McIDAS-X User's Guide

Version 2012.2

McIDAS User's Guide
Version 2012.2 Table of Contents

Copyright Statement and Trademarks

Chapter 1 - Installing and Configuring McIDAS-X

- Installing McIDAS-X on Unix or Mac OS X Workstations
- Installing McIDAS-X on Windows 7 Workstations
- Installing McIDAS-X on Windows XP Workstations
- Configuring McIDAS-X Sessions with .mcidasrc
- Registering MD File Schemas

Chapter 2 - Introduction

- Conventions
- Workstation Characteristics
- Starting and Ending a McIDAS Session
- Entering and Editing Commands
- Displaying Images and Graphics
- Using Strings and Function Keys
- Using the McIDAS-X Graphical User Interface
- Using the ADDE
- Getting Started with ADDE

Chapter 3 - Commands

- Documentation Conventions
- Commands by Categories
- Commands by Alphabetical Order

Chapter 4 - McBASI Interpreter

- McBASI Interpreter Commands and Reserved Words
- Functions
- McBASI Sample Programs

Chapter 5 - F Key Menu System

- Creating a Menu File
- Pseudocommands
- Defining Loops
- Sample Menu File

Appendices

- *A* - Global Keywords
- *B* - Julian Day Calendar
- *C* - Sensor Source Numbers
- *D* - Bands for Satellite Imagery
- *E* - Stability Parameters
- *F* - System Files
- *G* - MCPATH
- *H* - Running Commands Outside a McIDAS-X Session
- *I* - McIDAS-X Site Administration and Management

Copyright© 1997-2012
Space Science and Engineering Center (SSEC)
University of Wisconsin - Madison
All Rights Reserved

Permission is granted to make and distribute verbatim copies of this document, provided the copyright notice and this permission are preserved on all copies.

Permission is further granted to modify, amend or otherwise alter this document, and to distribute, including electronically, the modified, amended or otherwise altered document provided the copyright notice and this permission are preserved on all copies and derivative works thereof. In addition, the following notice must be added to this copyright page by each individual or organization that modifies, amends or otherwise alters this document: "This is NOT a verbatim version of the original SSEC document. Portions have been modified, amended or otherwise altered by *[name and address of modifying individual or organization]*."

SSEC makes no warranty of any kind with regard to the software, hardware or accompanying documentation, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. SSEC does not indemnify any infringement of copyright, patent, or trademark through the use or modification of this software.

There is no expressed or implied warranty made to anyone as to the suitability of this software for any purpose. All risk of use is assumed by the user. Users agree not to hold SSEC, the University of Wisconsin-Madison, or any of its employees or assigns liable for any consequences resulting from the use of the McIDAS software.

Mention of any commercial company or product in this document does not constitute an endorsement by SSEC. Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this document, and SSEC was aware of the trademark claim, the designations are printed in caps or initial caps.

The information in this document is subject to change without notice. Considerable effort has been expended to make this document accurate and complete, but SSEC cannot assume responsibility for inaccuracies, omissions, manufacturers' claims or their representations.

Trademarks

The following trademarks and registered trademarks are referenced in the *McIDAS User's Guide*. Any omissions or errors are unintentional.

- Apple, Mac, Macintosh and Mac OS are registered trademarks of Apple Computer, Inc.
- HP, HP-UX and PA-RISC are registered trademarks of Hewlett Packard
- Intel is a trademark of the Intel Corporation
- Linux is a registered trademark of Linus Torvalds
- Microsoft and Windows are registered trademarks of Microsoft Corporation
- Oracle and Java are registered trademarks of Oracle and/or its affiliates
- Red Hat is a trademark of Red Hat, Inc.
- Sun, Sun Microsystems, the Sun Logo, Solaris and the Network is the Computer are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries
- UNIX is a registered trademark of The Open Company

Chapter 1

Installing and Configuring McIDAS-X

This chapter consists of the McIDAS-X installation and configuration instructions. There are five separate sections:

- Installing McIDAS-X on Unix or Mac OS X Workstations
- Installing McIDAS-X on Windows 7 Workstations
- Installing McIDAS-X on Windows XP Workstations
- Configuring McIDAS-X Sessions with .mcidasrc
- Registering MD File Schemas

If you're installing McIDAS-X, go to the section in the bullet list above that corresponds to your configuration.

Installing McIDAS-X on Unix or Mac OS X Workstations

If you are installing McIDAS-X on a workstation running the Windows 7 operating system, go to Installing McIDAS-X on Windows 7 Workstations. If you are installing McIDAS-X on a workstation running the Windows XP operating system, go to Installing McIDAS-X on Windows XP Workstations.

The system administrator or person who installs McIDAS-X will use these installation instructions. The instructions follow the table below, which lists the Unix and Mac OS X software versions compatible with McIDAS-X 2012.2.

McIDAS-X 2012.2 was tested - and is fully supported - on these versions only. In most cases, you should be able to install and run McIDAS-X 2012.2 on workstations running later editions in the same version series (e.g., if McIDAS-X is tested and supported on version 1.2.1, then versions 1.2.2, 1.2.3, etc. are also likely to work correctly). However, please keep in mind that McIDAS User Services is very limited in its ability to acquire and test versions other than those below and thus is limited in its ability to assist in troubleshooting problems that occur with other versions.

Vendor	**Operating System**	**Fortran compiler**	**C compiler**	**X Server / X Window System**
Apple®	Mac OS® X 10.6.8 Intel	*gfortran*: 4.6.0 (available for free at http://hpc.sourceforge.net)	*gcc*: 4.2.1 (packaged with Xcode)	XQuartz 2.7.3 (available for free at http://xquartz.macosforge.org)
	Mac OS X 10.7 Intel	*gfortran*: 4.7.1 (available for free at http://hpc.sourceforge.net)	*gcc*: 4.2.1 (packaged with Xcode)	XQuartz 2.7.3 (available for free at http://xquartz.macosforge.org)
	OS X Notes: 1. Due to a Tcl/Tk incompatibility with OS X 10.6 and 10.7, the McIDAS-X GUI, Configuration GUI and ASK1 command are not available/supported on OS X systems. 2. In order to build and install McIDAS-X on Mac OS X, you must install Xcode Development Tools on your workstation. Xcode Development Tools is included on the OS install media and is available from http://developer.apple.com.			
HP®	HP-UX® 11.11	*vendor:* HP Fortran 90 3.1	*vendor* : HP C/ANSI C 11.11.16	packaged with HP-UX 11.11
Red Hat™	Enterprise Linux® 6.1 Intel	*gfortran* : 4.4.5	*gcc* : 4.4.5	packaged with Enterprise Linux 6.1
	Enterprise Linux 5.7 Intel	*gfortran* : 4.1.2	*gcc* : 4.1.2	packaged with Enterprise Linux 5.7
	Enterprise Linux 5.2 PPC	*gfortran* : 4.1.2	*gcc* : 4.1.2	packaged with Enterprise Linux 5.2
	Linux Notes: 1. On Linux systems, we recommend that you use the GNOME desktop environment. Other desktops, such as KDE, may not display the McIDAS-X windows correctly. 2. Enterprise Linux does not install the **`compress`** utility by default. So you must install **`compress`** if you will be configuring your Enterprise Linux workstation as an ADDE remote server.			
Sun™ / Oracle®	Solaris™ 10 OS SPARC Platform Edition	*vendor* : Sun Studio ™ 11 --**or**-- *g77* : 3.4.6	*vendor* : Sun Studio 11 --**or**-- *gcc* : 3.4.6	packaged with Solaris 10
	Solaris 10	*g77* : 3.4.6	*gcc* : 3.4.6	packaged with

	OS x86 Platform Edition			Solaris 10
	Solaris Note: 1. The Solaris operating system is now maintained by Oracle. SSEC tests on Sun Solaris 10 which was purchased before Oracle acquired Sun Microsystems.			

The McIDAS-X installation consists of seven tasks:

- Preparing the Workstation
- Preparing the mcidas Account
- Obtaining McIDAS-X
- Building and Installing McIDAS-X
- Installing the McIDAS-X ADDE Remote Server
- Removing the Previous Version of McIDAS-X
- Preparing a User Account

If you have questions about the installation, call the McIDAS Help Desk at (608) 262-2455, or send email to McIDAS User Services.

Preparing the Workstation

Before installing McIDAS-X, verify that the workstation is configured to the specifications listed in the table on the previous page. Then go to the section below that applies to your configuration and complete its instructions.

- Unix Workstations
- Mac OS X Workstations

Unix Workstations

Read the two sections below and complete the tasks that apply to your configuration.

Enabling Inter-Process Communication

On HP-UX, Linux and Solaris systems, the system administrator should use the command **`ipcs`** to verify that your workstation configuration has Inter-Process Communication (IPC) enabled in the system kernel. If IPC is not running, rebuild your system kernel with all IPC functions enabled.

Allocating Sufficient Shared Memory

McIDAS-X uses shared memory for frame allocation. On HP-UX systems, the shared memory facility is activated by default and has a sufficient maximum segment size for most McIDAS-X sessions. However, this is not true for Solaris and Linux systems. Therefore, on Solaris and Linux systems the system administrator must force the kernel to activate the shared memory system at boot time and set the maximum segment size to 512 MB or greater the first time McIDAS-X is installed on a Solaris or Linux workstation.

Mac OS X Workstations

Read the two sections below and complete the tasks as instructed.

Allocating Sufficient Shared Memory

McIDAS-X uses shared memory for frame allocation. On some operating systems, the shared memory facility is activated by default and has a sufficient maximum segment size for most McIDAS-X sessions. However, this is not true for Mac OS X systems. Therefore, on Mac OS X systems the system administrator must force the kernel to activate the shared memory system at boot time and set the maximum segment size to 512 MB or greater the first time McIDAS-X is installed on a Mac OS X workstation.

Creating the /home link on Mac OS X

Mac OS X systems use **`/Users`** instead of **`/home`** as on other Unix-based systems. You can create a link from **`/home`** to **`/Users`** by running the command below.

Type: **`sudo ln -s /Users /home`**

Preparing the mcidas Account

Complete the instructions below to prepare the **mcidas** account on Unix and Mac OS X workstations.

1. If the workstation does not have a **mcidas** account, have the system administrator make one. The new account's default **umask** should be set to 755.

2. Log on to the workstation as **mcidas**.

3. Modify the **PATH** of the **mcidas** account so that it contains the necessary directories when you log on; refer to the table below. The default **PATH** of the **mcidas** account may not contain all the required directories or have them in the correct order.

Operating System	**Modification**
All operating systems	If you will be building McIDAS-X using the gcc and g77 compilers, you must also include the directory containing the GCC compilers in the **mcidas** account's **PATH**. See below for other changes needed for each operating system.
Enterprise Linux 5.0+	none necessary
Mac OS X 10.6+	none necessary
HP-UX 11.11	add this directory to the **PATH** : **/usr/bin/X11** also add these two directories: **/opt/ansic/bin** and **/opt/fortran90/bin** before these directories in the **PATH** : **/bin** and **/usr/bin**
Solaris 10	add these two directories to the **PATH** : **/opt/SUNWspro/bin** and **/usr/ccs/bin** and if the **PATH** contains: **/usr/ucb** or **/usr/local/bin** it must follow: **/opt/SUNWspro/bin** and **/usr/ccs/bin** and **/usr/bin**

4. Log out and log on again for the changes to take effect.

Obtaining McIDAS-X 2012.2

Note: For convenience, the McIDAS-X installation files now include both the McIDAS-X core package and the McIDAS-XRD sidecar package (see the McIDAS-XRD Software page for details). Installation of McIDAS-XRD is still optional (done by specifying a non-default argument during the build and install procedure).

The McIDAS-X 2012.2 installation files are listed below.

File Name	Description
mcidas2012.2.sh	shell script that does the McIDAS-X and/or -XRD build and install
mcidas2012.2.tar.Z	compressed tar file containing the McIDAS-X and -XRD software
mcinet2012.2.sh	shell script that configures the McIDAS-X ADDE remote server

Complete the instructions below to obtain the McIDAS-X 2012.2 installation files from the McIDAS Website.

1. Use your Web browser to download the files listed above. Access the McIDAS Website, and follow the link for the McIDAS-X 2012.2 software. Each site has its own login and password for downloading files. Have your site coordinator contact the McIDAS Help Desk if you can't remember yours.

2. Log on to the McIDAS-X workstation as user **mcidas** and copy or move the downloaded files to the **~mcidas** directory.

Building and Installing McIDAS-X 2012.2

Note: For convenience, the McIDAS-X installation files now include both the McIDAS-X core package and the McIDAS-XRD sidecar package (see the McIDAS-XRD Software page for details). Installation of McIDAS-XRD is still optional (done by specifying a non-default value for the *packages* parameter in the steps below).

Complete the steps below to build and install McIDAS-X 2012.2 and/or McIDAS-XRD 2012.2. Run each command from the **~mcidas** directory.

1. Log on to the workstation as **mcidas**.

2. Build the desired package (McIDAS-X 2012.2 and/or McIDAS-XRD 2012.2) using the appropriate command below. If building McIDAS-X (by specifying **x** or **all** in the *packages* parameter), this step first makes the **~mcidas** installation directories and **~mcidas/mcidas2012.2** package directories (see the tables below). Then, it uncompresses the data files, source files, and man pages. Finally, it makes the help files, compiles the source code, and writes the compiler output to the file **~mcidas/mcidas2012.2/src/makelog**.

   Type: **sh ./mcidas2012.2.sh make** *packages*

or

   Type: **sh ./mcidas2012.2.sh** *flags* **make** *packages*

where *packages* is one of the three options described below.

- **x**
  The **x** option builds only the McIDAS-X package. It is also the default (if you don't specify anything in the *packages* parameter.)
- **xrd**
  The **xrd** option builds only the McIDAS-XRD package.
- **all**
  The **all** option builds both the McIDAS-X and McIDAS-XRD packages, in that order.

and *flags* is one or more of the three optional flags described below.

- **-gcc**
  The **-gcc** flag is used to build McIDAS-X using the GCC (GNU Compiler Collection) compilers on an HP-UX, Mac OS X or Solaris SPARC system. If you don't specify this **-gcc** flag, McIDAS-X is built using the default compilers: vendor on HP-UX and Solaris SPARC systems; gcc/gfortran on Mac OS X systems and, if present, on Linux systems; gcc/g77 on Linux and Solaris x86 systems that do not have gfortran.
- **-noopt**
  The **-noopt** flag is used to build McIDAS-X with no optimization. If you don't specify this **-noopt** flag, McIDAS-X is built using the default optimization: -O2 for builds with vendor compilers and -O1 for builds with gcc/g77. The only exception is Solaris builds with vendor compilers, which are done with no optimization.
- **-mysql**
  The **-mysql** flag is used to build McIDAS-X using the MySQL database libraries. This optional flag is mainly for use by McIDAS-XCD sites that elect to use a MySQL database on their -XCD workstations that serve real-time GRIB files from the NOAAPORT data stream. (The database makes access to the GRIB data much faster for -X client workstations.) If using the database on an -XCD workstation, the **-mysql** flag must be specified when building both McIDAS-X and McIDAS-XCD. If the database is in a non-standard install location, specify the flag as **-mysql=*directory***, where ***directory*** is the root MySQL install directory.

Examples:

- **sh ./mcidas2012.2.sh make x**
  This command builds McIDAS-X 2012.2 using the default compilers.
- **sh ./mcidas2012.2.sh -gcc make all**
  This command builds McIDAS-X 2012.2 then McIDAS-XRD 2012.2 using the gcc/g77 compilers.

The message below is displayed if McIDAS-X is selected (by specifying **x** or **all** in the *packages* parameter) and the build is successfully completed.

```
./mcidas2012.2.sh: McIDAS-X make completed normally.
```

If McIDAS-XRD is selected (by specifying **xrd** or **all** in the *packages* parameter) and the build is successfully completed, the message is the same as that above, except *McIDAS-X* is replaced with *McIDAS-XRD*.

3. Once McIDAS-X and/or McIDAS-XRD 2012.2 are successfully built, uninstall the previous version of the corresponding package(s).

First verify that no McIDAS-X sessions are running. Then, if the previous version is 2009.1 or later, enter the command below to uninstall the previous version of the packages, replacing *version#* with the number of the version (for example, 2012.1). If the previous version is 2008a or earlier, complete the uninstall procedure (step 3) in the 2009.1 version of this manual. Then return to these instructions and proceed to step 4 below.

Type: **sh ./mcidas***version#***.sh uninstall all**

The messages below are displayed if the uninstall is successfully completed.

```
./mcidasversion#.sh: McIDAS-X uninstall completed normally. ./mcidasversion#.sh:
McIDAS-XRD uninstall completed normally.
```

4. Run the command below to install the packages (McIDAS-X and/or McIDAS-XRD 2012.2) that you built in step 2 above. You must specify the same option in *packages* (**x**, **xrd** or **all**) as you did in step 2 above. This step creates hard links from the files in the package directories to the appropriate **~mcidas** installation directories.

Type: **sh ./mcidas2012.2.sh install** *packages*

The message below is displayed if McIDAS-X is selected (by specifying **x** or **all** in the *packages* parameter) and the install is successfully completed.

```
./mcidas2012.2.sh: McIDAS-X install completed normally.
```

If McIDAS-XRD is selected (by specifying **xrd** or **all** in the *packages* parameter) and the install is successfully completed, the message is the same as that above, except *McIDAS-X* is replaced with *McIDAS-XRD*.

Installation Directories

The install process makes hard links (as opposed to symbolic links) between the installation directories and the files in the package directories. The uninstall process simply removes the hard links. The McIDAS-X installation directories and their contents are listed below.

Directory	Contents
~mcidas/bin	program executables
~mcidas/data	data files
~mcidas/help	help files
~mcidas/inc	include files
~mcidas/lib	libraries
~mcidas/man	man files for subroutines and functions
~mcidas/tcl	Tcl and Tk executables and libraries

Other McIDAS packages also make hard links in the installation directories. Thus the installation directories are the repositories for all McIDAS packages currently installed.

Package Directories

Each version of McIDAS-X and other McIDAS packages (for example, McIDAS-XCD) builds its own set of directories. The names of the directories depend on the package name and version number. The McIDAS-X package directories and their contents are listed below.

Directory	Contents
~mcidas/mcidas201*x***/data**	McIDAS-X data files
~mcidas/mcidas201*x***/bzip2**	BZIP2 1.0.5 files for the radar servers
~mcidas/mcidas201*x***/g2clib**	g2clib 1.0.4 libraries for GRIB2
~mcidas/mcidas201*x***/gzip**	gzip 1.2.4 files for gzip compression
~mcidas/mcidas201*x***/hdf**	HDF 4.2.5 files for the MODIS servers

~mcidas/mcidas201*x*/jasper	jasper 1.701.0 libraries for GRIB2
~mcidas/mcidas201*x*/jpeg	JPEG v6b files for the FRMSAVE command
~mcidas/mcidas201*x*/libgeotiff	GeoTIFF 1.2.1 files for the GeoTIFF server
~mcidas/mcidas201*x*/man	McIDAS-X man files for subroutines and functions
~mcidas/mcidas201*x*/netcdf	netCDF 4.1 files for ADDE servers
~mcidas/mcidas201*x*/src	McIDAS-X source files, help files, and binaries
~mcidas/mcidas201*x*/tcl	Tcl 8.5.9 files for the GUI
~mcidas/mcidas201*x*/tiff	TIFF 3.6.1 files for the GeoTIFF server
~mcidas/mcidas201*x*/tk	Tk 8.5.9 files for the GUI
~mcidas/mcidas201*x*/zlib	zlib 1.1.4 files for the NEXRAD servers

Installing the McIDAS-X 2012.2 ADDE Remote Server

If this workstation has data that you want to make available to users on other workstations, you should install the McIDAS-X 2012.2 ADDE remote server. If the workstation does not have such data, proceed to *Removing the Previous Version of McIDAS-X*.

The 2012.2 ADDE remote server uses port 112 for gzip-compressed, uncompressed and compressed data transfers. See *Clients and Servers*, and *Appendix I, McIDAS-X Site Administration and Management* for more information about remote servers.

Continue with the procedure below to remove the previous version's ADDE remote server, and install the McIDAS-X 2012.2 ADDE remote server. If any other service is using port 112, you will be notified during the procedure and be given the option of overriding it with the McIDAS ADDE service.

1. Log on to the workstation as **root**. (Note that you cannot run the commands below using **sudo**; you must be logged on as **root**.)

2. Change to the home directory for user **mcidas**.

   Type: **cd ~mcidas**

3. Uninstall the previous version's ADDE remote server. Replace *version#* in the command below with the previous version number of McIDAS-X (2012.1, for example). If you used an account other than **mcadde** for your ADDE remote server, insert that account name in place of **mcadde**.

   Type: **sh ./mcinet***version#***.sh uninstall mcadde**

This step removes the modifications made to the Internet services daemon (**inetd** on HP-UX and Solaris; **xinetd** on Linux and OS X) configuration files, forces **inetd** to reread its configuration files, and attempts to verify that **inetd** is no longer listening for ADDE requests from McIDAS-X clients.

To install the McIDAS-X 2012.2 ADDE remote server, go to step 4. Otherwise, proceed to *Removing the Previous Version of McIDAS-X*.

4. Install the McIDAS-X 2012.2 ADDE remote server. If you use an account other than **mcadde** for your ADDE remote server, insert that account name in place of **mcadde** in the command below.

   Type: **sh ./mcinet2012.2.sh install mcadde**

This step modifies the Internet services daemon (**inetd**) configuration files to make **inetd** aware of the ADDE service, forces **inetd** to reread its configuration files, and attempts to verify that **inetd** is listening for ADDE requests from McIDAS-X clients.

Notes:

1. You can check the status of the ADDE remote server at any time by running the command **sh ./mcinet***version#***.sh status** from the **~mcidas** directory while logged in as **root**.
2. If you make changes to your **inetd** configuration you can restart **inetd** by running the command **sh ./mcinet***version#***.sh restart** from the **~mcidas** directory while logged in as **root**.

If you have installed the ADDE remote server and want to make datasets that are accessible to McIDAS-X client workstations, go to *Appendix I, McIDAS-X Site Administration and Management* and its *Acquiring and Naming Data on a Remote Server* page for more information.

Removing the Previous Version of McIDAS-X

In versions 2009.1 and later, the McIDAS-X installation files include both the McIDAS-X core package and the McIDAS-XRD sidecar package.

If you have installed and tested a new version of McIDAS-X and/or McIDAS-XRD and are satisfied they are working properly, you can remove the previous version of the packages. If the previous version is 2009.1 or later, complete the numbered instructions below, running each command from the **~mcidas** directory. If the previous version is 2008a or earlier, complete the unmake procedure(s) in the 2009.1 version of this manual. Then return to this manual and proceed with its instructions.

1. Log on to the workstation as **mcidas**.

2. Remove the **mcidas***version#* directory tree and its contents, replacing *version#* with the previous version number.

   Type: **sh ./mcidas*version#*.sh unmake all**

The messages below are displayed when the unmake is successfully completed.

```
./mcidasversion#.sh: McIDAS-X unmake completed normally. ./mcidasversion#.sh:
McIDAS-XRD unmake completed normally.
```

3. Remove the following files:

- **mcidas***version#***.sh**
- **mcidas***version#***.tar.Z**
- **mcinet***version#***.sh**

Preparing a User Account

The McIDAS-X software is installed in the **mcidas** account. However, you can not run McIDAS-X as user **mcidas**. Ask your system administrator to create individual accounts for running McIDAS-X. Then, the user or system administrator must complete these two tasks for each account:

- Setting the Environment
- Enabling McIDAS-X Keys

Setting the Environment

1. Log on to the account and modify the environment variable **PATH** in your **$HOME/.profile** (ksh) or **$HOME/.cshrc** (csh) file. Insert **$HOME/mcidas/bin**, the directory with your site's locally developed code (the suggested directory is **~mclocal/mcidas/bin**), and **~mcidas/bin**, in that order, at the beginning of your **PATH**.

2. You may also need to modify your **PATH** if it does not contain all of the required directories or have them in the correct order. If you will be writing and compiling your own McIDAS software, update your **PATH** with the modifications listed in step 3 of *Preparing the mcidas Account*.

If you will *not* be writing and compiling McIDAS software, update your **PATH** with the modifications listed below.

Operating System	**Modification**
HP-UX 11.11	add this directory to your path: **/usr/bin/X11**
Enterprise Linux 5.0+	none necessary
Mac OS X 10.6+	none necessary
Solaris 10	none necessary

3. Log out and log on again for the changes to take effect.

Enabling McIDAS-X Keys

The actions required to enable certain keys used by McIDAS-X (for example, **Alt A**, **Ctrl F11**) vary depending on your workstation and its configuration. This section describes these actions for workstations running the following SSEC-recommended configurations:

- HP-UX using the Visual Users Environment window manager
- Linux using the GNOME desktop environment
- Mac OS X using the Apple's X11 desktop environment
- Solaris using the OPEN LOOK window manager
- Solaris using the Common Desktop Environment

These directions assume you are in the account's **$HOME** directory and logged on to the workstation console.

Enabling McIDAS-X Keys for HP-UX Visual Users Environment --Follow the steps below if you use an HP-UX workstation console and the Visual Users Environment window manager (**vuewm**).

1. Set the environment variable **EDITOR**. If you use sh or ksh, enter the command below.

   Type: **export EDITOR; EDITOR=vi**

If you use **csh**, enter the command below.

   Type: **setenv EDITOR vi**

2. Use the command below to display the **vi** editor and the **vuewm** resource file in a separate window.

   Type: **/usr/vue/bin/vueaction EditResources**

3. Add the line below to the **vuewm** resource file using the **vi** editor, write the file, and quit **vi**.

```
Vuewm*windowMenu: NoAcceleratorWindowMenu
```

This allows McIDAS-X to use the **Alt F4** key.

4. Use the command below to display the **vi** editor and the **vuewm** initialization file in a separate window.

Type: **/usr/vue/bin/vueaction EditVuewmrc**

5. Delete or comment out the line below in the **Keys VueKeyBindings** section of the **vuewm** initialization file. Then, write the file and quit **vi**.

```
Meta<key>F6 window f.next_key transient
```

This allows McIDAS-X to use the **Alt F6** key.

6. Log out and log on again for the changes to take effect.

Enabling McIDAS-X Keys for Linux and GNOME --Follow the steps below if you use an Linux workstation console and the GNOME desktop environment.

1. Bring up the *Keyboard Shortcuts* window by clicking the following in order: **Main Menu** (in lower-left corner), **Preferences**, **Keyboard Shortcuts**.

2. Highlight each shortcut associated with a key combination you want to use in McIDAS-X (e.g., <ALT>F2) and press the Backspace key to change it to *Disabled*. This allows McIDAS-X to use the key combination for strings and other applications.

Enabling McIDAS-X Keys for Mac OS X and X11 --Follow the steps below if you use an Mac OS X workstation console and Apple's X11 desktop environment.

1. Start X11 by double-clicking the **X11** icon in the *Applications>Utilities* folder. If X11 is already running, click the **X11** icon in the dock to make the application active.

