing a DEM. The depression-filling algorithm is very robust and can handle very large DEMs and special cases such as nested depressions and islands. In addition to filling the depressions in the DEM, RiverTools mask (RTM) files are created for (1) the single-pixel pits (*_sp_pits.rtm), (2) the “minima” pixels at the bottom of depressions (*_mins.rtm), and (3) the pixels in each depression that must be raised to create a depressionless DEM (*.pits.rtm). These RTM files can be selected and viewed with the **Display** → **Masked Region** dialog. Information messages generated during processing are written to a log file whose name begins with the data set prefix and ends with “.log”.

Syntax

```
RT_Make_Filled_DEM [, DEM_FILE=filename]
```

Keywords

DEM_FILE

The name of a RiverTools grid (RTG) file that contains a DEM with depressions that are to be filled. If unspecified, the current DEM is used, which typically has a name that ends in “_DEM.rtg”.

RT_Make_Flow_Grid

The RT_Make_Flow_Grid procedure is used to create a RiverTools D8 flow grid for a DEM in which depressions have been filled.

Syntax

```
RT_Make_Flow_Grid, Flow_Grid_File [, /LINK_FLATS] [, /IMPOSED]  
[, /REFINE]
```

Arguments

Flow_Grid_File

The name of the D8 flow grid file to be created. This filename should typically be of the form “<prefix>_flow.rtg”, where the prefix matches the DEM or data set prefix.

Keywords

LINK_FLATS

Set this keyword to assign flow directions within flats via “iterative linking.” This option is selected by default unless the IMPOSED keyword is set.

IMPOSED

Set this keyword to assign flow directions within flats by imposing artificial topography within each flat via the algorithm of Garbrecht and Martz (1997).

REFINE

Set this keyword to refine flow directions within flats by redirecting flow toward any neighbor pixel with the same elevation that has a larger contributing area.

RT_Make_River_Files

The RT_Make_River_Files procedure uses a RiverTools treefile (*_tree.rtv), a pruning method and a pruning threshold to create a RiverTools linkfile (*_links.rtv) and streamfile (*_streams.rtv). These vector files contain attribute information for every link and Horton-Strahler stream in the river network.

Syntax

```
RT_Make_River_Files, Basin_Prefix, Prune_Method, Threshold  
[, LOGFILE=filename] [, TREEFILE=filename]
```

Arguments

Basin_Prefix

A basin prefix, as a string, to be used in creating the filenames for the new linkfile and streamfile. For example, if the basin prefix is 'Deer1', then the linkfile and streamfile will be named 'Deer1_links.rtv' and 'Deer1_streams.rtv'.

Prune_Method

A string that indicates the pruning method to be used, which may be 'AREA', 'ORDER', or 'CONVEXITY'. Pruning by area is the most common method, but is not well-suited to regions which have a variable drainage density. Pruning by order is a good choice when drainage density is variable, because the order of a stream can only increase when it converges with other similar-sized streams. Pruning by convexity is only a viable option for high-resolution DEMs in which hillslopes are well-resolved. With this option, the inflection point from concave down to concave up is identified separately for every flow path, and only the concave-up portion of flow paths are retained. Additional pruning methods will be added in future releases, since this is still an active area of research.

Threshold

A numerical threshold that must be supplied when the Prune_Method is 'AREA' or 'ORDER'. With the 'AREA' method, this threshold has units of square kilometers.

Keywords

LOGFILE

A text file to which information messages should be written. If unspecified, the default log file, with extension ‘*.log’ is used. If the log file already exists, messages are appended at the end.

TREEFILE

A RiverTools treefile which is to be “pruned” to create the new linkfile and streamfile. If unspecified, the default filename is determined from the current basin prefix and the extension “*_tree.rtv”.

RT_Make_RTI_File

The `RT_Make_RTI_File` procedure is used to create a new RiverTools Information (RTI) file.

Syntax

`RT_Make_RTI_File, RTI_File, Info`

Arguments

RTI_File

The filename for the new RTI file, which may include a full path.

Info

A structure which contains georeferencing information for a DEM which is to be saved into the new RTI file. An empty structure of this type can be obtained with the `RT_Get_RTI_Record` procedure. See the documentation for that procedure for a description of the fields in an RTI record.

See Also

[RT_Get_RTI_Record](#), [RT_GUI_View_DEM_Info](#)

RT_Make_Theoretical_DEM

The RT_Make_Theoretical_DEM procedure creates a DEM with a data type of DOUBLE for any of several theoretical surfaces. These DEMs are often useful for testing new algorithms.

Syntax

```
RT_Make_Theoretical_DEM, DEM_File, RTI_File [, Ncols] [, Nrows] [, Scale_X]
[, Scale_Y] [, DEM_TYPE=string] [, /ZRANGE]
```

Arguments

DEM_File

The filename for the new DEM grid file, which should end in “.rtg”.

RTI_File

The filename for the new RTI file, which should end in “.rti”.

Ncols

The number of columns in the new DEM. The default is 200.

Nrows

The number of rows in the new DEM. The default is 200.

Scale_X

A scale factor for the x-axis. The default is 1.