2. Click the **X11** menu at the top of the display, then click **Preferences** to bring up the *X11 Preferences* window.

3. Click the **Input** tab, then uncheck the checkbox labeled *Enable key equivalents under X11*.

Steps 1-3 allow the **open-apple** keys (on each side of the spacebar) to function as the **Alt** keys in McIDAS-X.

4. Start a Terminal by double-clicking the **Terminal** icon in the *Applications>Utilities* folder. If a Terminal is already running, click the **Terminal** icon in the dock to make the application active. Then use the **xmodmap** utility to configure a key to function as the **Insert** key during McIDAS-X command entry. Choose the appropriate command below, depending on whether you have a fullsize (desktop) keyboard or a laptop.

On a fullsize keyboard, run the command below to configure the **help** key to function as **Insert**.

Type: **/usr/X11R6/bin/xmodmap -e 'keysym Help = Insert'**

On a laptop, run the command below to configure the **option (alt)** key to function as **Insert**.

Type: **/usr/X11R6/bin/xmodmap -e 'keysym Mode_switch = Insert'**

Enabling McIDAS-X Keys for Solaris OPEN LOOK --Follow the steps below if you use a Solaris workstation console and the OPEN LOOK window manager (**olwm**).

1. Copy files **.openwin-init** and **.Xdefaults** to the account's home directory if they do not already reside there.

Type: **cp $OPENWINHOME/lib/Xdefaults .Xdefaults**

Type: **cp $OPENWINHOME/lib/openwin-init .openwin-init**

Type: **chmod u+w .Xdefaults**

Type: **chmod u+wx .openwin-init**

2. Add the following line to the account's **.Xdefaults** file.

```
OpenWindows.KeyboardCommands: SunView1
```

This allows McIDAS-X to use the **Alt Q** key.

3. Add the following line to the account's **.openwin-init** file.

```
xmodmap -e 'keycode 16 = F11' -e 'keycode 18 = F12'
```

This allows McIDAS-X to use the **F11** and **F12** keys, and their Alt, Shift and Ctrl forms.

4. Log out and log on again for the changes to take effect.

Enabling McIDAS-X Keys for Solaris Common Desktop Environment --Follow the steps below if you use a Solaris workstation console and the Common Desktop Environment (**dtwm**).

1. Create or modify the account's Xdefaults file **$HOME/.Xdefaults** and add the following line.

```
Dtwm*windowMenu: NoAcceleratorWindowMenu
```

2. Select the *Edit Dtwmrc* icon from the Desktop_Tools section of the Application Manager. The Application Manager is started by clicking its icon on the Front Panel.

3. Delete or comment out the line below in the **Keys DefaultKeyBindings** and **Keys DtKeyBindings** sections in the account's **dtwmrc** file.

```
Alt<Key>F6 window f.next_key transient
```

This allows McIDAS-X to use the **Alt F6** key.

4. Log out and log on again for the changes to take effect.

This concludes the instructions for installing McIDAS-X and configuring user accounts. Continue with *Configuring McIDAS-X Sessions with .mcidasrc*, which provides information about tailoring McIDAS-X sessions to your preferences.

Installing McIDAS-X on Windows 7 Workstations

If you are installing McIDAS-X on a workstation running the Windows XP operating system, go to Installing McIDAS-X on Windows XP Workstations. If you're installing McIDAS-X on a workstation running the HP-UX, Linux, Solaris or Mac OS X operating system, go to Installing McIDAS-X on Unix or Mac OS X Workstations.

This section contains background information about and the procedure for installing McIDAS-X on PC workstations running Microsoft Windows 7 with Subsystem for UNIX-based Applications (SUA). Hereafter, the terms *McIDAS-X for Win7* and *McIDAS-X for Win7 2012.2* refer specifically to this package. The McIDAS-X command suite, which is identical in all three packages (Unix, Windows 7 and Windows XP), is referred to as *McIDAS-X* or *McIDAS-X 2012.2*.

Because the Win7 package is distributed as both binary and source code, and the Unix package is distributed as source code only, which must be compiled (built) on-site, the installation files and procedures for these packages are different.

The table below lists the software versions compatible with McIDAS-X for Win7 2012.2. McIDAS-X for Win7 2012.2 was tested on these versions only. The McIDAS Users' Group is extremely limited in its ability to support sites running McIDAS-X for Win7 2012.2 with other software versions.

Operating System	**Fortran compiler**	**C compiler**	**X server**
Microsoft® Windows® 7 Enterprise with Subsystem for UNIX-based Applications (SUA) and SUA Community Complete Toolset	*g77* : 3.3	*gcc* : 3.3	see below

System Configuration Notes:

1. Because McIDAS-X for Win7 is distributed as a binary package, the Fortran and C compilers are needed only if you do local development on the workstation. The compilers are packaged with the SUA software.

2. We do not have a specific X server that we recommend for Windows 7. The two that are used most at SSEC are Xming (packaged with SUA Community Complete Toolset) and X-Win32. Neither work flawlessly, as we have encountered problems with both, typically with cutting and pasting or when using cursor-based commands like CUR, PC or CM. If you encounter these types of problems with your X server, you may find that setting or unsetting the **-noXCursor** flag in your $HOME/.mcidasrc file improves performance.

3. Unlike Unix, Windows 7 does not have a single rooted file system. Rather than mounting a drive on any arbitrary mount point in a file system, the drive must be mounted on a drive letter (e.g., C: or D:). Because SUA uses the Windows 7 file system, the pathnames to files may look different than they do in Unix. There are two types of pathnames on a Windows 7 system running SUA: Win32 and POSIX. The Win32 format (e.g., **c:\Windows\SUA\bin**) is the standard format used in Windows and DOS environments. It is used in all Windows 7 Win32 applications. The POSIX format (e.g., **/dev/fs/C/Windows/SUA/bin**) is the standard format used in Unix environments. It is used with SUA applications, including McIDAS-X.

The installation instructions appear below. The system administrator or person who installs McIDAS-X for XP will use these installation instructions, which consist of six tasks:

- Preparing the Windows 7 Workstation
- Preparing the mcidas Account on Windows 7
- Obtaining McIDAS-X for Win7
- Installing McIDAS-X for Win7
- Removing the Previous Version of McIDAS-X for Win7
- Preparing a User Account on Windows 7

If you have questions about the installation, call the McIDAS Help Desk at (608) 262-2455, or send email to McIDAS User Services .

Preparing the Windows 7 Workstation

Before installing McIDAS-X for Win7, verify that the workstation is configured to the specifications listed on the previous page. Then complete the four tasks below.

- Preparing to Install SUA Using the Windows Interface
- Installing SUA Using the Windows Interface
- Installing the SUA Community Complete Toolset
- Disabling Physical Address Extension (PAE)

Preparing to Install SUA Using the Windows Interface

Complete the instructions below to create a new menu item enabling you to download the SUA installation package. For more information about SUA, see the Microsoft SUA Overview.

1. Log on to the workstation as **Administrator**. (Note that you cannot complete the instructions below as another user with escalated privileges; you must be logged on as **Administrator**.)

2. Click **Start**, and then click **Control Panel** to bring up the *Control Panel* window.

3. In the *Control Panel* window, click **Programs** and under *Programs and Features* click **Turn Windows features on or off**.

4. If the *User Account Control* dialog box opens, click **Continue**. Otherwise, proceed to the next step.

5. In the *Windows Features* dialog box, select the **Subsystem for UNIX-based Applications** check box, then click **OK**.

Installing SUA Using the Windows Interface

Complete the instructions below to install the SUA package.

1. Log on to the workstation as **Administrator**. (Note that you cannot complete the instructions below as another user with escalated privileges; you must be logged on as **Administrator**.)

2. Click **Start** and then, in the in the *Subsystem for UNIX-Based Applications* program folder, select **Download Utilities for Subsystem for UNIX-Based Applications** to open a webpage with link to download the SUA package.

3. Download the appropriate SUA installation file for your workstation configuration (e.g., 64bit x86, 32bit x86, 64bit Itanium).

4. Double-click the downloaded file to run the installation wizard.

5. After completing the customer information and license agreement, select **Custom Install** for the *Installation Type* and choose to install all features on the local drive.

6. In the *Security Settings* accept the defaults of none selected, and in the *Installation Location*, accept the default of **C:\Windows\SUA**.

Installing the SUA Community Complete Toolset

Complete the instructions below to download and install the SUA Community Complete Toolset.

1. Log on to the workstation as **Administrator**. (Note that you cannot complete the instructions below as another user with escalated privileges; you must be logged on as **Administrator**.)

2. Download the 32 bit version of the **Complete Toolset** from the SUA Community webpage. (Note: You **must** download and install the 32 bit [not 64 bit] version, regardless of your workstation configuration.)

3. Open an SUA Korn Shell and change to the folder where you downloaded the package in step 2 above. For example, if you downloaded it to the **Downloads** folder,

   Type: **cd Downloads**

4. Run the installation file. For example, if the file is named **pkg-current-bundlecomplete60.exe**,

   Type: **./pkg-current-bundlecomplete60.exe**

The command above starts the SUA Community Complete Toolset installation. This process uses the **pkg** tools to install many common UNIX utility programs, and will attempt to contact various websites to auto-download updates. A connection to the Internet is recommended. Many packages will print warnings about the case sensitivity of the filesystem, but these can be safely ignored. The final step runs a GUI installer for XMing, an X Server that many people at SSEC use with McIDAS-X for Win7.

Disabling Physical Address Extension (PAE)

McIDAS-X may not run correctly or at all on 32 bit Windows 7 systems if Physical Address Extension (PAE) is enabled. This problem, which applies only to 32 bit (not 64 bit) Windows, can occur if OptIn configuration is set. The OptIn configuration enables Data Execution Prevention (DEP), which implicitly enables PAE.

You can check if PAE is enabled on your 32 bit Windows 7 system with **Start > Control Panel > System and Security > System**. If PAE is enabled, the bottom of the *System* section will include a line saying "*Physical Address Extension*". If it's enabled you should disable it and DEP by running the following commands in a DOS Command Prompt while logged on as **Administrator** (not another user with escalated privileges).

Type: **bcdedit /set nx AlwaysOff**

Type: **bcdedit /set pae ForceDisable**

For more information about PAE and DEP, see http://msdn.microsoft.com/en-us/library/windows/desktop/aa366796%28v=vs.85%29.aspx.

Preparing the mcidas Account on Windows 7

Complete the instructions below to prepare the **mcidas** account on Windows 7 workstations.

1. Log on to the workstation as **Administrator**. (Note that you cannot complete the instructions below as another user with escalated privileges; you must be logged on as **Administrator**.)

2. If the workstation does not have a **mcidas** account, make one.

3. Set the Local Path for the **mcidas** account to **C:\Users\mcidas**. To do so, right-click on *My Computer* then click **Manage** to bring up the *Computer Management* window. Then browse to the **mcidas** user by expanding **Local Users and Groups > Users > mcidas** to bring up the *mcidas Properties* window. Click its **Profile** tab, select the radio button next to **Local path** and enter **C:\Users\mcidas** in its text field and click **OK**.

If this workstation is a member of a domain controlled by an Active Directory server, the network administrator will need to configure the home directory for the **mcidas** account on the server to match the one used here.

4. Create two links to **/dev/fs/C/Users**, set ownership of the **mcidas** user's home directory and give everyone Read access. To do so, start an SUA Korn Shell and run the four commands below in order.

Type: **ln -s /dev/fs/C/Users /home**

Type: **ln -s /dev/fs/C/Users /dev/fs/C/home**

Type: **cd /home**

Type: **chown mcidas mcidas**

Type: **chmod 755 mcidas**

Obtaining McIDAS-X for Win7 2012.2

Note: For convenience, the McIDAS-X for Win7 installation files include both the McIDAS-X core package and the McIDAS-XRD sidecar package (see the McIDAS-XRD Software page for details). Installation of McIDAS-XRD is optional (done by specifying a non-default argument during the install procedure).

File Name	Description
mcidas2012.2_win7.sh	shell script that installs McIDAS-X and/or -XRD
mcidas2012.2_win7.tar.gz	gzip-compressed file containing the McIDAS-X and -XRD software
mcidas.mcenv	environment variables used by McIDAS-X
mcidas.profile	template for creating a **.profile** file in user accounts that don't have a pre-existing **.profile**

Complete the instructions below to obtain the McIDAS-X for Win7 2012.2 installation files from the McIDAS Website.

1. Use your Web browser to download the files listed above. Access the McIDAS Website, and follow the link for the McIDAS-X for Win7 2012.2 software. Each site has its own login and password for downloading files. Have your site coordinator contact the McIDAS Help Desk if you can't remember yours.

2. Log on to the McIDAS-X workstation as user **mcidas** and copy or move the downloaded files to the **~mcidas** directory.

Installing McIDAS-X for Win7 2012.2

Note: For convenience, the McIDAS-X for Win7 installation files include both the McIDAS-X core package and the McIDAS-XRD sidecar package (see the McIDAS-XRD Software page for details). Installation of McIDAS-XRD is optional (done by specifying a non-default value for the *packages* parameter in step 3 below).

Complete the steps below to install McIDAS-X for Win7 2012.2 and/or McIDAS-XRD for Win7 2012.2. Run each command from the **~mcidas** directory.

1. Log on to the workstation as **mcidas** and start an SUA Korn Shell.

   If there is a previous version of the package(s) you will be installing (McIDAS-X for Win7 and/or McIDAS-XRD for Win7) already installed, go to step 2. If this is the initial installation of the package(s), go to step 3.

2. Uninstall the previous version of the McIDAS-X and McIDAS-XRD packages.

   First verify that no McIDAS-X sessions are running. Then enter the command below to uninstall the previous version of the packages, replacing *version#* with the number of the version (for example, 2012.1).

   Type: **sh ./mcidas***version#***_win7.sh uninstall all**

   The messages below appear if the uninstall is successfully completed.

```
./mcidasversion#_win7.sh: McIDAS-X uninstall completed normally.
./mcidasversion#_win7.sh: McIDAS-XRD uninstall completed normally.
```

3. Install the desired package (McIDAS-X for Win7 2012.2 and/or McIDAS-XRD for Win7 2012.2) using the appropriate command below. If installing McIDAS-X for Win7 (by specifying **x** or **all** in the *packages* parameter), this step first makes the **~mcidas** installation directories and **~mcidas/mcidas2012.2** package directories (see the tables below). Then, it uncompresses the binary files, data files, source files, help files, and man pages. Finally, it creates hard links from the files in the **~mcidas/mcidas2012.2** package directories to the appropriate **~mcidas** installation directories.

   Type: **sh ./mcidas2012.2_win7.sh install** *packages*

   where *packages* is one of the three options described below.

   - **x**
     The **x** option builds only the McIDAS-X for Win7 package.
   - **xrd**
     The **xrd** option builds only the McIDAS-XRD for Win7 package.
   - **all**
     The **all** option builds both the McIDAS-X for Win7 and McIDAS-XRD for Win7 packages, in that order.

   The message below is displayed if McIDAS-X for Win7 is selected (by specifying **x** or **all** in the *packages* parameter) and the install is successfully completed.

```
./mcidas2012.2_win7.sh: McIDAS-X install completed normally.
```

   If McIDAS-XRD for Win7 is selected (by specifying **xrd** or **all** in the *packages* parameter) and the install is successfully completed, the message is the same as that above, except *McIDAS-X* is replaced with *McIDAS-XRD*.

Installation Directories

The install process makes hard links (as opposed to symbolic links) between the installation directories and the files in the package directories. The uninstall process simply removes the hard links. The McIDAS-X installation directories and their contents are listed below.

Directory	Contents
~mcidas/bin	program executables
~mcidas/data	data files

~mcidas/help	help files
~mcidas/inc	include files
~mcidas/lib	libraries
~mcidas/man	man files for subroutines and functions
~mcidas/tcl	Tcl and Tk executables and libraries

Other McIDAS packages also make hard links in the installation directories. Thus the installation directories are the repositories for all McIDAS packages currently installed.

Package Directories

Each version of McIDAS-X and other McIDAS packages (for example, McIDAS-XCD) builds its own set of directories. The names of the directories depend on the package name and version number. The McIDAS-X for Win7 package directories and their contents are listed below.

Directory	**Contents**
~mcidas/mcidas201*x*/bin	McIDAS-X binary executables
~mcidas/mcidas201*x*/data	McIDAS-X data files
~mcidas/mcidas201*x*/help	McIDAS-X help files
~mcidas/mcidas201*x*/inc	McIDAS-X include files
~mcidas/mcidas201*x*/lib	McIDAS-X library files
~mcidas/mcidas201*x*/man	McIDAS-X man files for subroutines and functions
~mcidas/mcidas201*x*/src	McIDAS-X source files
~mcidas/mcidas201*x*/tcl	Tcl/Tk 8.5.9 files for the GUI

Removing the Previous Version of McIDAS-X for Win7

The McIDAS-X for Win7 installation files include both the McIDAS-X for Win7 core package and the McIDAS-XRD for Win7 sidecar package.

If you have installed and tested a new version of McIDAS-X for Win7 and/or McIDAS-XRD for Win7 and are satisfied they are working properly, you can remove the previous version of the packages. To do so, complete the numbered instructions below, running each command from the **~mcidas** directory.

1. Log on to the workstation as **mcidas** and start an SUA Korn Shell.

2. Remove the **mcidas***version#* directory tree and its contents, replacing *version#* with the previous version number.

   Type: **sh ./mcidas***version#***_win7.sh unmake all**

The messages below are displayed when the unmake is successfully completed.

```
./mcidasversion#_win7.sh: McIDAS-X unmake completed normally.
./mcidasversion#_win7.sh: McIDAS-XRD unmake completed normally.
```

3. Remove the following files:

- **mcidas***version#***_win7.sh**
- **mcidas***version#***_win7.tar.gz**

Preparing a User Account on Windows 7

The McIDAS-X software is installed in the **mcidas** account. However, you can not run McIDAS-X as user **mcidas**. Individual user accounts must be created and configured for running McIDAS-X. The four tasks below must be completed for each user account.

- Making and Configuring the User Account
- Setting the SUA Environment
- Setting the Windows 7 Environment
- Setting the X Server Properties

The system administrator must complete the first task. The system administrator or user can complete the second, third and fourth tasks.

Making and Configuring the User Account

1. Log on to the workstation as **Administrator**. (Note that you cannot complete the instructions below as another user with escalated privileges; you must be logged on as **Administrator**.)

2. Create the user account if it doesn't already exist.

3. Set the Local Path for the user to their home directory, e.g., **C:\Users\johndoe**. To do so, right-click on *My Computer* then click **Manage** to bring up the *Computer Management* window. Then browse to the user by expanding **Local Users and Groups > Users >** *user*, where *user* is the user's account name, to bring up their *user Properties* window. Click its **Profile** tab, select the radio button next to **Local path** and enter the user's home directory, e.g., **C:\Users\johndoe**, in its text field and click **OK**.

If this workstation is a member of a domain controlled by an Active Directory server, the network administrator will need to configure the home directory for the *user* account on the server to match the one used here.

4. Create a link to **/dev/fs/C/Users**, set ownership of the user's home directory and give everyone Read access. To do so, start an SUA Korn Shell and run the four commands below in order, replacing *user* with the user's account name.

Type: **ln -s /dev/fs/C/Users /home**

Type: **cd /home**

Type: **chown** *user user*

Type: **chmod 755** *user*

Setting the SUA Environment

1. Log on to the user account and start an SUA Korn Shell.

2. Verify that the files **mcidas.profile** and **mcidas.mcenv** are present in the **~mcidas** directory, as specified in *Obtaining McIDAS-X for Win7*.

Type: **ls -l ~mcidas/mcidas.profile ~mcidas/mcidas.mcenv**

The **mcidas.profile** file may be used in step 6 below as a template for creating a **.profile** in user accounts that don't have a pre-existing **.profile**. The **mcidas.profile** file is a modified version of the **profile.usr** file supplied by SUA in the **/dev/fs/C/Windows/SUA/etc** directory. The **PATH** was modified to include the required McIDAS directories and a line was added to source the **.mcenv** file.

The **mcidas.mcenv** file contains the environment variables used by McIDAS-X for Win7. It is copied to **.mcenv** in step 7 below and is sourced (run) by **.profile** each time the user logs on. Its contents may be modified by the user to their individual preferences in step 8 below.

3. Change to the SUA home directory of the user account.

Type: **cd $HOME**

4. Copy the **mcidas.profile** and **mcidas.mcenv** files to the current directory.

Type: **cp ~mcidas/mcidas.profile .**

Type: **cp ~mcidas/mcidas.mcenv .**

5. Modify or create the **.profile** file as follows. If the file does not exist, go to step 6 to create it. If it already exists, add the following two lines (substitute the equivalent if using C shell) to the **PATH** environment variable section.

PATH=$HOME/mcidas/bin:~mclocal/mcidas/bin:~mcidas/bin:$PATH
export PATH

Also, add the following line anywhere in the file; then, go to step 7.

. $HOME/.mcenv

6. Create the **.profile** file by copying the **mcidas.profile** file to .**profile**.

   Type: **cp mcidas.profile .profile**

7. Copy the **mcidas.mcenv** file to **.mcenv**.

   Type: **cp mcidas.mcenv .mcenv**

8. Carefully read the **.profile** and **.mcenv** files and, if necessary, modify their settings and contents to your preferences.

Setting the Windows 7 Environment

Complete the task below to make the Windows 7 environment compatible for running McIDAS-X for Win7.

Creating a McIDAS-X Shortcut -- As noted in Starting McIDAS-X, you can start McIDAS-X from a Unix prompt (in an SUA Korn Shell). You can also create a shortcut on the Windows 7 desktop that lets you start and run McIDAS-X via the Configuration GUI. Complete the steps below to create the shortcut.

1. Right click on the Windows 7 desktop. Then select **New**, and click **Shortcut** to bring up the *Create Shortcut* window.

2. In the *Type the location of the item:* field, insert the following command (paying careful attention to forward and backward slashes).

```
%WINDIR%\posix.exe /u /c /bin/ksh -l -c "mcidas -config"
```

3. Click **Next**, then type a name for the shortcut, e.g., *McIDAS*, then click **Finish**.

4. A shortcut will appear on the desktop. This shortcut can then be copied anywhere. For example, if you want McIDAS-X to start automatically at login, copy the shortcut icon to your *Startup* folder. Note that you must have an X server running before you start McIDAS-X. Continue with the following section to set the X server properties.

Setting the X Server Properties

You must have an X server running in order to display the McIDAS-X for Win7 windows. We do not recommend a specific X server for use with McIDAS-X for Win7. So we cannot identify the exact X server configuration changes you will need to perform to make it compatible with McIDAS-X for Win7. However, many X servers require the changes described below.

- Emulating a Three-Button Mouse
- Mapping the Keyboard

Emulating a Three-Button Mouse --McIDAS-X for Win7 is designed for systems with a three-button mouse, as several commands require you to press the middle of the three buttons. If you have a two-button mouse, you must configure your X server to emulate a three-button mouse. Use the mouse section of your X server's configuration program to do so.

Mapping the Keyboard --Modification of the X server's keyboard mapping is sometimes required to enable certain keys used by McIDAS-X (for example, **Alt A**, **Ctrl F11**). Use the keyboard section of your X server's configuration program to select the proper keyboard mapping for your X server.

This concludes the instructions for installing McIDAS-X for Win7 and configuring user accounts. Continue with *Configuring McIDAS-X Sessions with .mcidasrc*, which provides information about tailoring McIDAS-X sessions to your preferences.

Installing McIDAS-X on Windows XP Workstations

If you are installing McIDAS-X on a workstation running the Windows 7 operating system, go to Installing McIDAS-X on Windows 7 Workstations. If you're installing McIDAS-X on a workstation running the HP-UX, Linux, Solaris or Mac OS X operating system, go to Installing McIDAS-X on Unix or Mac OS X Workstations.

This section contains background information about and the procedure for installing McIDAS-X on PC workstations running Microsoft Windows XP with Services for UNIX (SFU). Hereafter, the terms *McIDAS-X for XP* and *McIDAS-X for XP 2012.2* refer specifically to this package. The McIDAS-X command suite, which is identical in all three packages (Unix, Windows 7 and Windows XP), is referred to as *McIDAS-X* or *McIDAS-X 2012.2*.

Because the XP package is distributed as both binary and source code, and the Unix package is distributed as source code only, which must be compiled (built) on-site, the installation files and procedures for these packages are different.

The table below lists the software versions compatible with McIDAS-X for XP 2012.2. McIDAS-X for XP 2012.2 was tested on these versions only. The McIDAS Users' Group is extremely limited in its ability to support sites running McIDAS-X for XP 2012.2 with other software versions.

Operating System	**Fortran compiler**	**C compiler**	**X server**
Microsoft® Windows® XP Professional with Service Pack 3 and Services for UNIX 3.5	*g77* : 3.3	*gcc* : 3.3	see below

System Configuration Notes:

1. Because McIDAS-X for XP is distributed as a binary package, the Fortran and C compilers are needed only if you do local development on the workstation. The compilers are included in the Software Development Kit (SDK) available with Services for UNIX (SFU).

2. We do not have a specific X server that we recommend for XP. We have tried many, but have encountered problems with all of them, especially when cutting and pasting. Some work better than others; many people at SSEC use Exceed 6.1.1 (an older version that's no longer available), X-Win32 or MI/X (available for $25 from www.microimages.com).

3. McIDAS-X for XP will not run (it gives a *mcimagem: ERROR fork()ing* error at startup) if the Windows XP Physical Address Extension (PAE) is enabled. This problem may be encountered with Windows XP Service Pack 2 because SP2 uses the OptIn configuration by default. The OptIn configuration enables Data Execution Prevention (DEP), which implicitly enables PAE.

   You can check if PAE is enabled on your system with **Start > Control Panel > Performance and Maintenance > System** to bring up the *System Properties* window. If PAE is enabled, the bottom of the *Computer* section of the **General** tab will include a line saying "*Physical Address Extension*".

   The procedure to disable PAE depends on your computer's configuration. To disable PAE on a computer that is running a PAE kernel, add the **/NOPAE** switch to the **boot.ini** file. To disable it on a computer that is running a PAE kernel with DEP, add the **/NOPAE /NOEXECUTE=alwaysoff** switch to the **boot.ini** file.

   See the following websites for more information about PAE and DEP, including specifics about disabling PAE:

   - http://www.microsoft.com/whdc/system/platform/server/PAE/PAEdrv.mspx
   - http://www.microsoft.com/technet/prodtechnol/winxppro/maintain/sp2mempr.mspx
   - http://support.microsoft.com/?kbid=895575

The following section, titled *SFU Pathnames*, contains background information required for installing McIDAS-X for XP.

SFU Pathnames

Unlike Unix, Windows XP does not have a single rooted file system. Rather than mounting a drive on any arbitrary mount point in a file system, the drive must be mounted on a drive letter (e.g., C: or D:). Because SFU uses the XP file system, the pathnames to files may look different than they do in Unix.

There are two types of pathnames on a XP system running SFU: Win32 and POSIX. The Win32 format (e.g., **c:\SFU\usr\bin**) is the standard format used in Windows and DOS environments. It is used in all XP Win32 applications. The POSIX format (e.g., **/dev/fs/C/SFU/usr/bin**) is the standard format used in Unix environments. It is used with SFU applications, including McIDAS-X for XP.

For more information regarding POSIX pathnames and SFU, consult the SFU online documentation.

The installation instructions appear below. The system administrator or person who installs McIDAS-X for XP will use these installation instructions, which consist of five tasks:

- Preparing the mcidas Account on Windows XP

- Obtaining McIDAS-X for XP
- Installing McIDAS-X for XP
- Removing the Previous Version of McIDAS-X for XP
- Preparing a User Account on Windows XP

If you have questions about the installation, call the McIDAS Help Desk at (608) 262-2455, or send email to McIDAS User Services .

Preparing the mcidas Account on Windows XP

If the workstation does not have a **mcidas** account, have the system administrator make and configure the account by completing the steps below.

1. Log on to the workstation as **Administrator**.

2. Create the **mcidas** account if it doesn't already exist.

3. Set the Local Path for the **mcidas** account to **c:\home\mcidas**. To do so, right-click on *My Computer* then click **Manage** to bring up the *Computer Management* window. Then double-click the following in order: **Local Users and Groups**, **Users**, **mcidas** to bring up the *mcidas Properties* window. Click its **Profile** tab, select the radio button next to **Local path** and enter **c:\home\mcidas** in its text field and click **OK**.

   If this workstation is a member of a domain controlled by an Active Directory server, the network administrator will need to configure the home directory for the **mcidas** account on the server to match the one used here.

4. Create a link to **/dev/fs/C/home**, set ownership of the **mcidas** user's home directory and give everyone Read access. To do so, start an SFU Korn Shell and run the four commands below in order.

   Type: **ln -s /dev/fs/C/home /dev/fs/C/SFU**

   Type: **cd /home**

   Type: **chown** *mcidas mcidas*

   Type: **chmod 755** *mcidas*

Obtaining McIDAS-X for XP 2012.2

Note: For convenience, the McIDAS-X for XP installation files now include both the McIDAS-X core package and the McIDAS-XRD sidecar package (see the McIDAS-XRD Software page for details). Installation of McIDAS-XRD is still optional (done by specifying a non-default argument during the install procedure).

File Name	Description
mcidas2012.2_xp.sh	shell script that installs McIDAS-X and/or -XRD
mcidas2012.2_xp.tar.gz	gzip-compressed file containing the McIDAS-X and -XRD software
mcidas.mcenv	environment variables used by McIDAS-X
mcidas.profile	template for creating a **.profile** file in user accounts that don't have a pre-existing **.profile**

Complete the instructions below to obtain the McIDAS-X for XP 2012.2 installation files from the McIDAS Website.

1. Use your Web browser to download the files listed above. Access the McIDAS Website, and follow the link for the McIDAS-X for XP 2012.2 software. Each site has its own login and password for downloading files. Have your site coordinator contact the McIDAS Help Desk if you can't remember yours.

2. Log on to the McIDAS-X workstation as user **mcidas** and copy or move the downloaded files to the **~mcidas** directory.

Installing McIDAS-X for XP 2012.2

Note: For convenience, the McIDAS-X for XP installation files now include both the McIDAS-X core package and the McIDAS-XRD sidecar package (see the McIDAS-XRD Software page for details). Installation of McIDAS-XRD is still optional (done by specifying a non-default value for the *packages* parameter in step 3 below).

Complete the steps below to install McIDAS-X for XP 2012.2 and/or McIDAS-XRD for XP 2012.2. Run each command from the **~mcidas** directory.

1. Log on to the workstation as **mcidas** and start an SFU Korn Shell.

   If there is a previous version of the package(s) you will be installing (McIDAS-X for XP and/or McIDAS-XRD for XP) already installed, go to step 2. If this is the initial installation of the package(s), go to step 3.

2. Uninstall the previous version of the McIDAS-X and McIDAS-XRD packages.

   First verify that no McIDAS-X sessions are running. Then, if the previous version is 2009.1 or later, enter the command below to uninstall the previous version of the packages, replacing *version#* with the number of the version (for example, 2012.1). If the previous version is 2008a or earlier, complete the uninstall procedure (step 2) in the 2009.1 version of this manual. Then return to these instructions and proceed to step 3 below.

   Type: **sh ./mcidas***version#***_xp.sh uninstall all**

   The messages below appear if the uninstall is successfully completed.

```
./mcidasversion#_xp.sh: McIDAS-X uninstall completed normally.
./mcidasversion#_xp.sh: McIDAS-XRD uninstall completed normally.
```

3. Install the desired package (McIDAS-X for XP 2012.2 and/or McIDAS-XRD for XP 2012.2) using the appropriate command below. If installing McIDAS-X for XP (by specifying **x** or **all** in the *packages* parameter), this step first makes the **~mcidas** installation directories and **~mcidas/mcidas2012.2** package directories (see the tables below). Then, it uncompresses the binary files, data files, source files, help files, and man pages. Finally, it creates hard links from the files in the **~mcidas/mcidas2012.2** package directories to the appropriate **~mcidas** installation directories.

   Type: **sh ./mcidas2012.2_xp.sh install** *packages*

   where *packages* is one of the three options described below.

   - **x**
     The **x** option builds only the McIDAS-X for XP package.
   - **xrd**
     The **xrd** option builds only the McIDAS-XRD for XP package.
   - **all**
     The **all** option builds both the McIDAS-X for XP and McIDAS-XRD for XP packages, in that order.

   The message below is displayed if McIDAS-X for XP is selected (by specifying **x** or **all** in the *packages* parameter) and the install is successfully completed.

```
./mcidas2012.2_xp.sh: McIDAS-X install completed normally.
```

   If McIDAS-XRD for XP is selected (by specifying **xrd** or **all** in the *packages* parameter) and the install is successfully completed, the message is the same as that above, except *McIDAS-X* is replaced with *McIDAS-XRD*.

Installation Directories

The install process makes hard links (as opposed to symbolic links) between the installation directories and the files in the package directories. The uninstall process simply removes the hard links. The McIDAS-X installation directories and their contents are listed below.

Directory	Contents

~mcidas/bin	program executables
~mcidas/data	data files
~mcidas/help	help files
~mcidas/inc	include files
~mcidas/lib	libraries
~mcidas/man	man files for subroutines and functions
~mcidas/tcl	Tcl and Tk executables and libraries

Other McIDAS packages also make hard links in the installation directories. Thus the installation directories are the repositories for all McIDAS packages currently installed.

Package Directories

Each version of McIDAS-X and other McIDAS packages (for example, McIDAS-XCD) builds its own set of directories. The names of the directories depend on the package name and version number. The McIDAS-X for XP package directories and their contents are listed below.

Directory	**Contents**
~mcidas/mcidas201*x*/bin	McIDAS-X binary executables
~mcidas/mcidas201*x*/data	McIDAS-X data files
~mcidas/mcidas201*x*/help	McIDAS-X help files
~mcidas/mcidas201*x*/inc	McIDAS-X include files
~mcidas/mcidas201*x*/lib	McIDAS-X library files
~mcidas/mcidas201*x*/man	McIDAS-X man files for subroutines and functions
~mcidas/mcidas201*x*/src	McIDAS-X source files
~mcidas/mcidas201*x*/tcl	Tcl/Tk 8.5.9 files for the GUI

Removing the Previous Version of McIDAS-X for XP

In versions 2009.1 and later, the McIDAS-X for XP installation files include both the McIDAS-X for XP core package and the McIDAS-XRD for XP sidecar package.

If you have installed and tested a new version of McIDAS-X for XP and/or McIDAS-XRD for XP and are satisfied they are working properly, you can remove the previous version of the packages. If the previous version is 2009.1 or later, complete the numbered instructions below, running each command from the **~mcidas** directory. If the previous version is 2008a or earlier, complete the unmake procedure(s) in the 2009.1 version of this manual. Then return to this manual and proceed with its instructions.

1. Log on to the workstation as **mcidas** and start an SFU Korn Shell.

2. Remove the **mcidas***version#* directory tree and its contents, replacing *version#* with the previous version number.

   Type: **sh ./mcidas***version#***_xp.sh unmake all**

The messages below are displayed when the unmake is successfully completed.

```
./mcidasversion#_xp.sh: McIDAS-X unmake completed normally. ./mcidasversion#_xp.sh:
McIDAS-XRD unmake completed normally.
```

3. Remove the following files:

- **mcidas***version#***_xp.sh**
- **mcidas***version#***_xp.tar.gz**

Preparing a User Account on Windows XP

The McIDAS-X software is installed in the **mcidas** account. However, you can not run McIDAS-X as user **mcidas**. Individual user accounts must be created and configured for running McIDAS-X. The four tasks below must be completed for each user account.

- Making and Configuring the User Account
- Setting the SFU Environment
- Setting the Windows XP Environment
- Setting the X Server Properties

The system administrator must complete the first task. The system administrator or user can complete the second, third and fourth tasks.

Making and Configuring the User Account

1. Log on to the workstation as **Administrator**.

2. Create the user account if it doesn't already exist.

3. Set the Local Path for the user to their home directory, e.g., **c:\home\johndoe**. To do so, right-click on *My Computer* then click **Manage** to bring up the *Computer Management* window. Then double-click the following in order: **Local Users and Groups**, **Users**, *user*, where *user* is the user's account name, to bring up their *user Properties* window. Click its **Profile** tab, select the radio button next to **Local path** and enter the user's home directory, e.g., **c:\home\johndoe**, in its text field and click **OK**.

   If this workstation is a member of a domain controlled by an Active Directory server, the network administrator will need to configure the home directory for the *user* account on the server to match the one used here.

4. Create a link to **/dev/fs/C/home**, set ownership of the user's home directory and give everyone Read access. To do so, start an SFU Korn Shell and run the four commands below in order, replacing *user* with the user's account name.

   Type: **ln -s /dev/fs/C/home /dev/fs/C/SFU**

   Type: **cd /home**

   Type: **chown** *user user*

   Type: **chmod 755** *user*

Setting the SFU Environment

1. Log on to the user account and start an SFU Korn Shell.

2. Verify that the files **mcidas.profile** and **mcidas.mcenv** are present in the **~mcidas** directory, as specified in *Obtaining McIDAS-X for XP*.

   Type: **ls -l ~mcidas/mcidas.profile ~mcidas/mcidas.mcenv**

The **mcidas.profile** file may be used in step 6 below as a template for creating a **.profile** in user accounts that don't have a pre-existing **.profile**. The **mcidas.profile** file is a modified version of the **profile.usr** file supplied by SFU in the **/dev/fs/C/SFU/etc** directory. The **PATH** was modified to include the required McIDAS directories and a line was added to source the **.mcenv** file.

The **mcidas.mcenv** file contains the environment variables used by McIDAS-X for XP. It is copied to **.mcenv** in step 7 below and is sourced (run) by **.profile** each time the user logs on. Its contents may be modified by the user to their individual preferences in step 8 below.

3. Change to the SFU home directory of the user account.

   Type: **cd $HOME**

4. Copy the **mcidas.profile** and **mcidas.mcenv** files to the current directory.

   Type: **cp ~mcidas/mcidas.profile .**

   Type: **cp ~mcidas/mcidas.mcenv .**

5. Modify or create the **.profile** file as follows. If the file does not exist, go to step 6 to create it. If it already exists, add the following two lines (substitute the equivalent if using C shell) to the **PATH** environment variable section.

PATH=$HOME/mcidas/bin:~mclocal/mcidas/bin:~mcidas/bin:$PATH

export PATH

Also, add the following line anywhere in the file; then, go to step 7.

. $HOME/.mcenv

6. Create the **.profile** file by copying the **mcidas.profile** file to .**profile**.

   Type: **cp mcidas.profile .profile**

7. Copy the **mcidas.mcenv** file to **.mcenv**.

   Type: **cp mcidas.mcenv .mcenv**

8. Carefully read the **.profile** and **.mcenv** files and, if necessary, modify their settings and contents to your preferences.

Setting the Windows XP Environment

Complete the tasks below to make the Windows XP environment compatible for running McIDAS-X for XP.

- Setting the Number of Desktop Colors
- Creating a McIDAS-X Shortcut

Setting the Number Of Desktop Colors --McIDAS-X for XP uses 256 colors. You can, however, configure your display for more colors so other Windows applications and the McIDAS COMBINE command can use the additional colors. Complete the steps below to set the palette.

1. Right click on the XP desktop. Then select **Properties** to bring up the *Display Properties* window.

2. Click the **Settings** tab, then select the appropriate value in the *Color quality* pull down list.

3. Click **OK** to activate the change.

Creating a McIDAS-X Shortcut --It is easiest to start McIDAS-X from a shortcut on the XP desktop. Complete the steps below to create a McIDAS-X shortcut.

1. Right click on the XP desktop. Then select **New**, and click **Shortcut** to bring up the *Create Shortcut* window.

2. In the *Type the location of the item:* field, insert the following command (paying careful attention to forward and backward slashes), then click **Next >**.

```
%WINDIR%\System32\posix.exe /u /c /bin/ksh -l -c mcidas
```

3. Type a name for the shortcut, e.g., *McIDAS*, then click **Finish**.

4. A shortcut will appear on the desktop. This shortcut can then be copied anywhere. For example, if you want McIDAS-X to start automatically at login, copy the shortcut icon to your *Startup* folder. Note that you must have an X server running before you start McIDAS-X. Continue with the following section to set the X server properties.

Setting the X Server Properties

You must have an X server running in order to display the McIDAS-X for XP windows. We do not recommend a specific X server for use with McIDAS-X for XP. So we cannot identify the exact X server configuration changes you will need to perform to make it compatible with McIDAS-X for XP. However, many X servers require the changes described below.

- Emulating a Three-Button Mouse
- Mapping the Keyboard

Emulating a Three-Button Mouse --McIDAS-X for XP is designed for systems with a three-button mouse, as several commands require you to press the middle of the three buttons. If you have a two-button mouse, you must configure your X server to emulate a three-button mouse. Use the mouse section of your X server's configuration program to do so.

Mapping the Keyboard --Modification of the X server's keyboard mapping is sometimes required to enable certain keys used by McIDAS-X (for example, **Alt A**, **Ctrl F11**). Use the keyboard section of your X server's configuration program to select the proper keyboard mapping for your X server.

This concludes the instructions for installing McIDAS-X for XP and configuring user accounts. Continue with *Configuring McIDAS-X Sessions with .mcidasrc*, which provides information about tailoring McIDAS-X sessions to your preferences.

Configuring McIDAS-X Sessions with .mcidasrc

Once you have installed McIDAS-X and configured your accounts, you can run McIDAS-X. See Chapter 2 for information about starting, using and ending McIDAS-X sessions.

The first time you run McIDAS-X, the **$HOME/.mcidasrc** configuration file is created. This file stores flags that configure the McIDAS-X session to your preferences. This section contains descriptions of the flags and how to modify their values using a text editor. Nearly all of the flags can also be changed using the McIDAS-X Configuration GUI described in *Starting McIDAS-X* in Chapter 2.

The following tables list the flags for changing the Image, and Text and Command Windows settings. The flags can appear in any order in **.mcidasrc**. However, the file is organized in a manner you may want to follow. Default values, if present, are indicated in the tables or in the detailed descriptions that follow.

Image Window Flags	**Description**
-autoResize	automatically resizes the Image Window to the frame size
-displayVisualMode	runs McIDAS-X using the visual mode set for the workstation display
-e *amount***k -e** *amount***m**	allocates a memory pool of amount kilobytes or megabytes for the MAKFRM command
-f *framespec*	sets the number and size of frames
-graphicsColors *number*	sets the number of graphics color levels (default = 16)
-igeometry *line+element*	sets the location of the Image Window
-imageColors *number*	sets the number of display levels (default = 128)
-manualResize	resizes the Image Window to the frame size only when you use the R command (default)
-optimizeForSpeed	displays images as quickly as possible (default)
-optimizeForSpace	allows the maximum number of frames for a given amount of memory, but displays images more slowly than **-optimizeForSpeed**

The table below lists the start-up flags for specifying the Text and Command Window setup. See the following sections for more information.

Text and Command Window Flags	**Description**
-c *'command'*	runs a specified command at startup
-cd *directory*	sets the working directory for the McIDAS-X session
-display *workstation***:0**	specifies the workstation for displaying the McIDAS-X session
-font *'font'*	sets the font used in the Text and Command Window
-fontListHead *'fonts'*	specifies the fonts for the beginning of the font list for the Text and Command Window
-fontListTail *'fonts'*	specifies the fonts for the end of the font list for the Text and Command Window
-ih *number*	sets the number of commands to store in the recall buffer and file **MCCMDHIST.TXT** (default=100)
-oh *number*	sets the maximum number of output lines in each text frame (default=1000)
-tcolbg *value 'color'*	sets colors for the background attributes in the McIDAS Text and Command Window
-tcolfg *value 'color'*	sets colors for the foreground attributes in the McIDAS Text and Command Window
-tgeometry *width***x***height***+***line***+***element*	sets the size and location of the McIDAS Text and Command Window

The following sections provide more information about changing your McIDAS setup using the start-up flags listed above:

- Setting the Number of Frames
- Allocating Additional Frame Memory

- Running Commands at Startup
- Setting the Text Fonts
- Setting the Command and Output Recall
- Setting the Workstation Display
- Setting the Working Directory
- Setting the Configuration GUI Option
- Setting the McIDAS Text and Command Window Options
- Setting the McIDAS Image Window Options
- Using the MCIDASRC Environment Variable

Setting the Number of Frames

Use the **-f** *framespec* flag to set the number and size of the frames for your McIDAS-X session. You can use the following formats for *framespec*.

Format	Description
number	creates a specified number of 480x640 frames
*lines***x***elements*	creates a frame of specific size
*number@lines***x***elements*	creates a number of frames of a specific size

By default, the **.mcidasrc** file contains the line **-f 10**, meaning that McIDAS-X sessions start with ten frames that are 480 lines by 640 elements. You can add additional sets of frames of different sizes. For example, if you add the line **-f 4@512x512 -f 4@1024x1280**, sessions will have 18 frames: four 512 by 512 frames; four 1024 by 1280 frames, and ten 480 by 640 default frames.

The order of the *framespec* options determines the frame numbers. The frames are numbered starting with 1. For example, if you start your session with **-f 6 -f 512x512 -f 512x512 -f 2@1024x1280**, your session has 20 frames with the following numbers and sizes.

- frames 1-10; 480 lines by 640 elements (default frames)
- frames 11-16; 480 lines by 640 elements
- frames 17-18; 512 lines by 512 elements
- frames 19-20; 1024 lines by 1280 elements

You can remove the default by commenting out the line **-f 10** in your **.mcidasrc** file.

Allocating Additional Frame Memory

Use the **-e** *count* option to allocate a memory pool for creating additional frames with the MAKFRM command. See the *Image Window* section and the *Adding Frames* section in Chapter 2 for more information.

The default size of the **-e** memory pool is 0 bytes. Use a **k** to specify the number of bytes as kilobytes or an **m** to specify the memory in megabytes. For example, to allocate a 5 MB memory pool for MAKFRM, use the flag **-e 5m**. The amount of memory required to add a new frame is slightly more than the number of lines times the number of elements. The figure is slightly higher because additional memory is required for enhancements, stretch tables, graphics tables, etc.

Running Commands at Startup

You can specify commands to run automatically at startup using the**-c** '*command*' flag. Specify this flag at startup or in your **.mcidasrc** file as many times as needed to run any set of commands. Each **-c** *command* flag is run asynchronously from all other **-c** flags. To run certain commands in a sequence, specify them all in one **-c** flag. For example, the command **-c** '**ECHO{HELLO};ECHO** "**THERE** ' runs two ECHO commands in the sequence they are entered.

To run the commands from the Unix shell, prefix the command with an exclamation point (!). See the exclamation point (!) command in Chapter 3 for more information. If you do not want the command output displayed in the Text and Command Window, add an at symbol (@) after the exclamation point (!).

Setting the Text Fonts

The **-font** option sets the font that the Text and Command Window automatically uses. The fonts are listed by pixel size and the default font is 9x15 bold. You can specify a list of fonts for the Text and Command Window using the **-fontListHead** '*fonts*' and **-fontListTail** '*fonts*' flags. Once you set the fonts with these flags, you can change the font used in the Text and Command Window using the **Alt <** and **Alt >** keys.

You can specify multiple fonts for the **-fontListHead** and **-fontListTail** flags, but separate them with commas, for example, **-fontListHead '5x8,6x9,6x13bold'**. The **.mcidasrc** file lists some sample fonts.

Setting the Command and Output Recall

The **-ih** *number* flag sets the number of commands that you can recall via the ampersand (&) and caret (^) keys in the McIDAS Text and Command

Window, and via the Command History GUI under the main GUI window's *List* menu. The default is 100. You can also access the commands, which are stored in the file **MCCMDHIST.TXT**, via the SEE command or a text editor for cutting and pasting.

The **-oh** *number* flag sets the number of output lines to store in each text frame in the McIDAS Text and Command Window. The default is 1000. Use the Page Up, Page Down and arrow keys or the McIDAS-X command TFIND to scroll the output lines.

Setting the Workstation Display

The **-display** *workstation*:**0** flag sets the workstation display for the McIDAS-X session. Typically, you will not have to set this at startup because it is in your **.profile** file. This flag is useful for displaying the McIDAS-X session on a workstation not listed in your **.profile**.

Setting the Working Directory

The **-cd** *directory* flag sets the working directory for the McIDAS-X session. Therefore, Unix commands that are run from the Text and Command Window (e.g., **!ls -l AREA*** or **OS "pwd**) will use it as the current directory and thus give the same output that they would if run at a Unix prompt while in that directory. If this flag is not set, the working directory is set to the directory in which McIDAS-X was started.

Setting the Configuration GUI Option

The **-config** flag makes the McIDAS-X Configuration GUI appear each time you start a McIDAS-X session. The Configuration GUI, which can also be started by running the command **mcidas -config** at the Unix prompt, contains sliders, checkboxes and text fields that let you override the settings of the corresponding flags in **.mcidasrc**. It makes one-time (for the current session only) configuration changes easier and less error-prone. It also contains an option to save its settings in **.mcidasrc**. See *Starting McIDAS-X* in Chapter 2 for more information about the Configuration GUI.

Setting the McIDAS Text and Command Window Options

You can specify the following start-up options for the McIDAS Text and Command Window.

- size and location
- colors

To specify the size and location of the McIDAS Text and Command Window use the **-tgeometry** *geometry* flag, where *geometry* is the character width and height of the window and the line and element location. For example, to create a Text and Command Window that is 85 characters by 26 characters, located at line 300 and element 400, enter the flag **-tgeometry 85x26+300+400**.

You can specify colors to use for the Text and Command Window borders, backgrounds, command text, error messages, etc. To change the display, use the **-tcolfg** *value'color'* and **-tcolbg** *value'color'* flags. The **-tcolbg** flag sets colors for the background attributes in the McIDAS Text and Command Window and **-tcolfg** sets the attributes for the foreground. Typically, the foreground attributes are text and the background attributes are window backgrounds.

The table below lists the *values* for several McIDAS Text and Command Window attributes. The *value* specifies the attribute to change. The colors shown in the Default Color column are the initial **.mcidasrc** file defaults. For example, to change the color of error messages to yellow, use the flag **-tcolfg14 'yellow'**. The **.mcidasrc** file provides a complete list of the attributes that you can change and their corresponding *values*.

Window Attribute	Value	Default Color
local text	7	white
echoed commands	10	green
command editor	11	cyan
communications status messages	13	magenta
error messages	14	yellow
line continuation marks	96	yellow

Setting the McIDAS Image Window Options

You can specify the following start-up options for the Image Window.

- location
- automatic resizing
- graphics colors and image levels
- display speed and memory use

To set the location of the McIDAS Image Window, use the **-igeometry** *geometry* flag where geometry is the line and element location. For example, to start a session with the Image Window in the center, use the flag **-igeometry 768+1024**.

The **-manualResize** flag is the default resize flag. This flag requires you to use the R command to resize the Image Window to the size of the

displayed frame. To automatically resize the Image Window to the size of the frame, specify the **-autoResize** flag.

Use the **-graphicsColors** *number* option to change the number of graphics colors available. The default number of graphics colors is 16; the maximum is 128. Use the **-imageColors** *number* option to change the number of display levels available. The default number of display levels is 128; the maximum is 253 or 254, depending on your hardware. The more display levels you specify, the less graphics colors there are available and vice versa.

The **-optimizeForSpeed** and **-optimizeForSpace** options change the amount of memory used for displaying images and loops. The **-optimizeForSpeed** option, which is the default, displays images as quickly as possible. This option requires a large amount of memory. The **-optimizeForSpace** option displays images using a minimum amount of memory. This option greatly slows looping, but it more than doubles the amount of frames that can be created for a given amount of memory. Because the looping is slow, you can only use this option if you are running a local McIDAS-X session on your workstation.

The **-displayVisualMode** option forces McIDAS-X to run in the visual mode set for the workstation display, e.g., TrueColor, StaticColor, etc. The default (when this flag is not set) is to run McIDAS-X in the 8-bit PseudoColor mode. Set this flag only if your display does *not* support the 8-bit PseudoColor mode, as setting it may significantly slowdown commands that write image or graphic output.

Using the MCIDASRC Environment Variable

The **MCIDASRC** environment variable lets you store and use multiple versions of the **.mcidasrc** session configuration file. Using the **MCIDASRC** environment variable is most beneficial to users who often change session configurations or those who have several users that run McIDAS-X from the same account. In cases like these, you simply create a different configuration file for each user and/or purpose, then use **MCIDASRC** to select the desired file for a given McIDAS-X session. Maintaining multiple configuration files like this is easier and less error-prone than constantly changing your **.mcidasrc** file.

For example, if you have configuration files named **.mcidasrc**, **.mcidasrc_jeff**, **.mcidasrc_mary**, **.mcidasrc_sue** and **.mcidasrc_tom** in in your **/home/team1/** directory, enter the following Unix commands (substitute the equivalent if using C shell) to start a McIDAS-X session using the specified configuration file (e.g., **.mcidasrc_jeff** or **.mcidasrc_mary**).

Type: **export MCIDASRC; MCIDASRC=/home/team1/** *filename*

Type: **mcidas**

Registering MD File Schemas

Each version of McIDAS-X includes copies of the LW files **DCGFSMOS**, **DCGRET**, **DCIHGT**, **DCIRAB**, **DCISEN**, **DCISFC**, **DCISHP**, **DCNAMMOS**, **DCNGMMOS**, **DCPIRP**, **DCRAOB**, **DCRAWI**, **DCSVCA**, **DCSYN** and **DCUPPR** in the **~mcidas/data** directory. These files define the GFS, GRET, IHGT, IRAB, IRSG, ISEN, ISFC, ISHP, NAM, NGM, PIRP, RAOB, RSIG, RAWI, SVCA, SYN, and UPPR MD file schemas.