Scale_Y

A scale factor for the y-axis. The default is 1.

Keywords

DEM_TYPE

The type of theoretical surface used to create the DEM, as a string, which can be: ‘CONE’ (an outward-draining cone), ‘GAUSSIAN’ (a Gaussian “hill”), ‘INVCONE’ (an inverted, inward-draining cone), ‘INVPYRAMID’ (an inverted, 4-sided

pyramid), 'MONKEY' (a monkey saddle), 'PLANE' (an inclined plane), 'PYRAMID' (a 4-sided pyramid) or 'SADDLE' (a saddle). The default is 'CONE'.

ZRANGE

Set this keyword to rescale the range of values in the DEM (i.e. (max - min)) to have a particular value.

RT_Make_Treefile

The RT_Make_Treefile procedure is used to create a RiverTools treefile from a RiverTools D8 flow grid. Note that a D8 flow grid is a raster-formatted file while a treefile is a vector-formatted file.

Syntax

```
RT_Make_Treefile, Treefile [, /EDGES] [, /OUTLET] [, LOGFILE=filename]  
[, /USER]
```

Arguments

Treefile

The name of the treefile to be created. This filename should be of the form “*_tree.rtv”.

Keywords

EDGES

Set this keyword to process all basins that drain to a pixel at the edge of the DEM. This keyword is used for the “All basins draining to edge” option. The edge pixels are determined automatically.

OUTLET

Set this keyword to process only the basin that drains to the outlet pixel that is specified in the basin info file associated with the current basin prefix. A basin info file is a simple text file that is usually created via the **Extract** → **Basin Outlet** dialog.

LOGFILE

This keyword can be used to specify a text file other than the default, to which informational messages should be written. The default has the form <prefix>.log, where prefix is the current data set prefix.

USER

If this keyword is set, the user is prompted to select a text file containing coordinate pairs (one per line, in the DEM’s native coordinates) for a set of pixels in the DEM. All basins that drain to one of the specified pixels is processed.

RT_Not_Same_Order

The `RT_Not_Same_Order` function returns 1 if the byte order of the current computer differs from the byte order recorded in the RiverTools Information (RTI) file for the current data set, and 0 otherwise. This function is typically used with the `SWAP_ENDIAN` keyword to the IDL commands `OPENR` and `OPENW` when reading and writing data to binary files in a given data set. It is used to make sure that the new files have the same byte order as the other binary files in the data set.

Syntax

```
result = RT_Not_Same_Order()
```

Example

Create a simple binary file that has the same byte order as the current data set:

```
IDL> array = indgen(10,10)
IDL> openw, unit, 'TEST.dat', /get_lun, $
    swap_endian=RT_Not_Same_Order()
IDL> writeu, unit, array
IDL> free_lun, unit
```

See Also

[RT_Get_Byte_Order](#)

RT_Open_Plot

The RT_Open_Plot procedure opens a RiverTools plot to be sent to the current output device (e.g. a window, a printer or a PostScript file).

Syntax

```
RT_Open_Plot [, Plot_Type] [, BASEID=variable] [, NPLOTS=value] [, /ORDER]
[, /SCROLL] [, /SILENT] [, /SIZEABLE] [, WINID=variable]
[, WINTITLE=string] [, XOFF=value] [, XSIZE=value] [, YOFF=value]
[, YSIZE=value]
```

Arguments

Plot_Type

This is an optional string argument which can be used to specify the type of RiverTools plot that will be sent to the current device. It is meant to be used internally by RiverTools. Examples include 'CONTOUR', 'DENSITY' and 'RELIEF'.

Keywords

BASEID

If the output device is a window, this keyword can be used to obtain the long-integer ID of the top-level base that contains the window.

NPLOTS

When set to a number greater than one, this keyword allows you to send multiple plots to the output device. The plots are arranged in two columns and sized to fit the available plotting area.

ORDER

If set, this keyword specifies that the plot will be drawn from top to bottom. The default is from bottom to top. It only affects raster plots.

SCROLL

If the output device is a window, setting this keyword adds scroll bars to the window.

SILENT

Set this keyword to suppress all information messages.

SIZEABLE

If the output device is a graphics window, setting this keyword allows the window to be resized with the mouse.

WINID

If the output device is a window, this keyword can be used to obtain the long-integer window ID of the window.

WINTITLE

If the output device is a window, this keyword can be used to specify the title that is to appear in the window's title bar.

XOFF

If the output device is a window, this keyword specifies an x-offset for the window, in pixels from the upper left corner of the screen.

XSIZE

If the output device is a window, this keyword specifies the x-size in pixels. For other devices, the plot is scaled to fit the plotting area, using information from the **File** → **Printing Options** dialog.

YOFF

If the output device is a window, this keyword specifies an y-offset for the window, in pixels from the upper left corner of the screen.

YSIZE

If the output device is a window, this keyword specifies the y-size in pixels. For other devices, the plot is scaled to fit the plotting area, using information from the **File** → **Printing Options** dialog.

RT_Parse_Filepath

The RT_Parse_Filepath procedure parses a filepath to return its components. Note that the character used as a directory delimiter varies across platforms.