When you register a schema using McIDAS-X command SCHE (in the *Commands* section of this manual), the schema file is read and the schema's internal form is copied into LW file **SCHEMA**, which resides in the **$HOME/mcidas/data** directory. Command SCHE can also add locally developed schemas to the workstation.

Commands that read data in an MD file (for example, PTDISP, SFCLIST, UACROSS) or edit a preexisting MD file (for example, MDE) do not require the MD file's schema type to be registered. However, commands that create an MD file (for example, TXT2MD, ISENT or MDU MAKE) do not work unless the schema is registered. Therefore, you should register the supplied schemas using the SCHE command after installing McIDAS-X for the first time.

Once a schema is registered on the workstation, you don't have to register it each time you install a new version of McIDAS-X. When a schema is updated, it is noted in the *McIDAS-X Upgrade Procedure* document supplied by SSEC for your site coordinator with McIDAS-X upgrades.

To register the MD file schemas, first complete the software installation procedure. Then start McIDAS-X and use the McIDAS-X command SCHE to register each schema.

For example, to register the schema ISFC,

Type: **SCHE DCISFC**

To register the schemas IRAB and IRSG,

Type: **SCHE DCIRAB**

Use the list schema command, LSCHE, to verify the registered schema types and version numbers or to search the schemas for a specific key.

Chapter 2

Introduction

This chapter provides general information about McIDAS-X. It consists of these sections:

- Conventions
- Workstation Characteristics
- Starting and Ending a McIDAS Session
- Entering and Editing Commands
- Displaying Images and Graphics
- Using Strings and Function Keys
- Using the McIDAS-X Graphical User Interface
- Using the ADDE
- Getting Started with ADDE

Conventions

The following typographical conventions are used in this chapter:

- **bold** - McIDAS keyboard entries appear in bold; type these entries exactly as they appear
- *italics* - variables that you must enter, such as command parameters, appear in italics
- **Courier bold** - file names, system variables, directories, paths, program names and operating-system-level commands (Unix) appear in this font

---

Workstation Characteristics

The chart below shows the McIDAS-X workstation characteristics.

Characteristic	**Comment**
Gray or color levels (user-defined) - image - graphics	 up to 253 or 254 * up to 128
Bit plane graphics	yes
Frame size	user-defined
Add frames dynamically	yes
Zoom and roam	yes
Loop rate (frames/second)	hardware-dependent
Run commands outside the McIDAS session	yes
Run multiple sessions on one workstation	yes
Remote operation	yes
NTSC capable	yes, but requires additional hardware
* depends on hardware	

Starting and Ending a McIDAS Session

This section describes how to start and end a McIDAS-X session. It consists of these sections:

- Starting McIDAS-X
- Ending McIDAS-X

Starting McIDAS-X

To start a McIDAS-X session, log on to your account. Then, run one of the following three commands from a Unix prompt.

Type: **mcidas**

or

Type: **mcidas** *flags*

or

Type: **mcidas -config**

The first command (**mcidas**) starts a McIDAS-X session using the current settings of the flags in the **.mcidasrc** file. The flags and their defaults are described in *Configuring McIDAS-X Sessions with .mcidasrc*. If McIDAS-X has not been previously run in your account, the **.mcidasrc** file does not exist. In this case, the **mcidas** command displays the McIDAS-X Configuration GUI (described below) and creates the **.mcidasrc** file. You can have the Configuration GUI appear each time you start a session with the **mcidas** command by adding the **-config** flag to your **.mcidasrc** file.

The second command (**mcidas** *flags*) is similar to the first, except it allows you to override one or more of the flags' settings in the **.mcidasrc** file by specifying the flags on the command line. This *flags* option makes it easy to override selected settings in **.mcidasrc** for the current session. Without the *flags* option, you would have to change the settings in **.mcidasrc**, start a session with the **mcidas** command, then edit **.mcidasrc** again to change the settings back to their previous values.

The third command (**mcidas -config**) displays a McIDAS-X Configuration Graphical User Interface (GUI) window with sliders, checkboxes and text fields that are used to select values that override the settings of the corresponding flags in the **.mcidasrc** file. This Configuration GUI is designed to make one-time (for the current session only) configuration changes easier and less error-prone than other methods like changing flag settings in the **.mcidasrc** file or specifying the flags in the command line like the **mcidas** *flags* option above. The Configuration GUI also includes a "Save Settings" button to save the GUI settings in the **.mcidasrc** file. However, caution should be used with this option because it replaces the current **.mcidasrc** file so its settings are lost. To make recovery easier, the option first moves the current **.mcidasrc** to **.mcidasrc.old** before creating the new **.mcidasrc** with the GUI settings.

When a McIDAS-X session is started by any of the three methods above, the following then occur:

- Two windows are displayed: the Text and Command Window and either the Image Window or the McIDAS-X Graphical User Interface, depending on the flags set in the **.mcidasrc** file. See the sections *Entering and Editing Commands* and *Displaying Images and Graphics* for more information about these windows.
- Shared memory segments are created, as is a uniquely named temporary directory in **$HOME/.mctmp** to store session files. McIDAS-X also creates files such as **FRAMED.001** in the temporary directory and adds the name of the temporary directory to **MCPATH** to make the files visible to the rest of McIDAS-X. The directory and its files are deleted when you end the McIDAS-X session.
- The **mcclean** program is run to delete any inactive directories that it finds in **$HOME/.mctmp** and remove any shared memory segments associated with the directory. Typically, there will not be any inactive temporary directories, but if a session ends in a catastrophic manner (e.g., with **kill -9**) the temporary directory may not be deleted.

Running multiple McIDAS-X sessions

Though not recommended because of possible file conflicts, you can run multiple McIDAS-X sessions from the same account. Multiple sessions include those started with the **mcidas** script and McIDAS-X commands run from the **mcenv** process or the Unix scheduler. For more information on using the **mcenv** process or Unix scripts, see *Appendix H, Running Commands Outside a McIDAS-X Session*. To differentiate between the files used for various sessions, each session stores many of its files in its own **$HOME/.mctmp** subdirectory. When you run multiple sessions, the **mccheck** program warns you if there are files in a **MCPATH** directory that belong in a **.mctmp** subdirectory. The **mccheck** program only provides warnings when you run multiple sessions.

File conflicts are still possible because certain files and directories may be used by all sessions. For example, the default scheduler file, **SKEDFILE**, is stored in **$HOME/mcidas/data** (not a **.mctmp** directory) because it often contains entries to run in subsequent sessions. Therefore, if you are running multiple sessions and the scheduler is on (SKU command), anything scheduled to run in one session runs in all of them. If you need to set different schedulers in separate sessions under the same account, specify the **-c 'SKED** *file*' flag when starting each session. The *file* should be a different scheduler file for each session. Then use keyword FILE to specify the different scheduler files in the SKE and SKU commands.

Ending McIDAS-X

To end the session, use the McIDAS-X command EXIT. It closes the Text and Command, and Image Windows, and ends all McIDAS processes. Running EXIT from the McIDAS command line is a cleaner way of ending the session than manually closing each window with the mouse.

Entering and Editing Commands

This section describes how to enter and edit McIDAS commands. It provides information about:

- The Text and Command Window
- Entering Commands

Text and Command Window

Use the Text and Command Window for entering McIDAS commands, displaying text output, and displaying workstation status information. You can reposition the Text and Command Window to a different location on the display and you can also resize it.

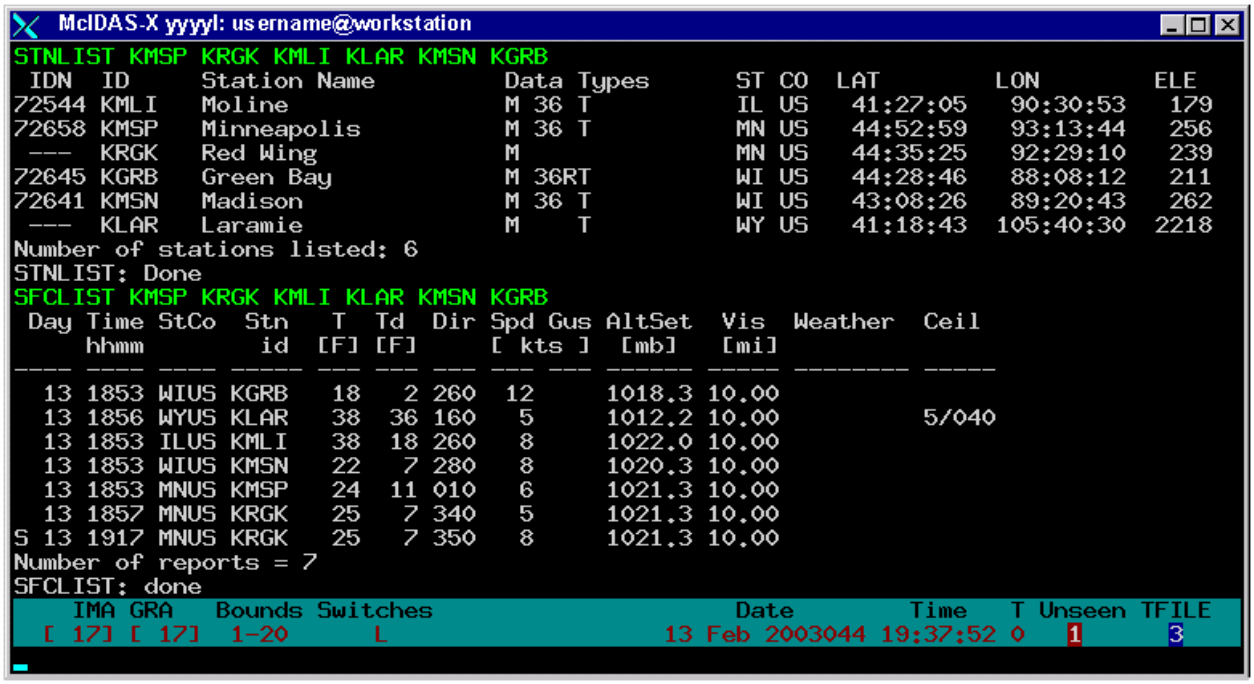
The Text and Command Window displays keyboard input regardless of whether it or the Image Window is active. Click in either window to make it active.

Use the numeric keypad's plus (+) key to raise the Image Window. To raise the Text and Command Window, press a numeric keypad number. The number you press determines which text frame is displayed in the window. On some workstations, you may have to first enable the **Num Lock** key or press the **Alt Graph** key and a numeric keypad number at the same time.

The Text and Command Window consists of these parts:

- Title Bar
- Text Frame
- Command Line
- Status Bar

The McIDAS-X Text and Command Window is shown below.



Title Bar

The title bar displays the version number of the package, the user's logon name and the workstation name.

Text Frame

As you type a command, the characters appear on the command line. When you press Enter, the command is echoed in the current text frame. If the command's output is text, it is displayed in the current text frame.

Ten text frames, numbered 0-9, are available for displaying text. Only one text frame can be displayed at a time. To view a text frame, press its number in the numeric keypad while the Text and Command Window is active. For example, pressing the 3 key on the numeric keypad displays text frame 3. If you are running McIDAS-X on a Sun workstation, you may have to first enable the Num Lock key, or press the Alt Graph key and the number at the same time.

Text frames 0-9 can contain thousands of lines. Specify the number lines in your **.mcidasrc** file. See Chapter 1 for more information. To scroll text output, use the Page Up, Page Down, and the up (↑) and down (↓) arrow keys. You can also use the TFIND command to search for specific expressions in the text frame.

Commands that produce text output allow you to specify a text frame number using the global keyword TWIN. For example, entering command

IMGLIST A/A.1 10 TWIN=1 while text frame 5 is raised overrides the default and directs the IMGLIST command output to text frame 1. If you direct the command's output to a nonactive frame, its number is displayed in the status bar's *Unseen* field.

The Text and Command Window is resizeable. Text lines that extend past the edge of the window are flagged with a > character on the right side. Use one of the following methods to view this text:

- increase the width of the window by dragging the sizing border; if you still can not see all of the text, reduce the font size by pressing the **Alt** and **<** keys at the same time; then resize the window again
- press the **Alt** and **]** keys at the same time to scroll to the right in the text frame; press the **Alt** and **[** keys to return to the left side of the text frame

Command Line

As you type a command, the characters appear on the command line at the bottom of the window. See *Entering Commands* for a list of editing keys and conventions that apply.

Status Bar

The status bar portion of the Text and Command Window is displayed above the command line. The status bar displays the current frame numbers, loop bounds, looping status, toggle switches, date, time, and text frame number. This information is updated as the state of your workstation changes; the time is updated every second.

An example status bar is shown below, and is described in the table that follows.

Field	Example	Description
(Edit Mode)	I	The command edit/entry mode: insert or overwrite. An *I* in this field means that whatever you type is inserted in the command line at the cursor position. If this field is blank, the overwrite mode is on; whatever you type, replaces characters at the cursor position.
IMA	1	The image frame currently being viewed.
GRA	1	The graphics frame currently being viewed.
Bounds	1-2	The image frame loop bounds. The word *random* appears here if random loop bounds have been set using the LS command.
Switches	K W L Z J Y	The active toggle commands. These single-letter commands switch certain display attributes on and off. Press the **Alt** key and the letter key to run the command. Letters appear in this field only when toggle state is not the default. For example, the K command's default state is image frames on. If you set the toggle to image frames off by pressing **Alt K**, the letter K appears in the toggle field. McIDAS-X toggle command defaults are listed below: **Toggle** / **Function** / **Default** K / turns image frames on/off / on W / turns graphics frames on/off / on L / turns looping on/off / off Z / turns zoom on/off / off J / turns graphics looping ability on/off / off Y / turns image looping ability on/off / on
(Text or Command Position)	already at bottom	Informational messages. Normally, there is no message in this field. Messages appear only if you try to scroll off any edge of the Text and Command Window using one of the arrow keys. For example, in the status bar above, the user tried to scroll down when the display already showed the last text line available. Possible messages include the following: • already at bottom • already at top • beginning of line • end of line
Date	31 Oct 2002304	The current date (in the format DD MMM YYYYDDD).
Time	21:07:08	The current time (in the format HH:MM:SS).
T	0	The text frame number currently displayed.
Unseen	2	Text frame containing text you have not viewed yet. The number disappears when you view the frame.

TFILE	4	Text frames activated for text capture using the TFILE command.

Entering Commands

The command line editing keys, command entering protocols, and methods of recalling previously-entered commands are described below.

Editing Keys

Listed below are the command line editing keys available with the default McIDAS command input editor. See the EDITMODE command for other available editors.

Key	Function
Home	moves the cursor to the beginning of the line
End	moves the cursor to the end of the line
Insert	toggles the insert typing mode
Delete	deletes the character over the cursor
Backspace	deletes the character to the left of the cursor
Right/Left arrows	moves the cursor one character to the right or left
Tab, Shift Tab	moves the cursor one parameter to the right or left
Enter	runs the command; if the text frame's bottom line is not displayed, Enter forces the display to the bottom showing the echoed command
Esc	erases the command from the command line and places the cursor at the front of the line
Alt ?	displays an abbreviated help for the current command in the command line
& and ^	recalls a command if the command line is empty or contains a command recalled with & or ^; the & recalls the previous command from a circular recall list; ^ scans the recall list in the opposite direction; the recall list consists of the number specified with the **-ih** flag in the **.mcidasrc** file; see *Recalling Commands* below for additional ways to recall commands for editing

NOTE: The Home, End, right/left arrows, Ins and Del keys on the numeric keypad do not work.

Command Protocols

Use the following protocols when entering McIDAS commands.

- The number of characters is system-dependent, but at least 512.
- Enter most command, parameter and keyword letters in uppercase. Lowercase letters should only be needed when a command uses a single(') or double quote (") to indicate a text field. To enter lowercase letters after the quote, use the Caps Lock key or press the **Shift** key while typing. The state of the Caps Lock light may be incorrect.
- After typing a command, press Enter to run it.
- Run system-defined, single-letter commands by pressing the **Alt** key (**open-apple** key on Mac OS X systems) and letter key at the same time, or by typing the letter and pressing **Enter**. You can enter single-letter commands using the **Alt** key while entering a multiple-letter command in the Text and Command Window.
- Enter single-letter strings with a pound sign (#). Press **Ctrl** and the letter if the single-letter string is a complete command.
- Press **Ctrl** and the left mouse button to echo the latitude and longitude at the cursor position to the command line. This option is useful for entering latitude and longitude values in commands such as IMGDISP.
- Generally, parameters and keywords are separated by one space. McIDAS interprets multiple spaces between parameters and keywords as a single space.
- Run multiple commands in a single entry by concatenating the commands. Concatenate commands by placing a semicolon (;) between commands. For example, SF 3;EG;MAP USA. To concatenate commands that use quote fields, substitute curly brackets ({ }) for the quotes. For example,
  PC L KMSN;ZA 3 10 {Madison};PC L KMIA;ZA 3 10 {Miami}.
- Use the question mark (?) command to display status. Active processes are listed with a Process IDentification (PID) number. To stop a command, use the / (Slash) or KILL command with the appropriate PID value. See Chapter 3 for more information.

Recalling Commands

As described in *Editing Keys* above, the & and ^ keys can be used to recall previously-entered commands from a buffer containing the commands entered via the command line during the current McIDAS session. Those commands, as well as commands entered during previous sessions and entered through the McIDAS-X GUI are also available in the file **MCCMDHIST.TXT**. In the file, each new McIDAS session is indicated with the line *Starting a new McIDAS session on <date and time>*. The number of commands stored in the file is equal to the value set in the **-ih** flag in the **.mcidasrc** file. You can access the commands in **MCCMDHIST.TXT** for recall and editing through the Command History GUI under the main

GUI window's *List* menu or through the SEE command or a text editor for cutting and pasting.

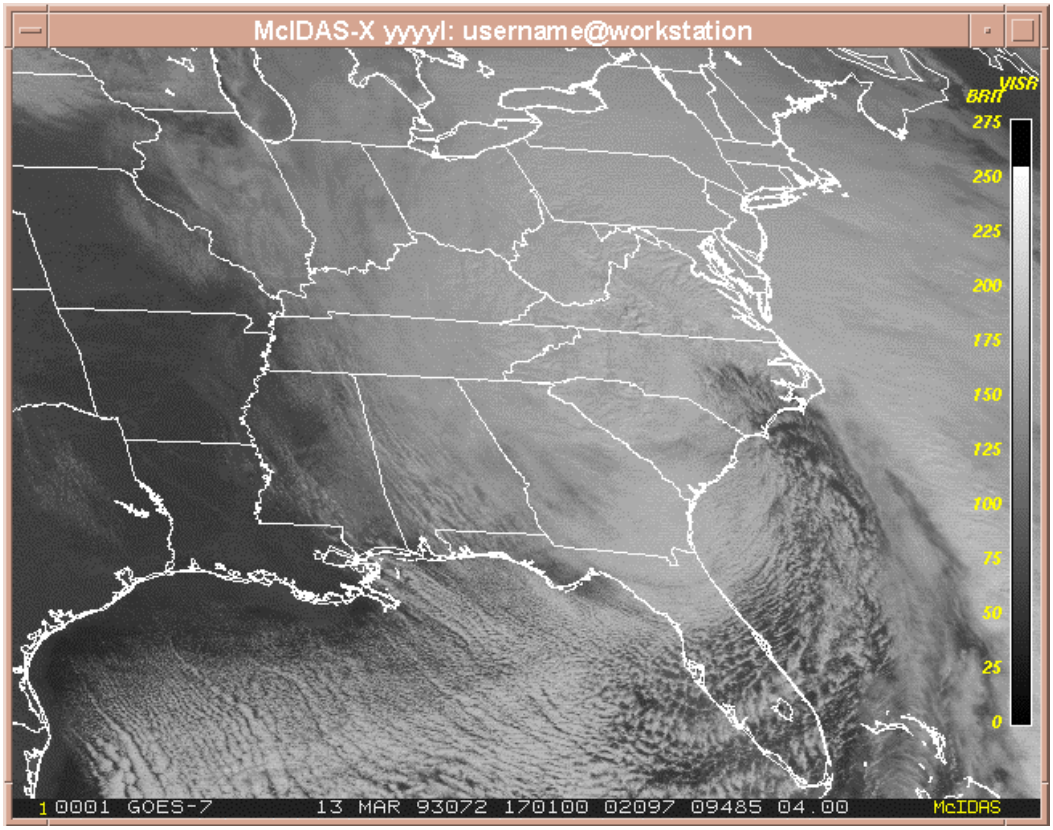
Displaying Images and Graphics

This section explains how to display images and graphics, adjust their configuration settings, and use the McIDAS cursor. It provides information about the following topics:

- Image Window
- McIDAS-X Independent Graphics
- Resizing, Zooming, and Roaming Frames
- Image and Graphics Colors
- Frame Panels
- Adding Frames
- McIDAS Cursor
- Displaying Loops

Image Window

The Image Window displays images and graphics generated by entering commands in the Text and Command Window. Images and graphics are displayed on frames. Graphics overlay the image. For example, in the McIDAS-X Image Window below, the map is displayed over the satellite image.



The title bar displays the McIDAS-X version number, the user's logon name, and the workstation name.

To make the Image Window active and display it on top of any overlapping windows, either click in the window or press the numeric keypad's plus (+) key while the Text and Command Window is active.

The information below applies to McIDAS-X sessions without independent graphics (the default mode). See the *McIDAS-X Independent Graphics* section for specific information about -X sessions with independent graphics.

Frames are numbered beginning with 1, and display both images and graphics, meaning that you cannot loop images or graphics independently. The current frame number is displayed at the bottom left of the Image Window. In the example above, 0001 means that frame 1 is displayed.

The following commands are used to control the Image Window display. See *Displaying Loops* for commands that control frame looping.

- SF displays a frame.
- K toggles the images on and off. When images are turned off, graphics are easier to see.
- W toggles the graphics on and off. When graphics are turned off, the image is easier to see.
- EG erases the graphics from the frame. EG I erases the image only.

More than 30 commands display or draw graphics on a frame. The new graphics are displayed on top of the previously displayed image and graphics.

Only the commands IMGDISP and IMGMAG display images on a frame. Displaying an image overwrites any previously displayed image. However, previously displayed graphics remain visible and unmodified.

McIDAS-X Independent Graphics

The information in the previous section applies to McIDAS-X sessions without independent graphics. This is the default mode at startup unless you include the command **-c 'TERM IG ON'** in your **.mcidasrc** file.

With independent graphics, McIDAS uses separate image and graphics frames, so you can display or loop them independently. This mode is useful if you create loops of colocated images because you don't have to draw the same map on every frame. Instead, you can draw the map on a single graphics frame and display that single graphics frame while looping through a series of image frames.

The following commands are valid only on sessions with independent graphics.

- SG displays a graphics frame.
- LS G and LB G define the graphics frame sequence for the loop.
- Y toggles the image frames to and from the loop control system.
- J toggles the graphics frames to and from the loop control system.

The following commands have new definitions on sessions with independent graphics.

- SF is redefined to display the specified image frame; the graphics frame is not changed.
- LS and LB are redefined to set the image frame sequence for the loop; the graphics frame sequence is not changed.

You can specify **-c 'TERM IG OFF'** or **-c 'TERM IG ON'** in your **.mcidasrc** file to have McIDAS-X automatically start in the desired mode. If a TERM IG command is not set in your **.mcidasrc** and you want to change to the independent graphics mode, you should run the appropriate TERM command when you first start McIDAS-X (before displaying images or graphics). Changing modes later may not work correctly. Problems are more likely to arise if you switch from ON to OFF than from OFF to ON. If problems do arise, it's best to exit and restart McIDAS-X.

Resizing, Zooming, and Roaming Frames

You may want to resize the Image Window because you've zoomed the current frame with the Z command, or because you've displayed a different size frame. Use the R command to resize the Image Window so that it shows the contents of the entire frame. If you specified the **-autoResize** flag at startup or in your **.mcidasrc** file, the Image Window is automatically resized to show the entire frame.

If you do not resize the McIDAS-X Image Window to view the entire frame, roam the frame to view the desired section. With the cursor in the Image Window, hold down the middle mouse button while moving the mouse. Release the button when the display is positioned where you want it. Press the right mouse button to reposition the frame so its upper-left corner appears in the upper-left corner of the Image Window.

Image and Graphics Colors

The number of colors used for displaying images and graphics is specified by the user. The colors used for displaying images are called image display levels. The colors used for displaying graphics are called graphics color levels.

In general, you don't need to know the specific image display level numbers. You do need to know the graphics color level numbers, because many commands require you to specify the output color by level number. For example, in the MAP command you specify a graphics color level number to pick the color of the map.

Each graphics color level is assigned a default color. For example, level 1 is assigned magenta. Graphics color level 0 has a special definition as the graphics background. You cannot draw graphics in level 0, but you can change it from the default color of black to a color of your choice for the background of the graphics you draw. Use the GU command to list and change the graphics color levels.

The number of colors used for displaying images and graphics depends on the number you request at startup and the number of colors available on your workstation. The maximum number of colors is 253 or 254, depending on your hardware.

The total of number of image display levels and graphics color levels cannot exceed the number of available colors (253 or 254). If you specify a total that exceeds the number of available colors, the workstation provides as many colors as possible. The more image display levels you request, the fewer colors there are available for graphics color levels and vice versa.

Use the **-imageColors** and **-graphicsColors** flags at startup or in your **.mcidasrc** file to specify the number of image display levels and graphics color levels in your McIDAS-X session. The default number of image display levels is 128. The default number of graphics color levels is 16 and the maximum is 128.

Because there are fewer image display levels (128 default, 254 maximum), than grayshades defined in most images (256), some image display levels are assigned multiple grayshades. For example, grayshades 0-16 may all be displayed as image display level 17. The SCALE ON and OFF options in the TERM command provide two methods of assigning grayshades to image display levels. See the TERM command in Chapter 3 for more information.

Frame Panels

The PANEL command divides a frame into sections called panels. This allows you to have many separate image and graphics displays in a single frame. For example, you could use the IMGDISP command to display different bands of image data in each of the panels in a frame. Or you could create a four-panel forecast chart using the GRDDISP command to contour different parameters, levels, or valid times from forecast grids.

At startup, all frames are single-panel, meaning that image and graphic commands write output to the entire frame. Single-panel frames have a panel number of 1. In multipanel frames, all panels are the same size and are numbered from left to right and from top to bottom. For example, the command PANEL 2 2 configures the current frame to have four equal-size panels, with panel 1 in the upper-left quadrant, panel 2 in the upper-right quadrant, panel 3 in the lower-left quadrant, and panel 4 in the lower-right quadrant.

When image and graphic commands write output to a multipanel frame, the output appears in only one of the frame's panels. By default, the output appears in the panel where the cursor is located. Use the global keyword PAN to override the default and display the output in a different panel. For example, MAP USA PAN=3 displays the map on panel 3. Use the PAN keyword to specify input for commands as well. For example, IMGPROBE PAN=1 lists image data in panel 1.

The commands listed below deal with multipanel frames in ways that you may not expect.

- CM, EB, EU and GU change the image or graphics colors in all panels because the frame uses a single color table.
- FRMSAVE, SVF and RVF save or restore entire frames, even if you specify the global keyword PAN.

The commands listed below have features added specifically for dealing with multipanel frames.

- EG lets you erase a range of panels by specifying beginning and ending panel numbers with the global keyword PAN.
- FRMLIST has a FORM=PANEL keyword that lists the frame directory information and size of each panel. The other FRMLIST options list only information for a single panel.
- ZA has a FULLFRAME=YES keyword that lets you write text to the entire frame instead of being restricted to one panel. This keyword is useful for labeling multipanel displays.

Adding Frames

By default, a McIDAS-X session starts with ten 480 by 640 frames. To start a McIDAS-X session with additional frames, use the **-f** start-up flag. If you plan to add more frames during your McIDAS-X session, also specify a memory pool for the frames at startup using the **-e** flag. See the section *Configuring McIDAS-X Sessions with .mcidasrc* for more information. Use the F command to list the current frame configuration for your session. To add more frames during the session, use the MAKFRM command. New frames are appended to the end of the current frame list.

Caution!
If your workstation does not have the available memory to create the number of frames requested in your MAKFRM entry, McIDAS-X tries to create frames in storage mode. If that fails, MAKFRM informs you that there is insufficient memory available.

McIDAS Cursor

When you move the pointer into the Image Window, it becomes the McIDAS cursor. The McIDAS cursor is used by many commands to list, locate or draw data on the frame. The cursor type, size, and color are defined with the CUR command.

Displaying Loops

McIDAS can display an automatically repeating sequence of frames much like a movie loop.

Defining Loops

Define the sequence of frames using either the LS or LB command. Command DR determines the amount of time that each frame is displayed.

Once a loop is defined, use the single-letter commands L, A and B to control the state of the display.

- L starts and stops the loop. If the loop is actively stepping through frames, pressing **Alt L** stops on the first frame of the loop. Similarly, if looping is inactive, pressing **Alt L** starts the animation.
- A manually advances one frame while the loop is stopped.
- B command backs up one frame while the loop is stopped.

In sessions with independent graphics, the Y and J commands toggle image and graphics frames to and from the loop control system. If Y is toggled on, image frames step through their LB or LS sequence when looping is started with the L command. If Y is toggled off, the displayed image frame does not change when looping is started. The J command controls graphics frame looping in the same way. At startup, Y is on and J is off, meaning that only image frames are looped with the L command.

Saving and Recalling Loops

After you define a loop or display using the commands described above, use the DEFLOOP command or the F Key Menu System pseudocommand *Snn to save it. You can define up to 99 loops. For more information, see Chapter 5, *F Key Menu System*, or DEFLOOP in *Chapter 3, Commands*.

Display saved loops in the Image Window using a defined function key, or using the MOVIE or DEFLOOP REST command. The MOVIE command automatically steps through predefined loops at user-defined rates.

Using Briefing Windows

In addition to displaying loops in the Image Window, you can also create briefing windows to view saved loops. Each briefing window can display a frame or loop different than those currently shown in the Image Window or other briefing windows.

Use the command BRIEF to create a briefing window. You can create up to ten additional briefing windows to display the loops. If you resize a briefing window, its displayed frames are automatically rescaled to fill the window.

You can only display loops 1 through 9 in a briefing window. Use the MOVIE command with the WIN keyword, or make the briefing window active and press a key between 1 and 9 that corresponds to the loop number. Use the A, B and L keys in briefing windows to control frame stepping.

Use the SYNC command to define and display frame sequences in multiple briefing windows. The loops are synchronized, making it easier to compare loops of visible and infrared satellite imagery, for example.

Automatically Updating Loops

If you have access to a server ingesting images in real time, you can use the SKE, SKL and SKU scheduler commands with the IMGDISP command to automatically display the latest image in a loop of frames. You can also adjust the loop's dwell rates or frame sequence to emphasize the new image. Two methods are available to do this.

You can run the DR AUTO command after each new image is displayed. This command rotates the dwell rates, assigning the longest dwell to the most recent image in the loop.

Or, you can define the loop's frame sequence using the LS command with the UPDATE=Y keyword, which automatically rotates the frame sequence each time a new image is displayed with IMGDISP. The new image moves to position 1 in the loop. Dwell rates are not changed, so the new image has the dwell initially set for position 1. The advantages of this method are that you only run the LS command once, and that the new image is displayed if you stop the loop with the L command.

Optimizing the Loop Display

McIDAS-X has configuration options that determine the quality of frame looping. The options are the **-optimizeForSpeed** or the **-optimizeForSpace** start-up flags. See the section titled *Configuring McIDAS-X Sessions with .mcidasrc* for more information.

Although McIDAS-X allows frames of different sizes in any session, you should make all frames in the loop bounds the same size when using the

L command for looping. Different size frames display poorly and are handled roughly by the window manager.

Using Strings and Function Keys

Strings are used to store long, difficult-to-remember, or commonly used commands or values. For example, you might use a string named LL to store a latitude/longitude coordinate (like -35:30:15 98:25:40) that you frequently use when copying or displaying images. Use the commands TE, TD, TL, and TU to work with strings. These commands are described below; for more information see *Chapter 3, Commands*.

- TE creates strings and enters them in the current string table.
- TD deletes strings from the current string table.
- TL lists strings in the current string table.
- TU is the string table utility. Use it to save and restore string tables.

The following conventions apply to creating strings:

- String names can be up to 12 alphanumeric characters. Letters must be in uppercase.
- The maximum number of characters that can be stored in a string is 160 characters, including spaces.
- String names can be single letters, but cannot be H, Y, or X. The strings named H and Y contain the current time and date and cannot be redefined or deleted. You cannot use the string name X because McIDAS interprets it as a place holder for default values in a command.

Use the TE or ASK1 command to change the value of an existing string. For example, if you want to change the value of the string STATIONS from DCA MIA MSP LAX to KDCA KMIA KMSP KLAX, you can either recreate the string and its value with TE, or run command ASK1 STATIONS and add the preceding K before each station.

Using Strings

To run a string containing a command, type a pound symbol (#) followed by the string name and press Enter. For example, if the string LAX contains the value PC L KLAX, type #LAX on the command line and press Enter to run the PC command.

To use a string's value in a command, type *#string* in the appropriate position on the command line. For example, if the string LL contains the value -35:30:15 98:25:40, the entry IMGDISP GOES/IR LATLON=#LL is converted to IMGDISP GOES/IR LATLON=-35:30:15 98:25:40 when you press Enter.

Strings can be nested, meaning that one string is used in another. For example, if you create a string called IMAGE using the command TE IMAGE "IMGDISP GOES/IR LATLON=#LL, the string #LL is expanded so that the string IMAGE contains the value IMGDISP GOES/IR LATLON=-35:30:15 98:25:40. Nested strings must be defined before they can be used in another string.

If the nested string contains a value that will change, use two pound signs in the command string instead of one. This forces the nested string's value to be expanded when the command string is run rather than when it is created with TE. For example, if the value of LL will change, create the string IMAGE with the command TE IMAGE "IMGDISP GOES/IR LATLON=##LL. The command string IMAGE now contains IMGDISP GOES/IR LATLON=#LL; in other words, it contains a string for latitude/longitude rather than an actual value. The value for the nested string LL will be substituted when you enter the command #IMAGE.

If a string with a single-letter name contains a command, run the command by typing *#letter* and pressing Enter, or by pressing the **Ctrl** key and the letter at the same time. For example, if the string T contains the value IMGPROBE LIST BOX TEMP, run the IMGPROBE command by entering #T, or by pressing **Ctrl T**.

Using Function Keys

If a command string is named KEYF*n*, where *n* is a number between 1 and 12, run the string by pressing the corresponding function key (F1 to F12). Strings named KEYAF*n*, KEYCF*n* or KEYSF*n*, are run by pressing the **Alt**, **Ctrl** or **Shift** key, and a function key at the same time. When you run the function key strings, McIDAS ignores the command line. This means that you can use a function key command even while entering a command on the command line. The function key combinations work correctly only if the workstation is configured according to the instructions in *Configuring a New User Account*.

The 48 function key combinations can also be used in a menuing system. See Chapter 5, *F Key Menu System*, for more information.

String Tables

When you create a string with the TE command, it is placed in the current string table. You lose the strings in the current table if you run TD or TU REST or you EXIT and restart McIDAS. Because the string table file (**STRTABLE**) is kept in the session's **.mctmp** subdirectory, it's deleted when you end your McIDAS session. To prevent this, move or copy **STRTABLE** to your **$HOME/mcidas/data** directory. Because the strings will be shared by all the sessions, this will increase the likelihood of conflicts if you run multiple McIDAS-X sessions from the same account.

Using the McIDAS-X Graphical User Interface

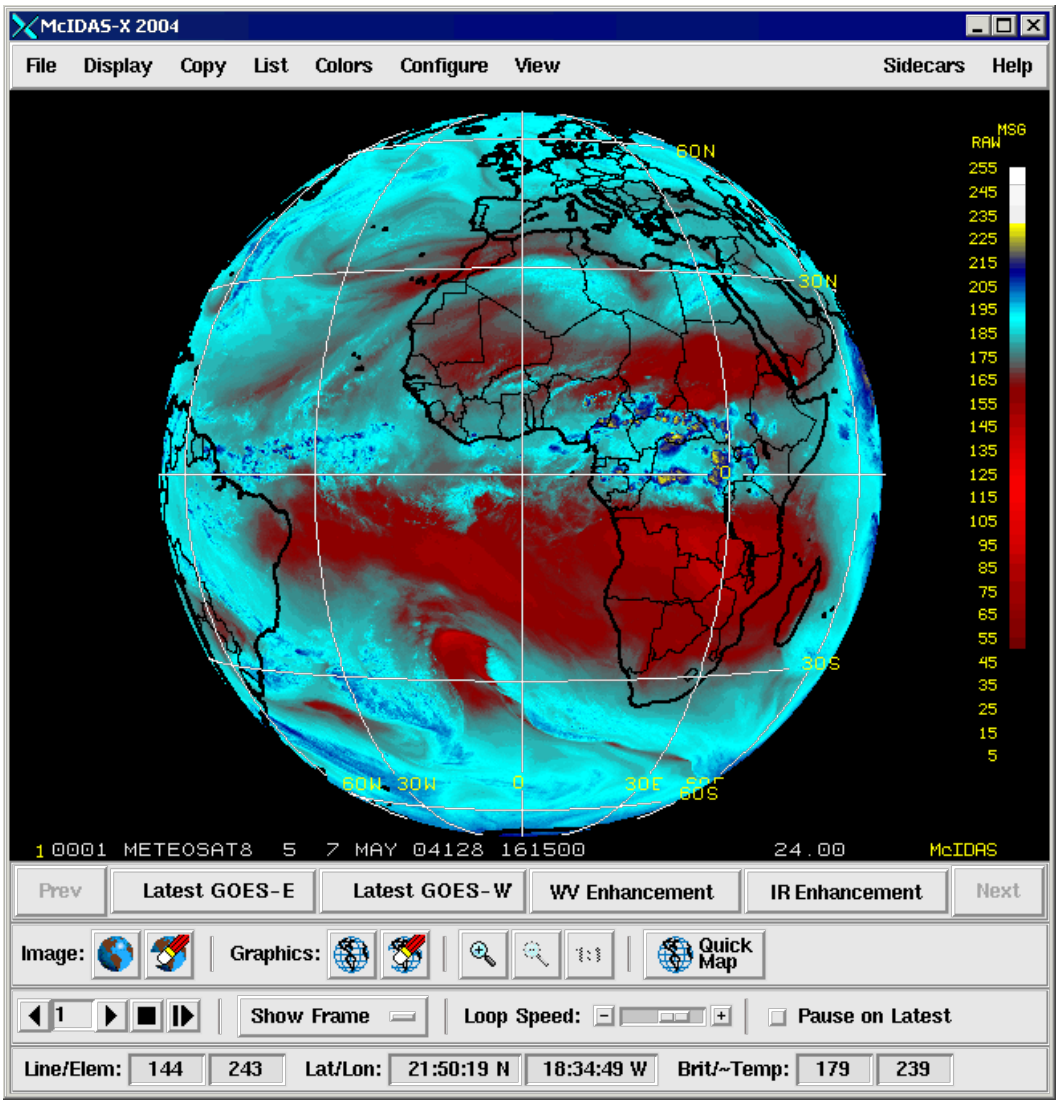
The McIDAS-X Graphical User Interface (GUI) lets you display images and data without using the McIDAS-X Text and Command Window. However, you can still run commands from the McIDAS-X Text and Command Window while the GUI is displayed.

This section provides a brief introduction to the McIDAS-X GUI. The GUI's Help menu provides more detailed information.

Starting and Exiting

You can start the GUI by running the command GUI from the McIDAS-X Text and Command Window. To start the GUI automatically when you start a McIDAS-X session, add the line **-c 'GUI'** to your **.mcidasrc** file.

When you start the GUI, a menu bar and frame controls are added to the Image Window as shown below. For information about using the GUI frame controls, select **Getting Started** from the Help menu.



To exit the GUI, choose **Exit GUI** from the File menu. The GUI closes and the standard McIDAS-X Image Window reappears. To exit both the GUI and the McIDAS-X session, choose **Exit McIDAS** from the File menu.

Displaying Real-time Data

You can display real-time images, grids, surface data, and upper air data using the McIDAS-X GUI. To display real-time grids, surface data and upper air data, the ADDE Client Routing Table must contain the groups RTGRIDS and RTPTSRC. To verify that the groups are present, select **ADDE Servers** in the File menu, then check that RTGRIDS and RTPTSRC are in the list of servers. If they are not present and you want to add them, select **Add ADDE Server** and then insert RTGRIDS in the *ADDE Name* field and insert the IP address of the appropriate server in the *IP Address* field. Repeat for the group RTPTSRC.

Changing the GUI Display

By default the McIDAS-X GUI uses the default attributes listed in the table below. You can change the default arguments and other Tcl/Tk settings by creating a file with any text editor called **.mcguirc** file. You must name the file **.mcguirc** or the McIDAS-X GUI will not display your changes. After you create the file, store the file in a directory that is named in the **MCPATH** environment variable. See *Appendix G, MCPATH* for more information. The *Tcl/Tk User's Guide* provides more information about the attributes you can change.

Attribute	**Default Argument**
window background color	*background gray 90
window foreground color	*foreground navy
active window background color	*activeBackground navy
active window foreground color	*activeForeground yellow
GUI font	*font "-adobe-courier-bold-r-normal–*-120*"
error message font	*Messagefont "-adobe-courier-bold-r-normal–*-140*"

The example below shows a **.mcguirc** file for the default attributes listed above.

```
*background gray90 *foreground navy *activeBackground navy *activeForeground yellow
*font "-adobe-courier-bold-r-normal--*-120*" *messageFont "-adobe-courier-bold-r-
normal--*-140*"
```

Using the ADDE

This section provides an overview of the ADDE (Abstract Data Distribution Environment) system. The ADDE allows your workstation to act as a client, efficiently accessing data from multiple McIDAS servers. This chapter introduces you to ADDE concepts and use. The following topics are included:

- Terminology
- Clients and Servers
- Data Naming Scheme
- Client and Server Look-up Tables
- Naming Data On Your Local Server
- Using Compressed Data Transfers
- Setting the ADDE Client Timeout
- Setting the Image Tracking Mode

Terminology

The terms below are used throughout this guide. Remember, because a workstation can store and display data, it can act as a server and a client.

Term	Definition
Alias	a short, user-defined name representing an ADDE dataset name; for example, the alias GV1 might represent the dataset name SSEC-RT/GOES-1KVIS
Client	workstation receiving and displaying data
Dataset	collection of one or more files with a common format; for example, one dataset may consist of areas 101 to 104, another dataset may consist of MD files 1 to 10
Dataset Name	name used by ADDE to reference a dataset; the dataset name consists of a group name and a descriptor name separated by a slash (/); for example, SSEC-RT/GOES-1KVIS
Descriptor	name used to reference a dataset; for example, a dataset of areas 105 to 108 containing GOES visible data at 4 km resolution might have the descriptor GOES-VIS-4K
File Number	number used to reference a McIDAS format file (the standard McIDAS numbering scheme)
Format	file format; McIDAS formats include area, grid and MD; non-McIDAS formats include HDF and netCDF
Group	name used to reference a collection of descriptors; used by the client to determine which server should receive data requests
Position	absolute or time-relative position of a file in a dataset; position numbers greater than zero represent an absolute position in the dataset; position numbers less than or equal to zero represent a relative position, 0 is the most recent and -1 is the next most recent; for example, if a dataset has four images with times 13 UTC, 14 UTC, 15 UTC and 12 UTC, they have the positions -2, -1, 0 and -3
Server	machine storing and supplying data
Type	data Type: image, grid, point, navigation, or text

Clients and Servers

ADDE distributes data using networked servers and clients. Servers store data and distribute it to the client. Clients request and receive data and run applications that use the data. Clients and servers communicate using the TCP/IP communications protocol.

Each account running McIDAS-X acts as both a client and local server. When a McIDAS-X client requests data from the local server, it searches for the data in the directories specified in the user's **MCPATH** environment variable or another directory specified in a REDIRECT entry.

The client can also request data from a remote server. A remote server can be either of the following:

- a different account on the same McIDAS-X workstation configured as a remote server
- an account on another McIDAS-X workstation configured as a remote server

The difference between a local server and a remote server on a McIDAS-X workstation is that the data in a local server is available only to the McIDAS-X sessions started under that account name. Data in a remote server account is accessible to all McIDAS-X ADDE clients. Remote servers are discussed more thoroughly in *Appendix I, McIDAS-X Site Administration and Management*.

Data Naming Scheme

In the non-ADDE core commands (for example, IGU and MDU), all image, grid and point files are referenced by file numbers. If you don't know the file numbers, finding data can be difficult. The ADDE commands use dataset names (in a group/descriptor format) that map to datasets. If the descriptors and group names follow a logical convention, it's easy to locate the data.

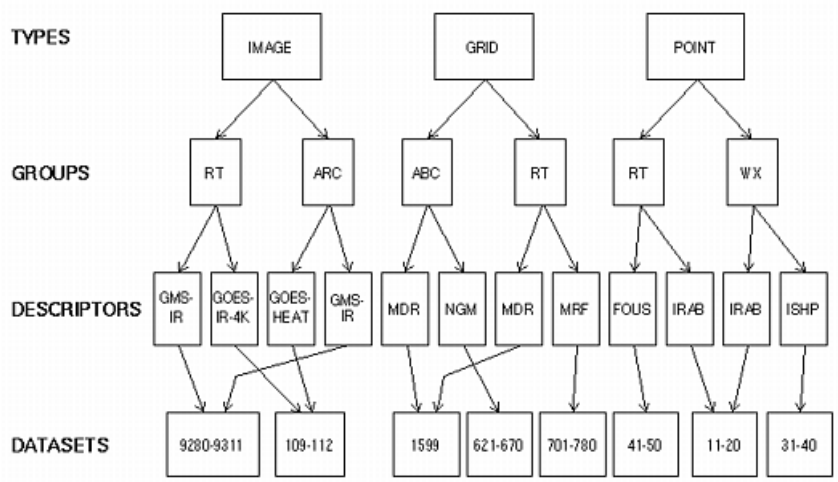
The naming scheme for datasets consists of three parts:

- type
- group
- descriptor

In the diagram below, *type* is the top tier in ADDE's hierarchical naming scheme. Image, grid and point indicate the type of data in the dataset files. McIDAS area files are image data; McIDAS grid files are grid data; McIDAS MD files are point data.

Group is the next tier in the naming scheme. A group name can be used only once under each type. The group tier in Figure 1 has three groups named RT. Notice that RT is used only once under each type.

Descriptors are the bottom tier in the naming scheme. Descriptors are not data files; they are names that point to datasets. Identical descriptors under different groups can point to the same or different datasets.



Since each ADDE command that accesses data works only with a single type, you don't have to specify the type in the command entry. For example, the command IMGDISP does not have a parameter or keyword for type because it limits its search to datasets of type IMAGE. The following command displays the most recent image in the dataset named SSEC-RT/GOES-IR-4K on frame 1, centered on St. Louis, Missouri.

IMGDISP SSEC-RT/GOES-IR-4K 1 STATION=STL

Client and Server Look-up Tables

Each ADDE client command starts a transaction by requesting data from a server. Both the client and server must recognize the command's specified dataset name in their look-up tables. On the client, this table is called the *routing table* because it determines which server to route the data request to. On the server, this table is called the *mapping table* because it maps the group and descriptor to a specific image, grid, point, navigation or text dataset.

Client Routing Table

The client routing table contains two lists: a list of group names with their associated server IP addresses, and a list of aliases with the dataset names they represent. When you enter an ADDE command, the client routing table is scanned for an entry with the specified group name or alias. If an entry is found, the data request is routed to the server specified in the entry. If the group name or alias is not found in the client routing table, the request is routed to the local server.

Use the DATALOC command to manage the list of group names. Use the AKA command to manage the list of aliases. Aliases allow you to use a short, easily remembered name to represent a dataset name in other ADDE commands. You can use this alias regardless of which server the dataset is stored on. For example, you could create the alias GV to represent the dataset name SSEC-RT/GOES-VIS-GLO.

Each user can access multiple client routing tables. This optional feature is useful when a site table has the default routing information and alias names for all users and each user has a local table with individualized routing information and alias names. The default file for your local table is **MCTABLE.TXT**. The site table, if it exists, is maintained by your site administrator or operations staff and is usually write-protected so you can't modify it. You may have to NFS mount its directory to access the file. Use the DATALOC and AKA commands to list information from all the client routing tables accessible to you.

If you decide to use multiple client routing tables, you must set the Unix environment variables **MCTABLE_READ** and **MCTABLE_WRITE**. The **MCTABLE_READ** value specifies, in order of precedence, the client routing tables accessed when searching for a group or alias match. The tables must be separated by semicolons (;) and specified as either file names or fully qualified path names (directory and file name). The **MCTABLE_WRITE** value specifies the single client routing table you can modify by creating or deleting group or alias entries with the ADD and DEL parameters of the DATALOC and AKA commands.

You cannot specify more than one table in **MCTABLE_WRITE**. SSEC recommends that the table you specify be the same as the first table in **MCTABLE_READ**. See the appropriate example below for your system.

McIDAS-X Example

If you want to use **$HOME/mcidas/data/MCTABLE.TXT** for your local table and **/data/ADDESITE.TXT** for your site table, enter the following Unix commands (substitute the equivalent if using C shell) to set the environment variables.

```
export MCTABLE_WRITE; MCTABLE_WRITE="$HOME/mcidas/data/MCTABLE.TXT" export
MCTABLE_READ; MCTABLE_READ="$HOME/mcidas/data/MCTABLE.TXT;/data/ADDESITE.TXT"
```

If you want these values for **MCTABLE_WRITE** and **MCTABLE_READ** set each time you login to your account, add the commands above to your **$HOME/.mcenv** file and add the following line to your **$HOME/.profile** (ksh) or **$HOME/.cshrc** (csh) file. Be sure to leave a space between the period (.) and **$HOME**.

```
. $HOME/.mcenv
```

Server Mapping Table

The server mapping table maps dataset names to the files that make up the datasets. When the server receives a data request from a client, it reads the mapping table to locate the correct dataset. The server sends the requested data back to the client so it can list or display the data.

Use the DSSERVE command to list or modify the server mapping table. Clients can only modify their local server's mapping table; they cannot modify a remote server's mapping table. The remote server's administrator must inform the client's administrator or user which ADDE group names the client may access. The client's administrator then modifies the client routing table with command DATALOC so applications can request data from groups located on the remote server.

Naming Data On Your Local Server

Use the DSSERVE command to create dataset names on your local server. DSSERVE assigns the group and descriptor to a dataset consisting of one or more files of the specified format. You can also add a description of up to 80 characters. For example, the following command assigns the dataset name PLANET/IMAGES to areas 7001 through 7199 on the local server:

DSSERVE ADD PLANET/IMAGES AREA 7001 7199 "Planetary images from Voyager

You can use the same group or descriptor with each data type. For example, if you have image, grid and point data for a case study of a major flood event, you may want to use the group name FLOOD for all the datasets. Or if you have large numbers of images that you use as part of a demonstration, you may want to use the group name DEMO with logical descriptor names like TSTORM, FOREST-FIRES, ECLIPSE and HURRICANE.

When naming data on your local server, you must use group names that are not mapped to other servers in your client routing table. Use the command entry DATALOC LIST to list the group names in your client routing table. Since ADDE commands that request data from a group name not found in the client routing table are automatically routed to the local server, you do not have to add your local server's group names to the table.

Using Compressed Data Transfers

There are three ADDE data transfer methods available for sending data from remote servers to your client workstation. The methods are gzip-compressed, compressed, and uncompressed.

The gzip-compressed and compressed methods are most beneficial to users who transfer large data files or those with slow data connections. However, they only reduce the amount of data during the transfer; they do not change the resultant file size or accounting (the byte counts remain the same). For example, the destination file size from an IMGCOPY command is the same no matter which of the three methods you use. If most of the datasets you access are on a local network with fast transfer rates, then using the uncompressed method is probably best because it minimizes overhead.

The data transfer method you use on your client workstation is determined by the setting of its **MCCOMPRESS** environment variable. Because it's an environment variable, the method you select is the default for the entire McIDAS-X session. The only way to change the default method is to exit your McIDAS-X session, change the setting of **MCCOMPRESS**, then restart McIDAS-X. To override the default setting on a command-by-command basis, use the **MCCOMPRESS** global keyword with any ADDE command. See *Appendix A, Global Keywords* for more information.

Any client can use the compressed or uncompressed methods because all remote servers have the ability to send data using those methods. However you can use the gzip-compressed method only if *all the remote servers you access are running version 2003 or later*. If any of the remote servers are running a version prior to 2003 (or if you're unsure of the version), do not select gzip-compressed because if you request gzip-compressed transfers from a remote server running a version prior to 2003, your McIDAS-X command and the **gunzip** command waiting for data will fail.

If **MCCOMPRESS** is not set on your client workstation, or if it's set to **NONE**, then the uncompressed data transfer method is used. If you want to use the compressed data transfer method, enter the following Unix command (substitute the equivalent if using C shell) on your client workstation.

Type: **export MCCOMPRESS; MCCOMPRESS=COMPRESS**

If you want to use the gzip-compressed data transfer method and you've verified that your client workstation and all the remote servers it accesses are running McIDAS-X 2003 or later, enter the following Unix command (substitute the equivalent if using C shell) on your client workstation.