Syntax

```
RT_Parse_Filepath, Filepath [, DIRECTORY=directory_name] [, ETYPE=string]  
[, EXTENSION=string] [, FILENAME=filename] [, FULL_PREFIX=string]  
[, PATH_ARRAY=string] [, RT_PREFIX=string] [, /REPORT]  
[, RT_EXTENSION=string]
```

Arguments

Filepath

The filepath, as a string in single-quotes, to an existing file.

Keywords

DIRECTORY

The directory which contains the file indicated by the filepath argument. If the filepath does not contain any directory delimiters, the null string is returned. Otherwise, the last character will always be the native directory delimiter.

ETYPE

The type of filename extension. If the filepath argument has one of the standard RiverTools extensions (.RTG, .RTM, or .RTV) then 'COMPOUND' is returned; otherwise 'SIMPLE' is returned.

EXTENSION

The filename extension, which consists of all characters after and including the last dot in the filepath argument.

FILENAME

The filename, which consists of all characters after the last directory delimiter in the filepath argument.

FULL_PREFIX

The full prefix, which consists of all characters in the filename that precede the last dot in the filepath argument.

PATH_ARRAY

A string array that contains one element for each of the nested directories listed in filepath argument.

REPORT

Set this keyword to print the various filepath components as a short report in the main log window.

RT_EXTENSION

The RiverTools extension, which consists of all characters after and including the last underscore in the filepath argument, but only when the extension is “.RTG”, “.RTM”, or “.RTV”. See the ETYPE keyword.

RT_PREFIX

The RiverTools prefix, which consists of all characters in the filename that precede the last underscore in the filepath argument, but only when the extension is “.RTG”, “.RTM”, or “.RTV”. See the ETYPE keyword.

RT_Plot_SST

The RT_Plot_SST procedure can be used to plot several different self-similar tree graphs for illustrative purposes. Self-similar tree graphs are useful as idealized models of river networks that allow scaling laws and other features to be better understood. The order of the plotted tree and many other attributes can be specified.

Syntax

```
RT_Plot_SST [, ANGLE=value] [, COLOR=value] [, L=array] [, MIX=array]
[, NL=array] [, ORDER=value] [, SIDES=array] [, THICK=value]
[, WINID=variable] [, X0=value] [, Y0=value] [, /SHREVE | /PERFECT |
/HORTON | /MANDELBROT]
```

Keywords

ANGLE

The angle in radians, measured from vertical, of the main stem of the tree. The default is 0.

COLOR

A color index that determines the color of the tree. Color indices for many basic colors can be obtained via the RT_Color_Index function; e.g. COLOR = RT_Color_Index('red').

HORTON

Set this keyword to plot the most basic Hortonian tree, which has SST generators $T(1)=1$ and $T(k)=0$ for all k greater than 1.

L

This keyword can be set to a 1D array that specifies the lengths for streams of each order, in normalized coordinates. The k th element gives the length of a stream of order k . The first (zero) element in the array is unused. The default is to double the length when order increases by one.

MANDELBROT

Set this keyword to plot a Mandelbrot-Vicsek tree, which has SST generators $T(1)=0$ and $T(k)=2^{k-2}$ for all k greater than 1.

MIX

This keyword can be set to an “interspersal array” that determines the order in which side tributaries of lower orders enter the highest order stream. This pattern is repeated for all lower order streams. For example, the order four stream in an order four Shreve tree could have MIX=[3, 2, 3, 1, 3, 2, 3] or MIX=[1,2,2,3,3,3,3] with one side tributary of order 3, two of order 2, and four of order 1.

NL

This keyword can be set to an array that gives the number of links in streams of each order. The zeroth element of the array is not used.

ORDER

The keyword specifies the maximum order of the SST. The default value is 5 and the maximum allowed value is 8.

PERFECT

Set this keyword to plot a “perfect” tree, which has no side tributaries. The SST generators are $T(k)=0$ for all values of k .

SHREVE

Set this keyword to plot an average Shreve tree, which has SST generators $T(k)=2^{(k-1)}$ for all values of k .

SIDES

This keyword can be set to an array that determines whether side tributaries enter from the left or right side. A value of 1 indicates the left side and a value of -1 indicates the right side. The pattern is specified for the highest order stream and is then repeated for all lower order streams. The zeroth element is unused.

THICK

The line thickness to be used for all streams. The default is 1.

WINID

This keyword can be used to obtain the long-integer index/ID of the graphics window.

X0

The x-coordinate of the outlet, in normalized coordinates, measured from the left side of the plotting area. The default is 0.5.

Y0

The y-coordinate of the outlet, in normalized coordinates, measured from the bottom of the plotting area. The default is 0.05.

See Also

[RT_Get_SST_Info](#)

RT_Print

The RT_Print procedure is used to print text to the main RiverTools window.

Syntax

RT_Print, *Message*

Arguments

Message

A scalar string or string array that is to be printed to the main log window. If Message is a string array, then each element is plotted on a separate line.

Example

```
RT_Print, 'Hello world.'  
RT_Print, ['Hello', 'world']
```

RT_Read_DEM

The RT_Read_DEM procedure reads the DEM for the current data set into a 2D array.