Type: **export MCCOMPRESS; MCCOMPRESS=GZIP**

If you want to set the environment variable each time you log on to your account, add the preferred command of the two above to your **$HOME/.mcenv** file and add the following line to your **$HOME/.profile** (ksh) or **$HOME/.cshrc** (csh) file. Be sure to leave a space between the period (**.**) and **$HOME**.

```
. $HOME/.mcenv
```

Setting the ADDE Client Timeout

By default, ADDE commands timeout if they do not receive any data from the server for a period of 120 seconds. You can change the timeout period with the environment variable **ADDETIMEOUT**. To do so, enter the following Unix command (substitute the equivalent if using C shell), where *numseconds* is the number of seconds.

Type: **export ADDETIMEOUT; ADDETIMEOUT=** *numseconds*

If you want to set the environment variable each time you log on to your account, add the command above to your **$HOME/.mcenv** file and add the following line to your **$HOME/.profile** (ksh) or **$HOME/.cshrc** (csh) file. Be sure to leave a space between the period (**.**) and **$HOME**.

```
. $HOME/.mcenv
```

Setting the Image Tracking Mode

Some ADDE image servers (e.g., Area, GVAR and MTSAT) are able to track the data as a new image is being received. Tracking means that clients that request data while it's being received by the server will wait until all of the requested data is available before completing the request. For example, if you run an IMGDISP command while the image is being received on a remote server that tracks, the IMGDISP command receives and displays data as it's being ingested by the server, and completes when its data request has been fulfilled.

The default tracking mode on your client workstation is determined by its **MCTRACK** environment variable setting (YES or NO; if it's not set then the server determines whether or not to track). Because **MCTRACK** is an environment variable, the value you select is the default for the entire McIDAS-X session. To override the default setting on a command-by-command basis, use the TRACKING=YES/NO keyword in the IMGCOPY, IMGDISP, IMGFILT, IMGMAG, IMGOPER, IMGPLOT, IMGPROBE, IMGREMAP, IMGRGB or IMGTRACK command.

Note that if a remote server is unable to track data or the server's administrator has disabled tracking, then the setting of your **MCTRACK** environment variable and the TRACKING= keyword are ignored and your IMG* command will receive only the data that was available when the command was issued.

If you want to track image data if the servers you access allow it, enter the following Unix command (substitute the equivalent if using C shell) on your client workstation.

Type: **export MCTRACK; MCTRACK=YES**

If you prefer to not track image data from any of the servers you access (so your IMG* command will receive only the data that was available when the command was issued), enter the following Unix command (substitute the equivalent if using C shell) on your client workstation.

Type: **export MCTRACK; MCTRACK=NO**

If you want to set the environment variable each time you log on to your account, add the preferred command of the two above to your **$HOME/.mcenv** file and add the following line to your **$HOME/.profile** (ksh) or **$HOME/.cshrc** (csh) file. Be sure to leave a space between the period (**.**) and **$HOME**.

```
. $HOME/.mcenv
```

Getting Started with ADDE

The exercise below introduces you to ADDE concepts such as dataset names, aliases, clients and servers. Use this exercise to practice adding entries to your client routing table and local server's mapping table, and displaying and copying image data from a remote server on an SSEC McIDAS-X workstation. The examples below use image datasets, but the concepts apply to all dataset types (grid, image, navigation, point and text). When you type a command, press Enter to run it. For more information about the commands, see *Chapter 3, Commands* and the online helps.

1. Log on to the workstation as user DEMO with project number 1234.

    Type: **LOGON DEMO 1234**

Logging on is required only if the ADDE remote servers you access require a user name and a project number for accounting. Administrators can allow access only to specific logon names and project numbers.

2. List your client routing table's groups and associated server IP addresses.

    Type: **DATALOC**

If this is the first time you are using the ADDE, the table is empty and the following listing is displayed.

```
Group Name Server IP Address
```

3. Modify your client routing table so that commands requesting data from a dataset in the group BLIZZARD are routed to the SSEC remote server with the IP address 128.104.109.190.

    Type: **DATALOC ADD BLIZZARD 128.104.109.190**

The following listing is displayed. The last line informs you that commands requesting data from a group with the Server IP Address <LOCAL-DATA> are routed to the local server

```
Group Name Server IP Address BLIZZARD 128.104.109.190 <LOCAL-DATA> indicates that
data will be accessed from the local data directory.
```

4. List all datasets of type IMAGE in the group BLIZZARD.

    Type: **DSINFO IMAGE BLIZZARD**

The following listing is displayed. The NumPos value is the number of positions in the dataset. This data is located on the remote server at 128.104.109.190.

```
Dataset Names of Type: IMAGE in Group: BLIZZARD Name NumPos Content G7-IR-4K 3
Storm of the Century G7 IR G7-VIS-1K 1 Storm of the Century G7 1K G7-VIS-4K 3 Storm
of the Century G7 VIS G7-VIS-FD 1 Storm of the Century G7 16K IMAGES 14 Storm of
the Century All Images M3-IR 3 Storm of the Century M3 IR M3-VIS 3 Storm of the
Century M3 VIS
```

5. List all the images in the dataset BLIZZARD/G7-VIS-4K.

    Type: **IMGLIST BLIZZARD/G7-VIS-4K.ALL**

The following listing is displayed. The Pos value is the absolute position of the image in the dataset

```
Image file directory listing for:BLIZZARD/G7-VIS-4K Pos Satellite/ Date Time Center
Band(s) sensor Lat Lon _______ 1 GOES-7 13 MAR 93072 17:01:00 26 87 1 2 GOES-7 13
MAR 93072 17:31:00 26 87 1 3 GOES-7 13 MAR 93072 18:01:00 26 87 1
```

6. List the two most recent images (positions 0 and -1) in the dataset BLIZZARD/G7-VIS-4K.

    Type: **IMGLIST BLIZZARD/G7-VIS-4K.-1**

The following listing is displayed. The Pos value is the absolute position of the image in the dataset. The image in position 3 is listed first because it is the most recent image, relative position 0. The image in position 2 is the second most recent image, relative position -1.

```
Image file directory listing for:BLIZZARD/G7-VIS-4K Pos Satellite/ Date Time Center
Band(s) sensor Lat Lon _______ 3 GOES-7 13 MAR 93072 18:01:00 26 87 1 2 GOES-7 13
MAR 93072 17:31:00 26 87 1
```

7. Create the aliases GV4 and GI4 for the dataset names BLIZZARD/G7-VIS-4K and BLIZZARD/G7-IR-4K.

Type: **AKA ADD GV4 BLIZZARD/G7-VIS-4K**

Type: **AKA ADD GI4 BLIZZARD/G7-IR-4K**

8. List the aliases in your client routing table.

Type: **AKA**

The following listing is displayed.

```
Alias Name Group/Descriptor GI4 BLIZZARD/G7-IR-4K GV4 BLIZZARD/G7-VIS-4K
```

9. Display the most recent image in the dataset with the alias name GV4. Display the image on frame 1 centered on Athens, Georgia, and draw a map on it. The AKA listing in step 7 shows that the alias GV4 represents dataset BLIZZARD/G7-VIS-4K.

Type: **IMGDISP GV4 1 STATION=KAHN SF=YES REFRESH='MAP VH 2'**

10. Assign the dataset name MYDATA/IMAGES to areas 1 through 9999 on the local server.

Type: **DSSERVE ADD MYDATA/IMAGES AREA 1 9999 "All areas on local server**

11. Use the IMGLIST command to find five empty areas on the local server.

Type: **IMGLIST MYDATA/IMAGES.991 995**

If data exists in this block of areas, move or delete the areas or use the IMGLIST command to find a different block of empty areas.

12. Assign the dataset name MYDATA/TEST-IMAGES to the empty block of areas.

Type: **DSSERVE ADD MYDATA/TEST-IMAGES AREA 991 995 "Scratch areas for testing**

13. List the datasets on the local server.

Type: **DSSERVE**

A listing similar to the following is displayed.

```
Group/Descriptor Type Format & Range RT Comment MYDATA/IMAGES IMAGE AREA 1-9999 All
areas on local server MYDATA/TEST-IMAGES IMAGE AREA 991-995 Scratch areas for
testing
```

14. Modify your client routing table so that commands which request data from the group MYDATA are routed to the local server.

Type: **DATALOC ADD MYDATA LOCAL-DATA**

This step is optional because ADDE commands that request data from a group not found in your client routing table are automatically routed to the local server. If you add your local server's groups to the table, you can use the DATALOC LIST command to list all the groups you can access, on local and remote servers.

15. List the groups in your client routing table.

Type: **DATALOC**

```
Group Name Server IP Address BLIZZARD 128.104.109.190 MYDATA <LOCAL-DATA> <LOCAL-
DATA> indicates that data will be accessed from the local data directory.
```

16. Create the alias TI for the dataset name MYDATA/TEST-IMAGES.

Type: **AKA ADD TI MYDATA/TEST-IMAGES**

17. List the aliases in your client routing table.

Type: **AKA**

The following listing is displayed.

```
Alias Name Group/Descriptor GI4 BLIZZARD/G7-IR-4K GV4 BLIZZARD/G7-VIS-4K TI
MYDATA/TEST-IMAGES
```

18. List all the images in the dataset with the alias name TI. The AKA listing in step 17 shows that the alias TI represents dataset MYDATA/TEST-IMAGES.

Type: **IMGLIST TI.ALL**

The following output is displayed because the dataset has no images.

```
Image file directory listing for:TI IMGLIST: No images satisfy the selection
criteria
```

19. Copy the most recent image in the dataset with the alias name GI4 to position 5 in the dataset with the alias name TI. Place Athens, Georgia, at its center. The AKA listing in step 17 shows that the aliases GI4 and TI represent datasets BLIZZARD/G7-IR-4K and MYDATA/TEST-IMAGES.

Type: **IMGCOPY GI4 TI.5 STATION=KAHN**

20. List all the images in the dataset with the alias name TI.

Type: **IMGLIST TI.ALL**

The following listing is displayed..

```
Image file directory listing for:BLIZZARD/G7-VIS-4K Pos Satellite/ Date Time Center
Band(s) sensor Lat Lon _______ 5 GOES-7 IR 13 MAR 93072 18:01:00 34 83 8
```

21. Display the image in position 5 in the dataset with the alias name TI. Display the image on frame 2 and draw a map on it.

Type: **IMGDISP TI.5 2 SF=YES REFRESH='MAP VH 2'**

Chapter 3

Commands

The Commands chapter consists of these sections:

- Documentation Conventions
- Commands by Categories
- Commands by Alphabetical Order

McIDAS commands in this chapter are organized alphabetically. To help you determine which command may be appropriate for a particular situation, refer to the *Commands by Categories*. If you want information about a particular command, refer to the *Commands by Alphabetical Order*.

Documentation Conventions

The conventions used in documenting the commands are:

- Interpreting Bold and Italicized Terms
- Recognizing Screen Displays and Text Files
- Interpreting Command Information
- Command Example

Interpreting Bold and Italicized Terms

Actual keyboard entries appear in **BOLD** type. Type these entries exactly as they appear. Command names, some parameters, all keywords and examples appear in bold. For example:

IGG MAKE 2 SUB 1

Variable entries, such as parameters, appear in *italics*. For example:

HELP *command*

When you enter commands, use capital letters and leave a space between each term or number in the command line. Replace terms in italics with the appropriate information. In the example above, replace *command* with the name of the command for which you want to display online help. To get help on the CUR command, enter the command below.

If you want to list the online documentation for command CUR, type:

HELP CUR

Recognizing Screen Displays and Text Files

Screen displays, system prompts and the contents of text files look like this:

```
Boxed text displayed in this typestyle indicates a screen message or lines in a
text file.
```

Interpreting Command Information

Most commands have Format, Parameter, Keyword, Remark and Example sections. To better understand the information in each of these sections, see the *Command Example*.

Command Example

This line gives a brief description of the command's function.

Format

The format contains the command name along with any positional parameters and keywords that it needs to perform its function. Commands can have several formats. Enter **BOLD** terms exactly as they appear in the command format. Terms in *italics* are variables. You must enter a value or option in place of them in the command. If a format is too long to fit on one line, the second line is indented.

Parameters

Parameters are numbers or letters that provide information to a command. To the right of each parameter is an explanation of its function. Enter parameters in the exact order specified in the command format. It is not always necessary to use every parameter. Parameters are listed in the documentation in the order they appear in the command line. Enter **BOLD** parameters exactly as they appear in the text. If a parameter is in *italics*, insert the appropriate number or letter in the command format. A quotation mark (quote field) is sometimes used to delineate a text string. The text string can be used as input to search a file, or to provide descriptive text to a command or database.

Keywords

Keywords further clarify a command function; they are optional. Keywords are one or more characters followed by an equal sign (or comma) and a value. For example, COLOR=3 LAT=20 40 UNIT=A can be entered as COLOR,3 LAT,20 40 UNIT,A. Type text values exactly as they are listed with the keyword. Use as many keywords as you like in a single command format. Although keywords are listed alphabetically in the documentation, their order in the command line is not important as long as they come after the parameters. To the right of each keyword is an explanation of its function. Most keywords may be truncated to three characters, though some require more. The required characters are given in capital letter. For example, **COLor** means that you can enter COL, COLO, or COLOR; **DERive** means that you can enter DER, DERI, DERIV, or DERIVE.

Remarks

Additional information about the command is contained here.

See Also

Mentioned here are other related commands to reference.

Examples

One or more examples are provided for each command. The first example is the shortest useful version of the command. If an example takes more than one line, the second line is indented. Below each bold command line is an explanation of what the entry does.

Commands by Categories

The command categories below are ordered from the basic to the complex. Basic commands are those that every user needs to know to run McIDAS (system utilities, frame and cursor control). Next are the Graphics and frame analysis commands, which are general applications. The Weather Analysis category contains those commands that list or display weather data. This category is followed by commands that are specific to each data type (image, point, grid and text). The Navigation category contains commands that are used to display dataset navigation data. The last categories are the optional tools and administrative utilities that are infrequently used, followed by commands that define or use ADDE datasets.

Category	**Category Class**
System Utilities	Basic
Other Utilities	
Frame and Cursor Control	
Graphics	Graphics and Frame Analysis
Frame Analysis	
Image and Graphic Coloring	
Weather Analysis	Weather Analysis
Image	Data Type Specific
Point	
Grid	
Text	
Navigation	Navigation
Tools	Tools and Administrative Utilities
Administrative Utilities	
ADDE Commands	ADDE dataset manipulation commands

System Utilities

Commands in this category perform basic workstation control tasks or list information about the current session.

Command	Purpose
& (Ampersand)	Recalls a previously entered command.
^ (Caret)	Recalls previously entered commands.
! (Exclamation Point)	Sends a command to the Unix shell.
+ (Plus)	Raises the Image Window.
? (Question Mark)	Lists the status of McIDAS commands and processes.
/ (Slash)	Stops McIDAS commands.
C	Lists a frame's directory.
CASEMODE	Sets the case mode for entering McIDAS-X commands.
CLEAR	Clears output from text frames.
COLORS	Lists the colors available with the global keyword TCOL.
DMAP	Lists information about files.
ECHO	Displays a line of text on a text frame.
EDITMODE	Lists and changes the McIDAS command input editor.
EXIT	Ends a McIDAS-X session and associated windows.
F	Displays the workstation's state.
FRMLIST	Lists frame directory information.
HELP	Lists online helps.
KILL	Terminates McIDAS commands.
LOGOFF	Logs you off the McIDAS session.
LOGON	Logs you on to the McIDAS session.
OS	Runs an operating system command.
TFILE	Sends text output to a disk file.
TFIND	Searches the current text frame.

Other Utilities

Commands in this category perform miscellaneous actions that don't fit in the other categories.

Command	Purpose
CCODE	Lists country codes.
COMBINE	Combines three McIDAS frames into a single 24-bit image.
CU	Lists a quantity in up to eight different units.
DATELIST	Lists the specified date in several formats.
FIND	Locates U. S. towns by name and state.
FRMSAVE	Saves a McIDAS frame to one of several image formats.
GLOBKEY	Lists global keyword parameters and examples.
LVF	Lists virtual frame files.
RVF	Restores virtual frames saved with command SVF.
S	Lists Metar, Synoptic and RAOB stations within the cursor bounds.
STNLIST	Lists weather stations.
STNPLOT	Plots stations on a navigated frame.
SUNLIT	Lists the sunrise and sunset times for a specified location.
SVF	Saves frames to a virtual frame file.
TIMEOUT	Runs a McIDAS command, stopping it after a waiting period.

Frame and Cursor Control

Commands in this category manipulate image frames, graphic frames and the cursor.

Command	Purpose
A	Advances one frame.
B	Backs up one frame position.
BRIEF	Opens a briefing window.
CUR	Defines cursor size, type and color.
DIST	Finds the distance on a navigated frame.
DR	Defines the dwell rates for frame looping.
EG	Erases graphics and images.
ERASE	Erases graphics, images or entire frames.
J	Toggles graphics frames to and from the loop control system.
K	Toggles image frames on and off.
L	Toggles frame looping on and off.
LB	Sets image or graphics frame loop bounds.
LS	Creates random loop sequences for image and graphics frames.
MAKFRM	Adds new frames to a McIDAS-X session.
O	Switches the image frame position to the opposite frame.
PANEL	Configures frames with multiple panels.
PC	Positions the cursor at a desired point on the frame.
PLAX	Calculates parallax at cursor position.
R	Resizes the Image Window to the size of the current frame.
SF	Displays a frame.
SG	Displays a graphics frame.
SYNC	Synchronizes loops running in multiple briefing windows.
TERM	Sets the state of a McIDAS toggle command or mode.
TOWNS	Lists the U.S. towns closest to the cursor location.
W	Toggles the graphics frame on and off.
X	Tracks cloud drift wind targets with one keystroke.
Y	Toggles image frames to and from the loop control system.
Z	Toggles the McIDAS-X zoom function on and off.
ZM	Sets the zoom factor to use with the Z command.

Graphics

Commands in this category manipulate graphics colors and settings, restore saved graphics, and draw text, symbols and shapes. Note that many commands in other categories also draw graphics.

Command	Purpose
CM	Circulates the graphics color table.
CW	Fills or erases the graphics in the cursor.
EG	Erases graphics and images.
ERASE	Erases graphics, images or entire frames.
FONT	McIDAS graphics font utility.
FRMLABEL	Displays an image label on a frame.
GD	Defines graphics display parameters.
GU	Graphics utility.
MAP	Displays a map and latitude/longitude lines on a graphics frame.
SHOWVG	Displays virtual graphics.
ZA	Writes text and/or symbols on the graphics frame.
ZLM	Draws graphics using the mouse.

Frame Analysis

Commands in this category use navigation, calibration or directory information from data displayed on a frame to list or draw data, or move the cursor. Many of the commands are also listed under other categories.

Command	Purpose
BAR	Draws a grayscale bar on a frame containing an image.
C	Lists a frame's directory.
D	Lists the digital values at the cursor center.
DIST	Finds the distance on a navigated frame.
E	Lists the earth coordinates at the cursor center.
FIND	Locates U. S. towns by name and state.
FRMLABEL	Displays an image label on a frame.
FRMLIST	Lists frame directory information.
IMGPLOT	Displays a scatter plot of data values from two images.
IMGPROBE	Lists or plots image data for a region of the displayed image.
MAP	Displays a map and latitude/longitude lines on a graphics frame.
PC	Positions the cursor at a desired point on the frame.
PCMW	Computes a velocity vector.
PLAX	Calculates parallax at cursor position.
S	Lists Metar, Synoptic and RAOB stations within the cursor bounds.
STNLIST	Lists weather stations.
STNPLOT	Plots stations on a navigated frame.
TOWNS	Lists the U.S. towns closest to the cursor location.
V	Toggles the velocity cursor on and off when tracking cloud drift winds.

Image and Graphic Coloring

Commands in this category are used to change image and graphics colors.

Command	Purpose
CM	Circulates the graphics color table.
EB	Black and white contrast stretching.
EU	Enhancement utility.
GU	Graphics utility.

Weather Analysis

Commands in this category list or display weather information. Those that deal with a specific data type (image, grid, point, or text) also appear in those categories.

Command	Purpose
ADT	Estimates tropical cyclone intensity using the Advanced Dvorak Technique.
CHILL	Calculates the wind chill index.
FRNTDISP	Displays fronts from NCEP frontal positions products.
HEAT	Calculates the heat index and relative humidity.
HODO	Plots a hodograph of upper-air data.
MOSRPT	Lists MOS forecasts from the GFS, NAM and NGM models.
OBSRPT	Lists the observational text reports.
RAOBCON	Contours upper-air data from IRAB-schema point files.
RAOBPLOT	Plots upper-air data from IRAB-schema point files.
RAOBRPT	Lists upper air (RAOB) observations.
SFCCON	Contours surface hourly data from ISFC-schema point files.
SFCLIST	Lists surface hourly data from point files.
SFCMG	Displays a surface meteorogram.
SFCPLOT	Plots surface hourly data from ISFC-schema point files.
SFCRPT	Lists surface hourly observations.
SYNRPT	Lists synoptic observations.
TAFRPT	Lists terminal aerodrome forecasts.
UACROSS	Plots a vertical cross section of upper-air data.
UALIST	Lists upper air data.
UAPLOT	Plots a sounding on a thermodynamic diagram.
WWDISP	Displays active weather warnings, watches and advisories.
WWLIST	Lists active weather warnings, watches and advisories.
WXTLIST	Lists weather text products.

Image

Commands in this category operate on image data, like satellite and radar images.

Command	Purpose
ADT	Estimates tropical cyclone intensity using the Advanced Dvorak Technique.
ADUMP	Lists the contents of an area in hexadecimal.
AVGI	Reduces image resolution by averaging.
AXFORM	Converts a McIDAS Area file to ASCII or binary flat files.
BAR	Draws a grayscale bar on a frame containing an image.
D	Lists the digital values at the cursor center.
EB	Black and white contrast stretching.
EU	Enhancement utility.
GEO	Finds and displays a geostationary satellite image over a point or region.
GRDIMG	Converts a grid to a digital image.
IMGCHA	Changes an image directory.
IMGCOPY	Copies image data from one dataset to another.
IMGDEL	Deletes images in a dataset.
IMGDISP	Displays image data.
IMGFILT	Applies filters to image data.
IMGLIST	Lists images in a dataset.
IMGMAG	Changes the magnification of a displayed image.
IMGOPER	Applies mathematical functions to image data.
IMGPLOT	Displays a scatter plot of data values from two images.
IMGPROBE	Lists or plots image data for a region of the displayed image.
IMGREMAP	Remaps images into different projections.
IMGRGB	Combines images and basemaps into a multi-banded image.
IMGTRACK	Redisplays a loop of images centered at a selected feature.
LEO	Finds and displays a polar orbiting satellite image over a point or region.
MAKNAV	Creates a navigation entry for an area.
MAP	Displays a map and latitude/longitude lines on a graphics frame.
PCMW	Computes a velocity vector.
PRDUTIL	Calibration utility for 1-byte PRD image files.
SU	Image data stretching utility.
TIRCAL	Calibrates POES (TIROS) areas.
V	Toggles the velocity cursor on and off when tracking cloud drift winds.
VASTBL	Creates calibration transforms for Mode AA VAS data.

Point

Commands in this category operate on point data, including McIDAS MD files containing conventional data such as synoptic, surface hourly, ship, buoy, and upper-air observations.

Command	Purpose
ALTIM	Calculates sounding data at standard aviation heights or pressures.
DOQTL	Selectively deletes MD files.
HODO	Plots a hodograph of upper-air data.
ISENT	Calculates theta surface levels for upper air data.
LSCHE	Lists an MD file schema.
MDE	Edits MD files.
MDU	MD file utility.
MDX	Plots or grids data from MD files or grids.
MFPLOT	Plots a frequency distribution of MD file data.
PTCON	Converts point data to grids; displays contours and plots.
PTCOPY	Copies point data from one dataset to another.
PTDISP	Displays point data.
PTLIST	Lists point data in a dataset.
RAOBCON	Contours upper-air data from IRAB-schema point files.
RAOBPLOT	Plots upper-air data from IRAB-schema point files.
SCHE	Adds a schema to the schema file.
SFCCON	Contours surface hourly data from ISFC-schema point files.
SFCLIST	Lists surface hourly data from point files.
SFCMG	Displays a surface meteorogram.
SFCPLOT	Plots surface hourly data from ISFC-schema point files.
TXT2MD	Creates an MD file from an ASCII text data file.
UACROSS	Plots a vertical cross section of upper-air data.
UALIST	Lists upper air data.
UAPLOT	Plots a sounding on a thermodynamic diagram.

Grid

Commands in this category operate on grid data, including numerical model forecast grids such as NAM, GFS, and RUC.

Command	Purpose
GRDCOPY	Copies grids from one dataset to another.
GRDDISP	Generates contours, streamlines or plots from grid data.
GRDIMG	Converts a grid to a digital image.
GRDINFO	Lists, modifies, or statistically analyzes grid point values.
GRDLIST	Lists grids in a dataset.
IGG	Grid utility.
IGU	Grid file utility.
MDX	Plots or grids data from MD files or grids.
PTCON	Converts point data to grids; displays contours and plots.
UACROSS	Plots a vertical cross section of upper-air data.

Text

Commands in this category operate on ASCII text files, LW format files and weather text data such as that from the National Weather Service NOAAPORT data stream.

Command	Purpose
DOSTOLW	Converts ASCII text format files to LW format text files.
FRNTDISP	Displays fronts from NCEP frontal positions products.
LWCON	Converts a McIDAS-OS2 LW file to a McIDAS-X LW file.
LWTODOS	Converts LW format files to ASCII text format files.
LWU	LW file utility.
MOSRPT	Lists MOS forecasts from the GFS, NAM and NGM models.
OBSRPT	Lists the observational text reports.
RAOBRPT	Lists upper air (RAOB) observations.
READ	Lists the contents of an ADDE text dataset.
SEE	Lists the text in an LW file.
SFCRPT	Lists surface hourly observations.
SYNRPT	Lists synoptic observations.
TAFRPT	Lists terminal aerodrome forecasts.
TFILE	Sends text output to a disk file.
TXT2MD	Creates an MD file from an ASCII text data file.
WWDISP	Displays active severe thunderstorm and tornado watches and warnings.
WWLIST	Lists active severe thunderstorm and tornado watches and warnings.
WXTLIST	Lists weather text products.

Navigation

Commands in this category list or display data from navigation datasets. See *Frame Analysis* for commands that use the navigation information of data displayed on a frame.

Command	**Purpose**
NAVCALC	Calculates satellite subpoints and orbit times.
NAVDISP	Displays polar orbital tracks and coverage.
NAVLIST	Lists system navigation entries.
PACORB	Displays polar orbital tracks and coverage for the Pacific Test Range.
USAORB	Displays polar orbital tracks and coverage for North America.

Tools

Commands in this category manipulate tools like the F Key Menu system, Soft Tablet, McBASI programs, BATCH files, strings, command schedulers and saved loops. These tools can be used to help simplify or automate complicated procedures. This category also includes commands to start the McIDAS-X GUI.