Syntax

```
RT_Read_DEM, DEM
```

Arguments

DEM

The returned 2D array containing the elevation values.

See Also

[RT_Read_Grid](#)

RT_Read_DEM_Info

The RT_Read_DEM_Info procedure is used to read georeferencing information for a DEM from an RTI file into a structure variable.

Syntax

```
RT_Read_DEM_Info, Info [, /CHECK] [, NCOLS=variable] [, NROWS=variable]  
[, /REPORT] [, RTI_FILE=filename] [, /SILENT]
```

Arguments

Info

A structure that contains all of the information read from the RTI file. The fields in this structure are described in the [RT_Get_RTI_Record](#) procedure.

Keywords

CHECK

Set this keyword to cause the information in the Info structure to be checked for consistency and for values that are not within allowed ranges. If errors are found, they are reported in a graphical error dialog.

NCOLS

This keyword can be used to obtain the number of columns in the DEM.

NROWS

This keyword can be used to obtain the number of rows in the DEM.

REPORT

Set this keyword to print a summary report of the information in the RTI file to the main log window.

RTI_FILE

This keyword can be set to the name of an RTI file. The default is the RTI file for the currently selected data set.

SILENT

Set this keyword to suppress all information messages.

See Also

[RT_Get_RTI_Record](#), [RT_GUI_View_DEM_Info](#)

RT_Read_Function

The `RT_Read_Function` procedure is used to read one or two columns of data from an ASCII file that contains multiple columns of data into 1D array variables. Lines that do not contain enough “words” to read the specified columns are skipped. Values in a line can be delimited by spaces, commas, semi-colons or colons.

Syntax

```
RT_Read_Function, Filename, X [, Y] [, SKIP_LINES] [, X_Column] [, Y_Column]
```

Arguments

Filename

The filename (in single quotes) of the ASCII file that contains multi-column tabular data.

X

A 1D array of values read from the specified ASCII file.

Y

A 1D array of values read from the specified ASCII file.

SKIP_LINES

The number of header lines to skip before reading data. The default is zero.

X_COLUMN

If specified, this indicates which column to read into the 1D array variable X. Column numbers start with 1 and the default is 1. If `X_COLUMN` is set to 0, then X will contain an array of ascending integers, starting with 0.

Y_COLUMN

If specified, this indicates which column to read into the 1D array variable Y. Column numbers start with 1 and the default is 2.

See Also

[RT_Read_XYZ_Values](#)

RT_Read_Grid

The RT_Read_Grid procedure reads a RiverTools Grid (RTG) file into a 2D IDL array.

Syntax

```
RT_Read_Grid, Grid, RTG_File [, /AUTOSCALE] [, FACTOR=value]  
[, NCOLS=value] [, NROWS=value] [, /REDUCE] [, TYPE=string]
```

Arguments

Grid

The 2D array of grid values to be read and returned.

RTG_File

The filename (in single quotes) of the RTG file to be read.

Keywords

AUTOSCALE

Set this keyword in order to rebin the grid to a reduced or enlarged size while reading it. The scale factor is determined automatically to make the new array size as close as possible to the maximum allowed window size (in the preferences) without exceeding it.

FACTOR

An integer scale factor used to reduce or enlarge the grid while reading it. The grid is reduced by this factor if the REDUCE keyword is set and enlarged otherwise.

NCOLS

The number of columns in the grid. If unspecified, the number of columns in the DEM of the current data set is used. RTG files in the same data set should have the same dimensions. This keyword is for use when reading grids that are not part of the current data set.

NROWS

The number of rows in the grid. If unspecified, the number of rows in the DEM of the current data set is used. RTG files in the same data set should have the same

dimensions. This keyword is for use when reading grids that are not part of the current data set.

REDUCE

Set this keyword to cause the grid to be rebinned to a reduced size as it is read from the RTG grid file. The integer scale factor for rebinning is set via the FACTOR keyword.

TYPE

The data type of the grid, which can be 'BYTE', 'INTEGER', 'LONG', 'FLOAT', or 'DOUBLE'. If unspecified, RiverTools will attempt to determine the data type from the file size and grid dimensions.

See Also

[RT_Read_DEM](#)

RT_Read_Horton_Data

The RT_Read_Horton_Data procedure reads stored statistics for the Horton-Strahler streams in a river network, for a specified attribute. This information is read from a file with extension “_stats.dat”. Units are given in terms of kilometers and square kilometers, except for elevation drop which is given in meters.

Syntax

```
RT_Read_Horton_Data, Data, Attribute [, Statistic] [, /SILENT]
```

Arguments

Data

The 1D array of statistics to be returned, with one element for each Horton-Strahler order in the river network of the current data set.

Attribute

A scalar string that specifies the attribute of interest, in single quotes. Allowed attributes are: ‘AREA’, ‘SLLEN’, ‘ACLEN’, ‘DROP’, ‘SLSLP’, ‘ACSLP’, ‘TLEN’, ‘MAG’, ‘LCLEN’, ‘RELIEF’, ‘DIAM’, ‘SINU’, ‘DDENS’, ‘SDENS’, ‘NLINKS’, ‘T1’, ‘T2’, ‘T3’, ‘ORDER’, ‘SNUMS’. The ‘ORDER’ keyword returns a ramp of stream order numbers, and the ‘SNUMS’ keyword returns the total number of Horton-Strahler streams of each order. The other attributes are: basin area, straight-line length, along-channel length, elevation drop across the segment, straight-line slope, along-channel slope, total channel length, magnitude, longest channel length, basin relief, network diameter, absolute sinuosity, drainage density, source density, number of links, number of side tributaries of one order lower, number of side tributaries of two orders lower, and number of side tributaries of three orders lower.

Statistic

A scalar string that specifies the statistic of interest, in single quotes. Allowed statistics are: ‘MEAN’, ‘MIN’, ‘MAX’, ‘RANGE’, ‘STDEV’, ‘MOM2’, ‘MOM3’, ‘MOM4’, ‘MOM5’, ‘MOM6’. The default is ‘MEAN’. Statistic names that start with ‘MOM’ refer to higher order moments.

Keywords

SILENT

Set this keyword to suppress the information message that is printed in the main log window.

See Also

[RT_Read_RTV_Data](#)

RT_Read_RTM

The RT_Read_RTM procedure reads mask cell IDs from a RiverTools Mask (RTM) file.

Syntax

```
RT_Read_RTM, Mask_IDs, Filename [, FACTOR=value] [, NUMCELLS=variable]
[, OK=variable] [, /REDUCE] [, /SILENT] [, SUBS=variable]
```

Arguments

Mask_IDs

An array of long-integer IDs read from the specified RTM file. This array contains a list of pixel IDs for one or more “mask cells” or masked regions in a RiverTools grid. If the RTM file contains information for more than one mask cell, IDs for separate cells are delimited by the long integer -1L. The first and last ID in the returned array should also be -1L.

FILENAME

The filename (in single quotes) of the RTM file to read IDs from.

Keywords

FACTOR

An integer scale factor which can be used to rescale the Mask_IDs so that they can be used as pixel IDs to subscript a reduced-size grid. The grid is reduced by this factor only if the REDUCE keyword is set. For more information, see [RT_Read_Grid](#).

NUMCELLS

Set this keyword to a named variable in order to obtain the number of mask cells in the specified RTM file.

OK

This keyword will return a value of 1b if no problems are encountered, and 0b otherwise.

REDUCE

See the explanation of the FACTOR keyword above.

SILENT

Set this keyword to suppress information messages sent to the main log window.

SUBS

Set this keyword to a named variable in order to obtain a 1D array which lists the array subscripts in Mask_IDs which contain the -1L delimiter. Since an RTM file begins (and ends) with this delimiter, adding 1 to the nth element in the SUBS array gives the starting subscript for the (n+1)st mask cell. For example, if the RTM file contains pixel IDs for 5 mask cells, SUBS(2)+1 is the subscript to the Mask_IDs array where the IDs for the 3rd mask cell begin.

Example

```
IDL> RT_Read_RTM, mask_IDs, file='KY_Sub_o3basins.rtm'
```

See Also

[RT_Get_RTM_Info](#), [RT_Get_RTM_Cell_Area](#), [RT_Read_Grid](#),
[RT_Read_RTM_As_Grid](#), [RT_Read_RTM_Cell](#)

RT_Read_RTM_As_Grid

The `RT_Read_RTM_As_Grid` procedure reads mask IDs from a RiverTools Mask (RTM) file and then “embeds” them in a mask grid.

Syntax

```
RT_Read_RTM_As_Grid, Mask_Grid, FILENAME=filename [, FACTOR=value]  
[, /REDUCE]
```

Arguments

Mask_Grid

The 2D array to be returned, which will have the same dimensions as the DEM in the current data set unless the `FACTOR` keyword is set.

Keywords

FILENAME

The filename (in single quotes) of the RTM file to read IDs from.

FACTOR

An integer scale factor used to create a grid that is rescaled to be smaller or larger than the DEM in the current data set. The grid is reduced by this factor if the `REDUCE` keyword is set and enlarged otherwise.

REDUCE

See the description of the `FACTOR` keyword.

See Also

[RT_Get_RTM_Cell_Area](#), [RT_Get_RTM_Info](#), [RT_Read_RTM](#),
[RT_Read_RTM_Cell](#)

RT_Read_RTM_Cell

The `RT_Read_RTM_Cell` procedure reads the IDs of a single mask cell from a RiverTools Mask (RTM) file. It is meant to be used together with the [RT_Get_RTM_Info](#) procedure as shown in the example below.

Syntax

```
RT_Read_RTM_Cell, Cell_Number, Cell_IDs, RTM_Info [, /SILENT]  
[, UNIT=value]
```

Arguments

Cell_Number

The number of the RTM cell to be read. Numbers start at 0 for the first cell in the RTM file and go up to the number of cells minus one. An RTM file may contain information for one or more mask cells.