Command	Purpose
ASK1	Defines a question to prompt the user for string input.
BATCH	Runs commands serially from a text file.
DEFLOOP	Lists, saves and restores loops.
GUI	Starts the McIDAS-X Graphical User Interface.
MCEVAL	Runs a McIDAS command substituting McIDAS string values.
MCMENU	Starts the F Key Menu System.
MOVIE	Displays a sequence of loops.
REPEAT	Repeats a McIDAS command.
RUN	Runs a McBASI program.
SKE	Enters a McIDAS command in the system command scheduler.
SKED	Starts a scheduler process.
SKL	Lists command scheduler entries.
SKU	Command scheduler utility.
TD	Deletes one or more strings from the current string table.
TE	Enters a string in the string table.
TL	Lists the strings in a string table.
TU	String table utility.

Administrative Utilities

Commands in this category configure the McIDAS session so it can access data on other workstations, or share data with other workstations.

Command	Purpose
AKA	Adds, deletes and lists aliases for dataset names.
DATALOC	Manages and lists the client routing table.
DSINFO	Lists datasets on local and remote servers.
DSSERVE	Manages and lists dataset names on the local server.
REDIRECT	Writes the LW file path redirection memory block.
SYSKEY	Sets system values for local McIDAS workstation networks.
SYSVAL	SYSKEY utility.
UCU	Inspects the User Common (UC) memory block.

ADDE Commands

Commands in this category operate on ADDE datasets.

Command	Purpose
ADT	Estimates tropical cyclone intensity using the Advanced Dvorak Technique.
AKA	Adds, deletes and lists aliases for dataset names.
ALTIM	Calculates sounding data at standard aviation heights or pressures.
DATALOC	Manages and lists the client routing table.
DSINFO	Lists datasets on local and remote servers.
DSSERVE	Manages and lists dataset names on the local server.
FRNTDISP	Displays fronts from NCEP frontal positions products.
GEO	Finds and displays a geostationary satellite image over a point or region.
GRDCOPY	Copies grids from one dataset to another.
GRDDISP	Generates contours, streamlines or plots from grid data.
GRDIMG	Converts a grid to a digital image.
GRDINFO	Lists, modifies, or statistically analyzes grid point values.
GRDLIST	Lists grids in a dataset.
HODO	Plots a hodograph of upper-air data.
IMGCHA	Changes an image directory.
IMGCOPY	Copies image data from one dataset to another.
IMGDEL	Deletes images in a dataset.
IMGDISP	Displays image data.
IMGFILT	Applies filters to image data.
IMGLIST	Lists images in a dataset.
IMGMAG	Changes the magnification of a displayed image.
IMGOPER	Applies mathematical functions to image data.
IMGPLOT	Displays a scatter plot of data values from two images.
IMGPROBE	Lists or plots image data for a region of the displayed image.
IMGREMAP	Remaps images into different projections.
IMGRGB	Combines images and basemaps into a multi-banded image.
IMGTRACK	Redisplays a loop of images centered at a selected feature.
LEO	Finds and displays a polar orbiting satellite image over a point or region.
MOSRPT	Lists MOS forecasts from the GFS, NAM and NGM models.
NAVCALC	Calculates satellite subpoints and orbit times.
NAVDISP	Displays polar orbital tracks and coverage.
NAVLIST	Lists system navigation entries.
OBSRPT	Lists the observational text reports.
PACORB	Displays polar orbital tracks and coverage for the Pacific Test Range.
PTCON	Converts point data to grids; displays contours and plots.
PTCOPY	Copies point data from one dataset to another.
PTDISP	Displays point data.
PTLIST	Lists point data in a dataset.
RAOBCON	Contours upper-air data from IRAB-schema point files.
RAOBPLOT	Plots upper-air data from IRAB-schema point files.
RAOBRPT	Lists upper air (RAOB) observations.
READ	Lists the contents of an ADDE text dataset.
SFCCON	Contours surface hourly data from ISFC-schema point files.
SFCLIST	Lists surface hourly data from point files.

SFCMG	Displays a surface meteorogram.
SFCPLOT	Plots surface hourly data from ISFC-schema point files.
SFCRPT	Lists surface hourly observations.
SYNRPT	Lists synoptic observations.
TAFRPT	Lists terminal aerodrome forecasts.
UACROSS	Plots a vertical cross section of upper-air data.
UALIST	Lists upper air data.
UAPLOT	Plots a sounding on a thermodynamic diagram.
USAORB	Displays polar orbital tracks and coverage for North America.
WWDISP	Displays active weather warnings, watches and advisories.
WWLIST	Lists active weather warnings, watches and advisories.
WXTLIST	Lists weather text products.

Commands by Alphabetical Order

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Miscellaneous

Miscellaneous

& (Ampersand), ^ (Caret), + (Plus), ! (Exclamation Point), ? (Question Mark), / (Slash)

A

A, ADT, ADUMP, AKA, ALTIM, ASK1, AVGI, AXFORM

B

B, BAR, BATCH, BRIEF

C

C, CASEMODE, CCODE, CHILL, CLEAR, CM, COLORS, COMBINE, CU, CUR, CW

D

D, DATALOC, DATELIST, DEFLOOP, DIST, DMAP, DOQTL, DOSTOLW, DR, DSINFO, DSSERVE

E

E, EB, ECHO, EDITMODE, EG, ERASE, EU, EXIT

F

F, FIND, FONT, FRMLABEL, FRMLIST, FRMSAVE, FRNTDISP

G

GD, GEO, GLOBKEY, GRDCOPY, GRDDISP, GRDIMG, GRDINFO, GRDLIST, GU, GUI

H

HEAT, HELP, HODO

I

IGG, IGU, IMGCHA, IMGCOPY, IMGDEL, IMGDISP, IMGFILT, IMGLIST, IMGMAG, IMGOPER, IMGPLOT, IMGPROBE, IMGREMAP, IMGRGB, IMGTRACK, ISENT

J

J

K

K, KILL

L

L, LB, LEO, LOGOFF, LOGON, LS, LSCHE, LVF, LWCON, LWTODOS, LWU

M

MAKFRM, MAKNAV, MAP, MCEVAL, MCMENU, MDE, MDU, MDX, MFPLOT, MOSRPT, MOVIE

N

NAVCALC, NAVDISP, NAVLIST

O

O, OBSRPT, OS

P

PACORB, PANEL, PC, PCMW, PLAX, PRDUTIL, PTCON, PTCOPY, PTDISP, PTLIST

Q

R

R, RAOBCON, RAOBPLOT, RAOBRPT, READ, REDIRECT, REPEAT, RUN, RVF

S

S, SCHE, SEE, SF, SFCCON, SFCLIST, SFCMG, SFCPLOT, SFCRPT, SG, SHOWVG, SKE, SKED, SKL, SKU, STNLIST, STNPLOT, SU, SUNLIT, SVF, SYNC, SYNRPT, SYSKEY, SYSVAL

T

TAFRPT, TD, TE, TERM, TFILE, TFIND, TIMEOUT, TIRCAL, TL, TOWNS, TU, TXT2MD

U

UACROSS, UALIST, UAPLOT, UCU, USAORB

V

V, VASTBL

W

W, WWDISP, WWLIST, WXTLIST

X

X

Y

Y

Z

Z, ZA, ZLM, ZM

& (Ampersand)

Recalls a previously entered command.

Remarks

The ampersand (&) displays a previously entered command on the screen's command line. You can backtrack up to 100 commands by repeatedly typing the ampersand.

You can modify the command line before entering the command.

The ampersand command works in this manner only if you have not typed any other characters on the command line.

See Also

The ^ (Caret) command tracks previously entered commands in the opposite direction.

---

^ (Caret)

Recalls previously entered commands.

Remarks

To recall commands in the direction opposite the ampersand (&),

Type: ^

Each press of the caret key backs up the command queue by one.

See Also

See the & (Ampersand) command description for more information.

+ (Plus)

Raises the Image Window.

Remarks

Use the numeric keypad's plus (+) key to raise (display on top of any overlapping windows) the Image Window. To raise the Text and Command Window, press a numeric keypad number. The number you press determines which text frame is displayed in the window.

! (Exclamation Point)

Sends a command to the Unix shell.

Format

! *command*

Parameter

command	Unix command to run; spaces between the exclamation point and the Unix command are optional

---

Remarks

This command lets you run a Unix command from the McIDAS-X Text and Command Window. The Unix command should include the correct path, if needed. The Unix command should not invoke a program, such as Telnet, that expects a user input while running.

Once you type the exclamation point (!), the case mode automatically changes to the opposite mode. For example, if you typed the previous command in upper case, all characters entered after the exclamation point (!) appears in lowercase unless you use the **Shift** or **Caps Lock** key.

This command performs the same function as the OS command, but differs in the following ways:

- You don't need to preface the Unix command with a space and double quotation mark (").
- The case mode is automatically switched.

See Also

The OS command also sends commands to the Unix shell.

Examples

! ls -1 /data/AREA*

This entry runs the Unix **ls** command to list all file names beginning with AREA in the **/data** directory.

! echo $MCPATH

This command runs the Unix **echo** command to list the current setting of the **MCPATH** environment variable.

? (Question Mark)

Lists the status of McIDAS commands and processes.

Remarks

The question mark (?) command lists all the McIDAS commands and processes running in your Unix account. The information listed depends on the workstation type. Output information is described below.

Header	Description
UID or USER	user ID of the person who started the process
PID	process identification number to use when stopping a command
PPID	parent process ID number
C	processor utilization for scheduling
TT or TTY	controlling workstation for the process
STAT	state of the process
STIME	start time of the process
TIME	amount of CPU time used
CMD or COMMAND	process or command; most McIDAS commands are listed in uppercase

Use command KILL to stop processes. Do not stop the IMPORT, SKED, or TCPIP processes; they are required to run McIDAS-X.

/ (Slash)

Stops McIDAS commands.

Format

/ *pid1 . . pidn*

Parameter

pid process ID numbers of the commands to stop

Remarks

To stop McIDAS commands, enter a slash followed by the Process IDentification (PID) numbers. If you don't know the PID numbers, use the ? (Question Mark) command to list them.

Examples

/ 374

This entry terminates the McIDAS command running under PID number 374.

/ 27347 27348 34272

This entry terminates McIDAS commands running under PID numbers 27347, 27348, and 34272.

A

Advances one frame.

Remarks

To advance the frame position by one,

Press: **Alt A**

or

Type: **A**

Press: **Enter**

ADT

Estimates tropical cyclone intensity using the Advanced Dvorak Technique.

Format

ADT *[keywords]*
ADT *history_file [keywords]*
ADT *history_file* DATASET=*source local [keywords]*
ADT AUTO *format fcst_file history_file [keywords]*
ADT DEL *history_file* DATE= *[keywords]*
ADT EDIT *history_file day time [keywords] "comment*
ADT LIST *history_file [keywords]*
ADT PLOT *history_file [keywords]*

Parameters

AUTO allows for automated ADT estimate utilizing NHC/JTWC forecasts and Laplacian analysis to objectively determine position of storm center

DEL deletes history file record; must be used with DATE= keyword to define date/time limits

EDIT edits contents of history file

LIST lists contents of history file; see the Remarks

PLOT plots a graph of intensity estimates from history file; see the Remarks

history_file history file to be used (default=path defined in environment variable **ODTHISTORY**)

format format of forecast file; the valid options are:

- **ATCF** tropical cyclone forecast records (default)
- **DISC** TPC WTNT4* or WTPZ3* storm specific discussions
- **PACWARN** JTWC WTPN3* tropical cyclone warning
- **GENERIC** see the Remarks for example format

fcst_file file name containing forecast information for the AUTO option; the directory location is defined by environment variable **ODTAUTO** (no AUTOFIX)

day adds comment to specified day in history file

time adds comment to specified time in history file

comment comment to add to history file for specified day and time

---

Keywords

ATCF= *stormID sourceID rawT finalT*
produces ATCF formatted listing for current ADT estimate or entire history file, or for portion of history file as defined by DATE= keyword (default=-1 XXXX 0 0), where:

stormID annual tropical cyclone number and basin value, e.g., 10W, 06L; valid basin values are:
L=Atlantic
E=East Pacific

C=Central Pacific
W=West Pacific
A=North Indian Ocean
B=South Indian Ocean
S=Australian Region (90E-135E)
P=Australian Region (east of 135E)

sourceID WMO identifier (e.g., NHC, PGTW, etc)

rawT flag determining raw T number output; specify 0 or 1, where 0=adjusted raw T number, 1=unadjusted raw T number

finalT flag determining final T number output; specify 0 or 1, where 0=6-hour weighted average, 1=3-hour straight linear average

the output file is created in the directory defined by the environment variable **ODTOUTPUT** using the naming convention *CIMSS_DVTO_yyyymmddhhmm_ssbasin_FIX*, where *yyyy*=year, *mmdd*=month (1-12) and day (01-31), *hhmm*=UTC time, *ss*=stormID, *basin*=sub basin

COLor= *CI MSLP T6hr T3hr AdjRawT UnAdjRawT 12hrAvgTIE CurrTIE outline text*
defines graphic color levels of the PLOT option's plotted parameters; specify 0 to omit from display (default=1 2 0 0 0 0 0 0 6 7)

DATe= *bday btime eday etime*
defines day and time range (from beginning day and time to ending day and time) for LIST, PLOT and DEL options; specify *bday* and *eday* in *yyyymondd* format (e.g., 2008SEP17) and *btime* and *etime* in *hhmmss* format (e.g., 131500)

DATAset= *source local*
source and local ADDE datasets; specify as *group/descriptor.position* or *alias.position*, where:

source identifies the infrared image to use as input for the ADT analysis (default=value defined by environment variable **ODTLSERVE**)

local identifies the dataset to store the subsected image; the latitude and longitude location is determined by information in *history_file* (default=value defined by environment variable **ODTLDATA**)

DOMain= defines domain for CI to MSLP conversion; valid options are **ATL** or **PAC**; see the Remarks (default=computed)

LAND= **YES** applies ADT land interaction rule (default)

NO does not apply ADT land interaction rule

MAXWradius= manually defines maximum wind radius in km; if the ADT estimate has determined an eye scene and a MAXW of less than 5 km is specified, the scene type will be changed to PINHOLE EYE; if a value of MAXW of greater than 45 km is specified, the scene type will be changed to LARGE EYE

OUTput= *loc file*
defines destination of LIST option output, where:

loc location to send the output; valid options are T, FILE and TFILE (default=T), where:
T - sends output to current text window
FILE - sends output to a file
TFILE - sends output to both current text window and a file

file directory and name of file created by FILE and TFILE options; the default filename is **ODTDUMP.ASC** and the directory it is written to is defined by the environment variable **ODTOUTPUT**

OVERride= **YES** interactively overrides the automated scene type; cannot be used with AUTO option

NO does not override the automated scene type (default)

RAWT= overrides the initial raw T intensity classification value as defined by the subjective Dvorak rules; this keyword should be used only when analysis of a tropical cyclone is initiated at a time later than initial formation (storm has a raw T value greater than 1.0); if 0.0 is specified in this keyword, the value will be calculated by ADT (default=1.0)

SEARch= **YES** performs search for maximum curved band location if ADT estimate has determined a CURVED BAND scene

		(default)
	NO	does not perform search for maximum curved band location
WIND=	**YES**	use with the PLOT option to plot its intensity values as maximum wind speed in knots rather than MSLP; speed and MSLP values are derived from the empirical relationship to CI number defined in *Dvorak* [1984] and *Shwechuck and Weir* [1980]
	NO	plots the intensity values as MSLP instead of wind speed (default)

Remarks

For a description of the ADT algorithm and detailed information about the ADT command and its options, see the *ADT Users' Guide*: http://cimss.ssec.wisc.edu/tropic/adt/source/adt.html

The units for pressure/wind are shown in millibars/knots for both the LIST and PLOT options. If the LIST option is used with ATCF keyword, the output file will be in fixed DVTO format.

When using the DOMAIN= keyword, ATL should be used for storms within the North Atlantic basin, while PAC should be used for storms within the Western Pacific basin. Domain selection will affect Raw T intensity estimates and corresponding CI pressure values (empirical CI number/pressure relationship).

Examples

ADT

This entry uses the current displayed image and current cursor location to perform an abbreviated ADT analysis. No time averaging or any other application will be performed. Output will be displayed in the current text frame and no history file will be written.

ADT KATRINA RAWT=6.0

This entry uses the current displayed image and current cursor location to perform a full ADT analysis with the initial intensity classification of 6.0. The output will be displayed in the current text frame and written to the file **KATRINA.ODT**.

ADT AUTO DISC katrina2005.280300.txt KAT05.ODT RAWT=0.0

This entry performs an automated full ADT analysis using the currently displayed image, and the initial intensity classification will be calculated. The storm center position will be read from the forecast product file, **katrina2005.280300.txt**. The output will be displayed in the current text frame and written to the file **KAT05.ODT**.

ADUMP

Lists the contents of an area in hexadecimal.

Format

ADUMP *area bline eline bele eele [keywords]*

Parameters

area	area number of the data to list
bline	beginning area line to list (default=1)
eline	ending area line to list (default=1)
bele	beginning area element to list (default=1)
eele	ending area element to list (default=50)

Keywords

DATa=	**ALL**	lists the prefix and data
	DOC	lists the documentation section of the prefix
	LNE	lists the area line data (default)
	LVL	lists the level section of the prefix
	NVC	lists the line data when the validity code is bad
	PFX	lists the prefix (validity code, documentation section and level section)
	VLC	lists the validity code of the prefix
FACt=	factor by which to divide data; useful for shifting 10-bit data (default=1)	
FORm=	**ALL**	lists the entire area directory
	STND	lists the abbreviated area directory (default)

Remarks

Line and element numbers are in area coordinates with the first line and element of the area being 0,0.

Lines with a bad validity code are displayed as zeros unless the NVC option is used.

When listing multibanded data, the actual number of band elements listed is the number of elements requested times the number of bands. For example, if area 8016 has five bands, the command ADUMP 8016 10 10 200 219 lists the 100 band elements present in area elements 200 to 219 on area line 10.

You can enter multiple options with keyword DATA.

Examples

ADUMP 100 240 340 100 300

This entry lists the data in elements 100 to 300 of lines 240 to 340 in area 100.

ADUMP 100 240 240 0 100 DATA=ALL

This entry lists the data and prefixes in elements 0 to 100 of line 240 in area 100.

ADUMP 100 0 200 0 100 DATA=VLC

This entry lists the validity codes in elements 0 to 100 of lines 0 to 200 in area 100.

ADUMP 104 1200 1230 1 400 DATA=NVC LNE

This entry lists the data and validity codes (even if they are bad) in elements 1 to 400 of lines 1200 to 1230 in area 104.

AKA

Adds, deletes and lists aliases for dataset names.

Formats

AKA ADD *alias dataset*
AKA DEL *alias*
AKA LIST *alias [keyword]*

Parameters

ADD	adds an alias for an ADDE dataset name
DEL	deletes an alias for an ADDE dataset name
LIST	lists the aliases (default)
alias	alias name; up to 24 characters (no default for ADD and DEL; default=all aliases for LIST)
dataset	ADDE dataset name that the alias represents; specify in group/descriptor format

---

Keyword

GROup=	lists all aliases for the specified group name (default=all groups)

Remarks

AKA adds, deletes or lists aliases in the client routing table.

You cannot assign the same alias to more than one ADDE dataset name because the ADD option overwrites an existing alias without warning. You may, however, assign more than one alias to an ADDE dataset name.

An alias can be used in place of the ADDE dataset name (group/descriptor) in other commands.

Examples

AKA

This entry lists all aliases and the dataset names they represent.

AKA ADD VIS RTSERV/G8-VIS-CONUS

This entry creates an alias named VIS that can be used to represent the dataset name RTSERV/G8-VIS-CONUS in other commands.

AKA ADD R RADAR/MDR_IMAGES

This entry creates an alias named R that can be used to represent the dataset name RADAR/MDR_IMAGES in other commands.

AKA LIST GROUP=RTSERV

This entry lists all aliases for dataset names in the group RTSERV.

AKA LIST VIS

This entry lists the alias VIS and the dataset name it represents.

AKA DEL VIS

This entry deletes the alias VIS, but does not affect the dataset that the alias represents.

ALTIM

Calculates sounding data at standard aviation heights or pressures.

Format

ALTIM *ddataset time day [keywords]*

Parameters

ddataset	destination ADDE point dataset name and absolute position; specify as *group/descriptor.position* or *alias.position*; only positive integers are valid for position
time	observation time (default=0 if current time is between 0 UTC and 11:59 UTC, otherwise default=12)
day	observation date; valid formats include YYYYDDD, YYYY/MM/DD and DD/MON/YYYY where YYYY, DDD, MM and DD are integers and MON is three or more letters, e.g., JAN or JANUARY (default=current)

Keywords

COuntry=	*c1 . . cN*	calculate and write data for soundings in these countries; specify up to six countries using their two-letter codes; use the CCODE command to list valid codes; see the Remarks (default=all countries)
DATaset=	*mand sig*	source ADDE point dataset names for mandatory- and significant-level data; specify as *group/descriptor* or *alias*; you must specify NONE in the *sig* parameter or SIG=NO keyword to use only mandatory data, i.e., ignore the significant data (*mand* default=RTPTSRC/UPPERMAND; *sig* default=RTPTSRC/UPPERSIG if *mand*=RTPTSRC/UPPERMAND, otherwise *sig* default=NONE)
DELete=	**YES**	delete the destination file before copying
	NO	do not delete the destination file before copying (default)
LAT=	*slat nlat*	calculate and write data for soundings in this latitude range, where *slat* and *nlat* are the southern and northern latitude limits, respectively; see the Remarks (default= –90 90)
LON=	*elon wlon*	calculate and write data for soundings in this longitude range, where *elon* and *wlon* are the eastern and western longitude limits, respectively; see the Remarks (default= –180 180)
OPTion=	**H**	calculate and write data at heights from 0 to 45,000 feet at 1,000 foot increments; see the Remarks (default)
	P	calculate and write data at standard pressures from 18,000 to 45,000 feet at 1,000 foot increments
SIG=	**YES**	include data from the significant-level dataset specified in the DATASET keyword (default)
	NO	do not include data from the significant-level dataset specified in the DATASET keyword; use only its mandatory-level dataset
STate=	*s1 . . sN*	calculate and write data for soundings in these states; specify up to six states using their two-letter state Post Office abbreviations; see the Remarks (default=all states)
TITLE=	*'text'*	title of the destination point file; must be 32 characters or less (default=Altimeter Interpolation: *yyyyddd*); *yyyyddd* in the default is the date of the source dataset

---

Remarks

ALTIM reads data from the source upper-air dataset and interpolates the values to standard aviation heights or pressures, then writes the data to the

destination dataset (an IHGT schema MD file). The IHGT schema must be registered with command SCHE DCIHGT before you use this command. If the specified destination file already exists, you must delete it with DEL=YES since ALTIM cannot append data to an existing file.

Because of the large number of calculations performed for each sounding, ALTIM may take a long time to complete. You can minimize this time and the number of soundings by specifying only the needed data with the LAT and LON, COUNTRY or STATE keywords.

Standard pressures, as specified with OPT=P, are the pressures of the standard atmosphere from 18,000 feet to 45,000 in 1000 foot increments (i.e., 18000, 19000, 20000, etc.). Use OPT=P to calculate the data at these "standard atmosphere" pressures. Use OPT=H to calculate the data at heights from 0 to 45,000 feet at 1000 foot increments, using the heights in the source dataset (converted to thousands of feet), not the standard atmosphere pressure model.

Examples:

ALTIM LOCAL/AVIATION.10 0

This entry calculates upper-air data from 0 to 45,000 feet at 1000 foot increments using today's 0 UTC data in the default source datasets RTPTSRC/UPPERMAND and RTPTSRC/UPPERSIG. The data is written to position 10 of dataset LOCAL/AVIATION, an IHGT-schema point file.

ALTIM LOCAL/AVIATION.10 12 DEL=YES ST=FL TITLE='12 UTC IHGT data for Florida' OPT=H

This entry calculates upper-air data at standard pressures from 18,000 to 45,000 feet at 1000 foot increments for the state of Florida. It uses today's 12 UTC data in the default datasets RTPTSRC/UPPERMAND and RTPTSRC/UPPERSIG. The data is written to position 10 of dataset LOCAL/AVIATION, an IHGT-schema point file. The DEL=YES and TITLE keywords delete the existing destination point file and create a new one with the title *12 UTC IHGT data for Florida*.

ASK1

Defines questions to prompt the user for string input.

Format

ASK1 *string1* *'question1'* . . *'stringN* *'questionN'*

Parameters

string1 . . *stringN*	string names to place the user's answers in
'question1' . . *'questionN'*	the corresponding questions to ask; quotes are required; use uppercase and/or lowercase letters

---

Remarks

The ASK1 command defines one or more questions and generates windows that prompts the user for a value for each of the questions. The values entered in the window are entered in the strings named by ASK1. After entering the values, click the OK button or press Enter. Everything you typed in the window is placed in the corresponding strings. To cancel the command, press the Cancel button or press Esc; the value of the strings will not change.

Returned strings greater than 160 characters are truncated to 160 characters.

ASK1 is well suited for the user interface tools like the F Key Menu System.

If the *string* already exists, its current value appears so you can edit it. Thus, it's easier to change an existing string's value with ASK1 than with the TE command.

Examples

ASK1 TIME 'Enter time for plot:'

This entry generates a window with the prompt "Enter time for plot:". Entering text and clicking the OK button or pressing the Enter key enters the new value in the string TIME. If no text is entered and the Cancel button is clicked, the string TIME does not change.

AVGI

Reduces image resolution by averaging.

Format

AVGI *sarea darea lres eres [keyword]*

Parameters

sarea	source area to average
darea	destination area of the averaged data
lres	resolution reduction for lines (default=1)
eres	resolution reduction for elements (default=*lres*)

Keyword

BAND=	band number from multibanded areas (default=lowest numbered band)

Remark

The area size cannot exceed 26000 elements.

Examples

AVGI 12 15

This entry averages area 12 by one line and element and places the new area in area 15.

AVGI 10 20 4

This entry blows down area 10 four times by averaging pixel values. The new area number is 20.

AXFORM

Converts a McIDAS Area file to ASCII or binary flat files.

Format

AXFORM *area file [keywords]*

Parameters

area area number

file output file; maximum of 100 characters; specify as a file name or a fully qualified path (directory and file name); do not include an extension with the file name; see the Remarks

---

Keywords

BANd= band numbers for a multibanded *area* (default=first band)

- **ALL** all bands of *area*

FLIp=

- **YES** flip bytes; specify this keyword when the byte order on the destination system is the reverse of the source system, e.g., from a PC to a Unix workstation
- **NO** do not flip bytes (default)

FORmat=

- **I1** output elements are 1-byte integers
- **I2** output elements are 2-byte integers
- **I4** output elements are 4-byte integers

(default=see the Remarks)

FTYpe=

- **ASC** ASCII text files with 25I10 format
- **BIN** binary files, as specified in keyword FORMAT (default)
- **BOTH** both ASCII and binary file types

MISs= missing data value for .LAT/.LON extension files (default=65535 Hex FFFF)

NAV=

- **YES** *inc* produces .LAT/.LON data files with an element increment; these files contain LAT/LON stored as degrees*100 (2 bytes per element; the increment is the number of data elements associated with each LAT/LON element (default=NO 1)

UNIt= physical quantity of the output files; see the Remarks (default=calibration units of *area*)

- **ALL** all calibration units defined by *area*

Remarks

AXFORM creates a set of files containing data from *area*. These files can be ported to other non-McIDAS systems. Data is represented as 2-dimensional arrays of binary or ASCII fields with the same dimensions as the original *area* lines/elements.