Cell_IDs

The long (4-byte) integer pixel IDs of the pixels in the requested mask cell that are to be returned

RTM_Info

A structure of the form returned by the `RT_Get_RTM_Info` procedure that contains descriptive information for an RTM file, including the filename. This argument is not optional; see the example below.

Keywords

SILENT

Set this keyword to suppress the information message that is usually printed in the main log window.

UNIT

This keyword can be used to specify the file unit number for an RTM file that is already open for reading. If unspecified, the file `RTM_Info.maskfile` is open and closed each time the procedure is called. When reading data for many RTM cells, it is more efficient to leave the file open until finished.

Example

This simple example shows how to compute the average elevation in the sixth order 3 basin that is stored in the RTM file, 'KY_Sub_o3basins.rtm', and then to create a density plot that shows the pixels in the basin as a black shape. Open the KY_Sub sample data set, and then type the following commands at the IDL command line.

```
IDL> RT_Get_RTM_Info, 'KY_Sub_o3basins.rtm', RTM_info
IDL> RT_Read_RTM_Cell, 5, cell_IDs, RTM_info
IDL> RT_Read_Grid, grid, 'KY_Sub_DEM.rtg'
IDL> avg = total(grid(cell_IDs)) / n_elements(cell_IDs)
IDL> RT_Print, 'The average elevation is: ' + RT_String(avg)
IDL> RT_Read_DEM_Info, info
IDL> RT_Open_Plot, XSIZE==info.ncols, YSIZE==info.nrows
IDL> !ORDER=1
IDL> grid(cell_IDs) = 0
IDL> RT_Tvscl, RT_Hist_Equal(grid)
```

See Also

[RT_Get_RTM_Info](#)

RT_Read_RTV_Data

The `RT_Read_RTV_Data` procedure reads attributes of channel segments (sublinks, links or streams) from a RiverTools Vector (RTV) file. Units are given in terms of kilometers and square kilometers, except for elevation drop which is given in meters.

Syntax

```
RT_Read_RTV_Data, Data, Attribute, Filetype [, Orders] [, Filter] [, N]
```

Arguments

Data

The returned 1D array that contains one measurement of the selected attribute for each channel segment.

Attribute

A scalar string that determines which attribute to return in Data. For all FILETYPEs, the following attributes are allowed: 'AREA', 'SLEN', 'ACLEN', 'DROP', 'SLSLP', and 'ACSLP'. These correspond to: basin area, straight-line length, along-channel length, elevation drop across the segment, straight-line slope, and along-channel slope. The default is 'AREA'.

If the FILETYPE is 'LINK' or 'STREAM', these additional attributes are allowed: 'TLEN', 'MAG', 'LLEN', 'RELIEF', 'DIAM', 'SINU', 'DDENS.', and 'SDENS.'. These correspond to: total channel length, magnitude, longest channel length, basin relief, network diameter, absolute sinuosity, drainage density, and source density.

If the FILETYPE is 'STREAM', these additional attributes are recognized: 'NLINKS', 'T1', 'T2', 'T3'. These correspond to: number of links, number of side tributaries of one order lower, number of side tributaries of two orders lower, and number of side tributaries of three orders lower.

Filetype

A scalar string that determines the type of RTV file to be used (given the currently selected basin prefix). The possibilities are 'TREE', 'LINK', and 'STREAM'. The default is 'STREAM'.

Orders

An optionally returned 1D array that contains the Strahler order for each channel segment.

Filter

An optional scalar string that can be used to filter the measurements by Strahler order. Only for channel segments that match the filter are returned in the array Data. Allowed filters are: 'ALL', 'INTERIOR', 'EXTERIOR', 'ORDER1', 'ORDER2', ..., 'ORDER10'. The string 'INTERIOR' indicates all channel segments whose order is not one.

N

The number of elements in the 1D arrays, Data and Orders.

RT_Read_Subgrid

The RT_Read_Subgrid procedure reads a (usually small) subgrid from a grid contained in an RTG file. This procedure is used by several of the interactive window tools, such as the Vector Zoom tool.

Syntax

RT_Read_Subgrid, *Grid_File*, *Grid_Type*, *Grid_Ncols*, *Grid_Nrows*, *X*, *Y*, *NX*, *NY*, *Subgrid* [, *OK*]

Arguments

Grid_File

The filename of an RTG grid file, or any other flat binary (row major) file, which contains a grid of values. This filename should be given in single quotes.

Grid_Type

A scalar string which indicates the data type of the grid stored in Grid_File. This may be 'BYTE', 'INTEGER', 'LONG', 'FLOAT', or 'DOUBLE'.

Grid_Ncols

The number of columns in the Grid_File grid.

Grid_Nrows

The number of columns in the Grid_File grid.

X

The column in the Grid_File grid to use as the center for the read operation

Y

The row in the grid in Grid_File grid to use as the center for the read operation.

NX

The number of columns to read from the Grid_File grid into Subgrid.

NY

The number of rows to read from the Grid_File grid into Subgrid.

Subgrid

The subgrid to be returned, as a 2D array.

OK

This will be set to 1b if no problems are encountered (such as input values that cause an attempt to read out of bounds) and 0b otherwise.

See Also

[RT_Write_Subgrid](#)

RT_Read_Text_File

The RT_Read_Text_File procedure reads the text in the input text file into a string array that has one line per array element.

Syntax

```
RT_Read_Text_File, Text_File, Text_Array [, Nlines] [, MAXLINES=value]
```

Arguments

Text_File

The name of the input text file, in single quotes.

Text_Array

The returned string array which contains the text.

Nlines

This optional argument returns the number of lines in Text_File, which is the number of elements in Text_Array.

Keywords

MAXLINES

This keyword imposes a maximum number of lines that should be read. If not specified, the RT_Count_Lines procedure is used to count the number of lines in the text file before reading and all lines are read.

See Also

[RT_Count_Lines](#), [RT_GUI_View_Text_File](#), [RT_Read_Function](#),
[RT_Read_XYZ_Values](#)

RT_Read_XYZ_Values

The `RT_Read_XYZ_Values` procedure reads XYZ coordinate triples from an input text file into three 1D arrays called X, Y and Z. Values in a line can be delimited by space characters, commas, semi-colons or colons. Lines with less than three values are automatically skipped.

Syntax

```
RT_Read_XYZ_Values, Text_File, X, Y, Z [, Xmin] [, Xmax] [, Ymin] [, Ymax] [, Zmin] [, Zmax] [, SKIP_LINES=value]
```

Arguments

Text_File

The name of the input text file, in single quotes.

X

A 1D array of x-coordinates.

Y

A 1D array of y-coordinates.

Z

A 1D array of z-coordinates.

Xmin

The minimum x-coordinate in X.

Xmax

The maximum x-coordinate in X.

Ymin

The minimum y-coordinate in Y.

Ymax

The maximum y-coordinate in Y.

Zmin

The minimum z-coordinate in Z.

Zmax

The maximum z-coordinate in Z.

Keywords**SKIP_LINES**

Set this keyword to the number of header lines that should be skipped before reading data. The default is zero.

See Also

[RT_Count_Lines](#), [RT_Read_Function](#), [RT_Read_Text_File](#)

RT_Remove_Window

The `RT_Remove_Window` procedure is used to remove a window from the roster of windows that are managed by RiverTools. To destroy a window, use the [RT_Close_Window](#) procedure, which calls `RT_Remove_Window`. This procedure is meant for use by programmers who want to extend the RiverTools environment.

Syntax

```
RT_Remove_Window, Base_ID
```

Arguments

Base_ID

The top-level base ID, a long (4-byte) integer, of the window that is to be removed from the roster.

See Also

[RT_Add_Window](#), [RT_Check_Window](#), [RT_Close_Window](#), [RT_Open_Plot](#), [RT_Set_Window](#)

RT_Resize

The RT_Resize function is used to rebin a 2D array, by an integer factor, to a larger or smaller size. It is similar to the IDL REBIN function, but it does not require the array dimensions of the new array to be multiples of the old dimensions. It also differs in its arguments, which lets you specify a scale factor instead of the new array size.

Syntax

```
RT_Resize, Array, Factor, Reduce [, New_Ncols] [, New_Nrows]
```

Arguments

Array

The 2D input array to be resized.

Factor

A positive integer that determines the factor by which the array will be reduced or enlarged. It is treated as a reduction factor if Reduce argument equals 1b and a magnification factor if it equals 0b.

Reduce

Set this argument to 1b for reduction and 0b for magnification.

New_Ncols

This argument can be used to return the number of columns in the resized array.

New_Nrows

This argument can be used to return the number of rows in the resized array.

Example

Start RiverTools+IDL and create a 2D array with 101 rows and 103 columns. An attempt to rebin it to half size with the IDL REBIN function produces an error, but the call to RT_Resize does not.

```
IDL> array = dist(101, 103)
IDL> a = REBIN(array, 50, 50)

% REBIN: Result dimensions must be integer factor of
original dimensions

IDL> a = RT_Resize(array, 2, 1b, new_ncols, new_nrows)
IDL> RT_Print, 'New number of columns = ' +
RT_String(new_ncols)
IDL> RT_Print, 'New number of rows = ' +
RT_String(new_nrows)
```

See Also

For more information, see REBIN in the *IDL Reference Guide*.

RT_Save_Function

The RT_Save_Function procedure writes two 1D arrays as two columns to a text file. The values are written as double precision, with a total of 18 characters and 8 digits after the decimal. Function-type (2-column) data saved with this procedure can be plotted in RiverTools with the **Display** → **Function** dialog.

Syntax

```
RT_Save_Function, X, Y [, Text_File] [, HEADER=string] [, /SILENT]
```

Arguments

X

A 1D array of values for the x-axis or abscissa.

Y

A 1D array of values for the y-axis or ordinate. This array must have the same number of elements as the array X.

Text_File

The name of the text file, in single quotes, to which the function values are to be saved. If unspecified, a native file selection dialog is opened to allow the filename to be specified.

Keywords

HEADER

An optional string or string array that is to be written at the top of the file, before the data values. If unspecified, RiverTools provides a simple three line header that lists the number of elements in the X and Y arrays.

SILENT

Set this keyword to suppress printing the information message in the main log window.

See Also

[RT_GUI_View_Text_File](#), [RT_Read_Function](#), [RT_Read_XYZ_Values](#)

RT_Set_View

The `RT_Set_View` procedure is used to set 3D view transformation that is used by IDL direct graphics procedures. It is similar to the IDL procedure `CREATE_VIEW`, but uses the mins and maxes of the `X`, `Y`, and `Z` arguments for a georeferenced surface to determine default scale parameters.

Syntax

```
RT_Set_View, X, Y, Z [, Xwin] [, Ywin] [, AX=value] [, AY=value] [, AZ=value]
[, /RESET] [, ZFAC=value] [, ZOOM=value]
```

Arguments

X

An array of x-coordinates that provide the distances between pixels along the x-axis in the 2D array `Z`. This array allows the surface stored in `Z` to be viewed at true scale.

Y

An array of y-coordinates that provide the distances between pixels along the x-axis in the 2D array `Z`. This array allows the surface stored in `Z` to be viewed at true scale.

Z

An array of z-coordinates that provide the surface height at a pixel or point. This could be a DEM or just a two-element array containing the min and max in the DEM.

Xwin

The x-size of the viewing window.

Ywin

The y-size of the viewing window.

Keywords

AX

An angle of rotation about the x-axis, in degrees.

AY

An angle of rotation about the y-axis, in degrees.

AZ

An angle of rotation about the z-axis, in degrees.

RESET

Set this keyword to reset all affected IDL system variables to their defaults.

ZFAC

An optional rescaling factor for the z-axis used for vertical exaggeration. The default is 1.

ZOOM

A positive factor used to rescale the x, y, and z axes that determines how close the viewed object appears to be from the viewer. The default is 1.

See Also

For more information, see `CREATE_VIEW` in the *IDL Reference Guide*.

RT_Set_Window

The `RT_Set_Window` procedure is used to set the focus to a particular RiverTools window.

Syntax

```
RT_Set_Window, Window_ID [,/SHOW]
```

Arguments

Window_ID

The long (4-byte) integer window index or ID. This index can be obtained when the window is first opened with [RT_Open_Plot](#).

Keywords

SHOW

Set this keyword to cause the window to be brought to the front if it is obscured by other windows.

See Also

[RT_Add_Window](#), [RT_Check_Window](#), [RT_Close_Window](#), [RT_Open_Plot](#), [RT_Remove_Window](#)

RT_String

The `RT_String` function is used to convert numbers to strings which have no leading or trailing blank spaces. It provides a simple and convenient method of printing numbers with formatting as shown in the example below.

Syntax

```
String = RT_String(Number [, FORMAT=value])
```

Arguments

Number

The number or array of numbers to be converted to a string or string array. This can have any data type.

Keywords

FORMAT

A FORTRAN-style format code. See the documentation for the IDL function, `STRING`, for details.

Example

```
IDL> PRINT, 'The number is: ' + RT_String(1000d).
IDL> k = INDGEN(5)
IDL> str = RT_String(k, FORMAT='(D8.2)')
IDL> PRINT, 'The number is: ' + str(3)
```

See Also

[RT_Print](#)

For more information, see `STRING` and `STRTRIM` in the *IDL Reference Guide*.

RT_Uncompress_File

The `RT_Uncompress_File` procedure uncompresses an input file that has been previously compressed with the well-known GZIP algorithm. The original compressed file is not deleted. This procedure could be used in a loop to uncompress a large number of files.

Syntax

`RT_Uncompress_File, In_File, Out_File`

Arguments

In_File

The input file to be uncompressed, which should have the filename extension “.gz”.

Out_File

The uncompressed output file.

See Also

[RT_Compress_File](#), [RT_Copy_File](#), [RT_Delete_File](#)

RT_Write_Subgrid

The RT_Write_Subgrid procedure is used to edit an RTG file by overwriting a (usually small) subgrid of values in the RTG file with the values that are contained in the Subgrid argument. In most cases it is better to edit RTG files with the interactive window tool called Value Zoom.

Syntax

```
RT_Write_Subgrid, Grid_File, Grid_Type, Grid_Ncols, Grid_Nrows, X, Y, Subgrid  
[, OK]
```

Arguments

Grid_File

The filename of the RTG grid file to be modified, or any other flat binary (row major) file, which contains a grid of values. This filename should be given in single quotes.

Grid_Type

A scalar string which indicates the data type of the grid stored in Grid_File. This may be 'BYTE', 'INTEGER', 'LONG', 'FLOAT', or 'DOUBLE'.

Grid_Ncols

The number of columns in the Grid_File grid.

Grid_Nrows

The number of columns in the Grid_File grid.

X

The column in the Grid_File file to use as the center for the write operation

Y

The row in the grid in Grid_File to use as the center for the write operation.

Subgrid

The subgrid (a 2D array) to be used to replace values in the Grid_File grid. Proper placement of the subgrid is easiest when the subgrid is square and has an odd number of rows and columns.

OK

This will be set to 1b if no problems are encountered (such as input values that cause an attempt to write out of bounds) and 0b otherwise.

See Also

[RT_Read_Subgrid](#)