An ASCII output text file documents the format of the output data files, and contains a listing for each data file generated. Output files are assigned the extensions below. The definitions for .LAT, .LON, .ALT, and .ALN assume keyword NAV=YES.

Extension	Meaning
.HDR	header file containing ASCII text describing area and the contents of the output data files
.nnn	data file containing binary data for a single calibration of a specific band; the range for *nnn* is 001 to 999
.A*nn*	data file containing ASCII data for a single calibration of a specific band; the range for *nn* is 01 to 99
.LAT	binary file containing a 2-dimensional array of latitudes corresponding to the elements in the output data files
.LON	binary file containing a 2-dimensional array of longitudes corresponding to the elements in the output data files
.ALT	ASCII file containing a 2-dimensional array of latitudes corresponding to the elements in the output data files
.ALN	ASCII file containing a 2-dimensional array of longitudes corresponding to the elements in the output data files

The defaults for keyword FORMAT are shown below. They assume FTYPE=BIN or BOTH.

If UNIT=	Then FORMAT=
BRIT	I1 (1-byte integer)
TEMP	I2 (2-byte integer)
RAD	I4 (4-byte integer)
ALB	I2 (2-byte integer)
RAW	number of bytes/element from area directory
other	I1 (1-byte integer)

Examples

AXFORM 12 GOES_VIS

This entry creates a binary file of brightness values from area 12 and places the information in **GOES_VIS.001**.

AXFORM 12 GOES_VIS FTYPE=BOTH

This entry creates binary and ASCII files of the brightness values from area 12. The data is placed in **GOES_VIS.001** and **GOES_VIS.A01**.

AXFORM 43 MULTBAND BAND=2 3 5 UNIT=TEMP NAV=YES 45

This entry creates binary files for the TEMP values from bands 2, 3 and 5 of area 43. The information is placed in **MULTBAND.001**, **MULTBAND.002** and **MULTBAND.003**, respectively. Latitude and longitude grid files are also created and placed in **MULTBAND.LAT** and **MULTBAND.LON**. The lat/lon resolution is 1 lat/lon point for every 45 image elements.

B

Backs up one frame position.

Remarks

To back up the frame position by one,

Press: **Alt B**

or

Type: **B**

Press: **Enter**

BAR

Draws a grayscale bar on a frame containing an image.

Format

BAR *frame [keywords]*

Parameter

frame frame number (default= current frame)

Keywords

ANNot= **NO** do not write text annotation, only the bar and label (default)

YES *justify line ele color size "text*

writes text annotation on frame, where:

justify justification of text; specify **L**, **R** or **C**, where L=left, R=right, C=center (default=C)

line beginning TV line to write text (default=7 on a 480x640 frame; def=computed on other sizes)

ele beginning TV element to write text (default=320 on a 480x640 frame; default=computed on other sizes)

color graphics color level of text (default=3)

size pixel size of text (default=11 on a 480x640 frame; default=computed on other sizes)

"text annotation text (default=*memo field* from IMGLIST FORM=ALL; if the image also has an *INFO AUX* block, like NEXRAD precip images, it is written below the *memo field* text)

COLor= graphics color level for bar labels (default=3)

GRA= graphics frame to display labels (default=current if running McIDAS-X with independent graphics; otherwise, number specified in the *frame* parameter)

LINt= label interval; must be a value greater than zero (default=computed)

LSIze= height of labels (default=computed)

ORIent= **VER** bar is displayed vertically (default)

HOR bar is displayed horizontally

POSition= *line ele short long*

user-specified location and size (default)

line starting TV line (default=computed)

ele starting TV element (default=computed)

short length in pixels of the bar's short axis (default=computed)

long length in pixels of the bar's long axis (default=computed)

CUR draws a bar the size the cursor at its current location

RANge= *min max* range of values to display (default=computed)

SU= stretch table name to apply to the grayscale bar; see the Remarks (default=table specified in the IMGDISP command)

Remarks

The BAR command displays a grayscale bar and labels. The bar is written to the image so an enhancement created with commands EU or EB appears on the bar. Because the labels are written as graphics, the EG command erases the labels but not the bar. The *frame* parameter specifies the frame to write the bar, and the GRA keyword specifies the frame to write the labels.

The POSITION keyword determines the location of the bar. If you do not use this keyword, the bar is drawn in a vertical orientation along the right side of the frame. If you use the POS=*line ele short long* format, *line* and *ele* define the TV line and element of the upper-left corner of the bar, and *short* and *long* define the dimensions of the bar. If you use ORIENT=VER, *short* is the horizontal length and *long* is the vertical length. If you use ORIENT=HOR, *short* is the vertical length and *long* is the horizontal length.

If the image was displayed with a stretch table (for example, using IMGDISP with the SU keyword), BAR's SU and RANGE keywords are ignored. If the image has an auxiliary block (for example, NEXRAD Level III radar images), the RANGE keyword is ignored and the LINT keyword is valid only if the image was displayed with a stretch table.

The following three rules determine the units of the labels.

- If the image was displayed with a stretch table, the label units are the same as the calibration units of the stretch table.
- If the image has an auxiliary block and was not displayed with a stretch table, the label units are the same as the units in the auxiliary block.
- If the image was not displayed with a stretch table and does not have an auxiliary block, the units are the same as the calibration units of the stretch table specified in the BAR command's SU keyword. If SU is not specified, the units are brightness values.

Examples

BAR

This entry displays a vertical grayscale bar and graphics labels along the right side of the current frame.

BAR ORIENT=HOR LINT=15 LSIZE=6 COLOR=5

This entry displays a horizontal grayscale bar and graphics labels along the bottom of the current frame. The labels are drawn every 15 brightness levels along the bar in graphics color level 5 with a character height of six pixels.

BAR POS=CUR

This entry displays a grayscale bar that fills the cursor at its current location. The dimensions of the cursor determine the orientation of the bar and labels. If the cursor is square or is longer in the vertical direction than in the horizontal direction, the bar and labels are oriented vertically. If the cursor is longer in the horizontal direction than in the vertical direction, the bar and labels are oriented horizontally.

BAR POS=20 20 10 600 ORIENT=HOR SU=QVISRB RANGE=180 330 LINT=15

This entry displays a horizontal grayscale bar and graphics labels along the top of the current frame. The bar, which displays only values between 180 and 330, is 10 pixels high, 600 pixels long, and starts at TV line 20, element 20. The labels appear every 15 values along the bar in units matching the calibration units of the stretch table QVISRB.

BAR ANNOT=YES L 445 5 2 8

This entry displays a vertical grayscale bar and graphics in the default location (along the right side of the frame) and displays the *memo field* of the currently displayed image in graphics color level 2 with a character height of eight pixels beginning at TV line 445, element 5. If the image also has an *INFO AUX* block (like NEXRAD precipitation images), it is displayed below the *memo field*. The *memo field* can be listed by specifying FORM=ALL with the IMGLIST command.

BATCH

Runs commands serially from a text file.

Formats

BATCH *file [keywords]*
BATCH *par1 par2 . . parn file [keywords]*

Parameters

file	name of the batch file containing the commands; see the Remarks
par1 . . n	value to substitute wherever the corresponding *%n* notation (%1, %2, etc.) appears in the batch file

---

Keywords

CONtinue=	**NO**	stops the batch file if any command stops abnormally (default)
	YES	ignores any abnormal terminations and continues with subsequent commands
DEST=	name of the text file to write commands into instead of running them; the file contains the list of expanded commands and can be used by other programs	
	C	directs commands to the text frame for viewing only

Remarks

The batch file containing the commands must be in text format, created with a text editor. Commands in the batch file may be up to 512 characters long. Commands are performed serially; each must finish before the next one begins.

Normal string table substitution may be used within each command and will take place as the command is run. Use the TERM command to change the state of the single letter command toggles.

To add a remark in the batch file, specify REM or REM? followed by one or more spaces and the remark text. The letters (REM) must be uppercase and must begin in column 1. Remarks prefaced by REM? are echoed to the text frame.

Pseudocommands

BATCH also allows for some rudimentary flow control by permitting the following eight pseudocommands in the batch file. The term *value* in the formats below may be up to 512 characters, and is one of the following:

- a character string (for example, 1234ABC)
- a McIDAS string, preceded by a pound sign (for example, #STATION)
- a variable, enclosed in percent symbols (for example, %STATION%)
- a BATCH command parameter number, preceded by a percent symbol (for example, %1)

SET *variable=value*

This pseudocommand replaces the value of the *variable*, which is only available while the BATCH command is running. Reference the variable elsewhere in the batch file by using *%variable%* or *#variable*. If *variable* is not defined with SET, it is obtained from the McIDAS string table. You can define and use up to 30 variables, which are simply names storing strings of characters. Variable names may be up to eight alphanumeric characters.

SETSTRING *string=value*

This pseudocommand replaces the value of *string* in the McIDAS string table.

IF *value1==value2 command*

This pseudocommand runs the specified command if the two values are identical.

IF NOT *value1***==***value2 command*

This pseudocommand runs the specified command if the two values are NOT identical.

IF NOT ERRORCODE *command*

This pseudocommand runs the specified command if the last McIDAS command completed successfully. (That is, it returned a zero exit status.)

IF ERRORCODE *command*

This pseudocommand runs the specified command if the last McIDAS command had an error. (That is, returned a non-zero exit status.)

GOTO *label*

This pseudocommand causes the BATCH command to jump forward to the next line tagged with *:label*. See the *:label* pseudocommand below.

:label

This pseudocommand tags a line with a label so that it can be found with the GOTO pseudocommand. The *label* is a character string of up to 12 characters (for example, DONE!). You can add a comment after the label as long as one or more spaces separate the label and comment.

Examples

BATCH KMIA 0 6 LIST.BAT

If your BATCH file, **LIST.BAT**, contains the line below, this entry runs the McIDAS command **SFCLIST KMIA TIME=0 6**.

```
SFCLIST %1 TIME=%2 %3
```

BATCH KTUL DISPLAY.BAT CONTINUE=YES

If your BATCH file, **DISPLAY.BAT**, contains the lines below, this entry sets the value of the variable ID to KTUL for use in the IMGDISP and PC commands. If station KTUL is in the image, it is displayed with a high resolution map, then the cursor is moved to the station location. If the image doesn't exist or station KTUL isn't in it, the message **** OOPS! Station must be invalid or not on image **** appears on text frame.

```
SET ID=%1 IMGDISP A.100 STA=%ID% REFRESH='MAP VH' IF NOT ERRORCODE GOTO CURSOR REM?
*** OOPS! Station must be invalid or not on image *** GOTO DONE :CURSOR REM? ***
Another successful image load! *** PC L %ID% :DONE
```

BRIEF

Opens a briefing window.

Format

BRIEF *window*

Parameter

window briefing window number; the range is 0-9 (default=0)

Remarks

Use briefing windows to display loops 1 through 9 saved with the DEFLOOP command. Do not create briefing windows with the same window number.

Use the following commands and procedures to control briefing windows and loops.

To:	**Do This:**
Advance one frame in the loop	Press **A**
Back up one frame in the loop	Press **B**
Close the briefing window	Choose Close from the Window menu or let the window automatically close when you exit McIDAS
Display a loop in a briefing window	Use the MOVIE command with the WIN keyword, or make the briefing window active and press the keyboard key that corresponds to the loop number
Resize the briefing window	Click and drag the window border
Return the briefing window to its original size	Press **R**
Start or stop looping	Press **L**

BRIEF makes the briefing window the same size as frame 1. When you display a loop in the briefing window that was saved on a frame of a different size, the loop's frames are rescaled to fit the briefing window. For example, if the loop was saved on 700 x 900 frames and the briefing window is 480 x 640, the loop's frames are rescaled to 480 x 640 to display the entire frame.

When you resize briefing windows using the window border, the frames displayed in the briefing window are automatically rescaled to fit the window. If you decrease the size of the briefing window, the graphics on the displayed frames will deteriorate.

The delay between frame advances in a briefing window is 250 milliseconds. Thus, the fastest loop speed you can attain in a briefing window is four frames per second, even if you specified a much faster loop speed when you saved the loop with DEFLOOP.

Examples

BRIEF

This entry creates briefing window 0. Use command DEFLOOP and/or MOVIE to display loops in the window.

BRIEF 3

This entry opens briefing window 3. Use command DEFLOOP and/or MOVIE to display loops in the window.

C

Lists a frame's directory.

Remarks

To list the current image frame's directory,

Press: **Alt C**

or

Type: **C**

Press: **Enter**

The output looks like this:

```
Frame SS Description Band Date Time Line Elem Data Source ----- -- ----------- ----
----- -------- ---- ----- ----------- 1 70 G-8 IMG 1 96171 16:02:00 3344 11935
A/ALL.4
```

Each field is defined below.

Field	**Definition**
Frame	frame number
SS	sensor source number
Description	sensor source description
Band	image band number
Date	image's Julian day, or date displayed if a graphic
Time	image's time, or the time displayed if a graphic
Line	line number, in image coordinates, of the upper-left corner of the frame
Elem	element number, in image coordinates, of the upper-left corner of the frame
Data Source	ADDE dataset name and position number if displayed with the IMGDISP command

---

CASEMODE

Sets the case mode for entering McIDAS-X commands.

Formats

CASEMODE LIST
CASEMODE SWAP
CASEMODE NONE

Parameters

LIST	lists the current case-swap mode, either ON or OFF (default)
SWAP	sets the automatic case-swap mode to ON so text is automatically entered in uppercase; this is the default mode when starting McIDAS-X; see the Remarks
NONE	sets the automatic case-swap mode to OFF to make the Caps Lock indicator light correct

---

Remarks

The CASEMODE command automatically changes entered text to upper or lowercase.

By default, McIDAS-X is started in the SWAP mode since most McIDAS-X command names, parameters and keywords are typed in uppercase. Using the SWAP mode, you don't have to press the Caps Lock key when you first start McIDAS-X to switch to uppercase for typing McIDAS-X commands. However, it also makes the Caps Lock indicator light incorrect. If you enter the command CASEMODE NONE, text is automatically entered in lowercase and the Caps Lock indicator light is correct.

You can use the **Shift** or **Caps Lock** key to switch from lowercase to uppercase and vice versa without using the CASEMODE command.

Examples

CASEMODE

This entry lists the current case-swapping mode.

CCODE

Lists country codes.

Format

CCODE *country*

Parameter

country country to list (default=all)

Remarks

If *country* is two characters, CCODE looks for the one country whose code matches it. If one character or more than two characters are specified, all string matches are found.

Examples

CCODE

This entry lists all country codes.

CCODE GB

This entry lists the country with GB as its country code.

CCODE FRANCE

This entry lists the country code for France.

CCODE ISLAND

This entry lists all countries that have the word ISLAND in them.

CCODE Z

This entry lists all country codes and countries containing the letter Z.

CHILL

Calculates the wind chill index.

Format

CHILL *temp t-units speed s-units*

Parameters

temp temperature for calculating the wind chill index

t-units units of the *temp* value; the output will also be in these units

F	Fahrenheit
C	Celsius
K	Kelvin

speed wind speed for calculating the wind chill index

s-units units of the specified wind speed

KTS	knots
KPH	kilometers per hour
MPH	miles per hour
MPS	meters per second
FPS	feet per second

Keyword

CHIlltype=	**CURRENT**	list wind chill value computed with the formula implemented by the National Weather Service in November 2001 (default)
	OLD	list wind chill value computed with the pre-2001 formula
	ALL	list wind chill value for both CURRENT and OLD formulas

Examples

CHILL -15 C 17 KTS

This entry lists the wind chill index for a temperature of -15° C and a wind speed of 17 knots.

CHILL 12 F 40 MPH

This entry lists the wind chill index for a temperature of 12° F and a wind speed of 40 miles per hour.

CLEAR

Clears output from text frames.

Format

CLEAR *bframe* *eframe*

Parameters

bframe	beginning text frame to clear; the range is 0 to 9 (default=current)
eframe	ending text frame to clear; the range is 0 to 9 (default=*bframe*)

Examples

CLEAR

This entry clears all text from the current text frame.

CLEAR 4

This entry clears text frame 4.

CLEAR 0 3

This entry clears text frames 0, 1, 2, and 3.

CM

Circulates the graphics color table.

Format

CM *blevel elevel [keywords]*

Parameters

blevel beginning graphics color level to loop

elevel ending graphics color level to loop

Keywords

GO=	**YES**	starts looping immediately
	NO	does not loop immediately (default)
INT=	interval between steps, in milliseconds; the range is 5 to 5000 (default=100)	
TIMeout=	number of steps to complete before stopping (default=500)	

Remarks

CM circulates the graphics colors through the specified graphics levels. Press the middle mouse button to begin, the right button to pause, and both buttons to stop the program. CM remains active until you press both mouse buttons or the program times out.

You can animate wind flow using the CM command to process the streamline output from the GRDDISP or the PTCON commands. You must use the PARAM=MOVIE keyword with these commands.

Examples

CM 1 3 INT=500

When you press the middle mouse button, the graphics will loop between color levels 1 and 3, pausing 500 msec (0.5 sec) between steps.

Time (sec)		0	0.5	1	1.5. . .
Level	1	red	blue	green	red . . .
	2	green	red	blue	green . . .
	3	blue	green	red	blue . . .

CM 2 4

This entry loops graphics color levels 2 to 4.

COLORS

Lists the colors available with the global keyword TCOL.

Format

COLORS *bcolor ecolor*

Parameters

bcolor	beginning color to list; the range is 0 to 127 (default=0)
ecolor	ending color to list; the range is 0 to 127 (default=*bcolor*)

Remarks

If you don't specify *bcolor* and *ecolor*, all 128 colors are listed. The 128 colors consist of eight sets of background colors, each with the same 16 text colors to choose from. For example, colors 0 through 15 have a black background, while colors 16 through 31 have a blue background.

Each color value is listed on a separate line in its defined text and background colors.

Use the global keyword TCOL to specify the desired text color.

See Also

See *Appendix A, Global Keywords* for more information about the global keyword TCOL.

Examples

COLORS

This entry lists all 128 colors.

COLORS 64 79

This entry lists colors 64 through 79, which have a red background.

COMBINE

Combines three McIDAS frames into a single 24-bit image.

Format

COMBINE *window f1 f2 f3 [keywords]*

Parameters

window	number of the combined window to create and display the 24-bit output image; the range is 0-9
f1, f2, f3	three image frame numbers to combine into the 24-bit output image; see the Remarks

---

Keywords

VIEw=	**YES**	opens the specified combined window, showing the 24-bit output image; use VIEW=YES only if the display has a 24-bit graphics card and monitor (default)
	NO	does not open the specified combined window, but does store the 24-bit output image in memory so it can be saved with the FRMSAVE command; see the Remarks

---

Remarks

COMBINE combines three McIDAS frames into a single 24-bit image. The output image is composed of the red component of the first frame (*f1)* and its associated graphics, the green component of the second frame (*f2*) and its associated graphics, and the blue component of the third frame (*f3*) and its associated graphics. Since only a single color component is used from each of the three frames, you should carefully select which colors you use in image enhancements and graphics on those frames. For example, if a magenta map is placed on the second frame (*f2*), it will be black in the COMBINE output image because magenta has no green component (it consists entirely of red and blue).

If you're on a 24-bit display, you can use VIEW=YES (the default) to create a combined window that displays the image. The new window has **combine window *n*** on its title bar, where *n* is the number specified in the *window* parameter. If you're not on a 24-bit display or you are running a script, you can use VIEW=NO to create and store the 24-bit image in memory. Then, use the FRMSAVE command to save the image in a JPEG or PPM format file.

If you use VIEW=YES (the default), the COMBINE window will be the size of frame 1, regardless of the size of the three frames specified in parameters *f1*, *f2* and *f3*. If you specify VIEW=NO, the COMBINE window will be the size of the largest frame among the three specified in parameters *f1*, *f2* and *f3*.

Image frames and graphics frames are "associated" once they have been viewed together. Sessions without independent graphics automatically associate each image frame with the graphics frame of the same number once the frame has been displayed. However, sessions with independent graphics have separate control (and display) of image and graphics frames so image frames are not necessarily associated with the graphics frames of the same number. For example, if you loop through image frames 1-3 while displaying graphics frame 1, all three of the image frames are associated with graphics frame 1. This is relevant because the 24-bit image that COMBINE outputs includes the graphics frames associated with the image frames specified in *f1*, *f2*, and *f3*.

Examples

COMBINE 0 1 2 3

This entry combines the red, green and blue components of the data displayed in image frames 1, 2, and 3 and their associated graphics frames into a 24-bit image. The 24-bit image is displayed in a separate window and labeled **combine window 0** on the display.

COMBINE 9 14 12 13 VIEW=NO

This entry combines the red, green and blue components of the data displayed in image frames 14, 12, and 13 and their associated graphics frames into a 24-bit image. The 24-bit image is not displayed, but it can be saved to a JPEG or PPM format file with the FRMSAVE command.

CU

Lists a quantity in up to eight different units.

Formats

CU *value unit nunits1 . . nunits8*
CU LIST

Parameters

value	amount of the unit; must be a real number
unit	unit to convert
nunits1 . . nunits8	new units converted to; specify up to eight units
LIST	lists a table of units and abbreviations (default)

Remarks

To square or cube the units, append 2 or 3 to the units. For example, IN2 means square inches and IN3 means cubic inches.

The tables below list the valid units and abbreviations.

Distance		**Speed**		**Temperature**	
CM	centimeters	FPS	feet per second	C	Celsius
DEGL	degrees of latitude	KTS	knots	F	Fahrenheit
FT	feet	KPH	kilometers per hour	K	Kelvin
IN	inches	MPH	miles per hour		
KM	kilometers	MPS	meters per second		
M	meters				
MI	miles				
MM	millimeters				
NMI	nautical miles				
YD	yards				

Weight		**Pressure**		**Time**	
G	grams	INHG	inches of mercury	DAY	days
KG	kilograms	MB	millibars	HR	hours
LB	pounds	HPA	hectopascals	MIN	minutes
OZ	ounces	PA	pascals	SEC	seconds
TON	tons			YR	years

Examples

CU

This entry lists the table of valid units and their abbreviations.

CU 1 MI IN FT YD NMI

This entry converts 1 mile to inches, feet, yards and nautical miles.

CU 30.37 INHG MB

This entry converts 30.37 inches of mercury to millibars.

CUR

Defines cursor size, type and color.

Formats

CUR *height width type color*
CUR *height width type blue green red*
CUR *option*

Parameters

height	cursor height in pixels; even size is rounded up to odd size; if entered as X, the size does not change (default=31)	
width	cursor width in pixels; even size is rounded up to odd size (default=*height*)	
type	**BOX**	box cursor
	XHAIR	crosshair cursor (default)
	XBOX	crosshair inside box cursor
	SOLID	solid box cursor
color	predefined color name; see the Remarks (default=red)	
blue	blue color intensity; the range is 0 to 255 (default=0)	
green	green color intensity; the range is 0 to 255 (default=0)	
red	red color intensity; the range is 0 to 255 (default=255)	
option	**ON**	makes the cursor visible (default)
	OFF	makes the cursor invisible

Remarks

Command GU COLORS lists the valid predefined cursor colors.

The color default only applies when entering CUR with no positional parameters. For all other command formats, the default is the current *color*.

Examples

CUR

This entry displays a 31 x 31 red crosshair cursor.

CUR 21

This entry makes the cursor 21 x 21 pixels.

CUR 20 40 BOX GREEN

This entry changes the cursor to a green box, 21 x 41 pixels.

CUR 40 30 XBOX BLUE

This entry makes a 41 x 31 pixel, blue, crosshair inside box cursor.

CUR X 53 XHAIR

This entry makes a crosshair cursor, 53 pixels wide; the height and color do not change.

CUR 32 25 BOX 200 80 120

This entry makes a 33 x 25 box cursor colored by the intensities for blue, green and red as 200, 80 and 120, respectively.

CUR OFF

This entry turns off the cursor making it invisible.

CW

Fills or erases the graphics in the cursor.

Format

CW *level frame height width*

Parameters

level	graphics color level; the range is zero to the maximum available for the display type (default=0 to erase all graphics within the cursor)
frame	graphics frame number (default=current)
height	height of the area to erase, in pixels (default=cursor height)
width	width of the area to erase, in pixels (default=cursor width)

See Also

Use command EG to erase the entire graphics frame.

Examples

CW

This entry erases the graphics within the cursor.

CW 1 3

This entry fills the cursor with graphics color level 1 on graphics frame 3.

CW 4 2 50 30

This entry fills a 50 x 30 box centered at the cursor with graphics color level 4 on graphics frame 2.

D

Lists the digital values at the cursor center.

Remarks

To list the digital values and other appropriate units of the image pixel under the cursor's center,

Press: **Alt D**

or

Type: **D**

Press: **Enter**

The D command lists the dataset name, file coordinates, image coordinates, raw digital values, brightness values, radiance, and blackbody temperatures, or other units appropriate to the data source.

The nominal image coordinates correspond to the cursor location. They may not be the original image coordinates of the displayed data value. Image coordinate values may vary if the image file was blown down from the original image and then redisplayed with a magnification factor of two or more.

Any 2-byte IR or dwell-sound GOES data converted to 1-byte data using the IMGCOPY command, assigns missing temperature or brightness values of zero. Brightness values of 255 also indicate missing values such as data off the earth edge or missing lines.

See Also

Use the IMGPROBE command to list the data for the area inside the cursor.

DATALOC

Manages and lists the client routing table.

Formats

DATALOC ADD *group address*
DATALOC DEL *group*
DATALOC LIST *group*
DATALOC SSHADD *address user file*
DATALOC SSHDEL *address*
DATALOC SSHLIST *address*
DATALOC HOST

Parameters

ADD	adds an entry to the client routing table; the entry maps a group name to a server's IP address
DEL	deletes an entry from the client routing table
LIST	lists the client routing table (default)
SSHADD	adds an entry to allow tunneling for ADDE requests
SSHDEL	deletes entry that allowed tunneling for ADDE requests
SSHLIST	lists the user name and file for tunneling address
HOST	refreshes the name server lookup for hosts; must be run if a server's numeric IP address is changed
group	group name (no default for ADD and DEL; default=all groups for LIST)
address	IP address of the server; specify LOCAL-DATA for the local server
user	user name on remote host used for tunneling
file	file name containing ssh keys for tunneling to a remote host

---

Remarks

DATALOC manages the client routing table, which links ADDE groups to IP addresses of servers. When you enter an ADDE command, the client routing table is scanned to locate the server that will receive the data request. When the server receives the data request, it scans its own mapping table to determine which data the client is requesting. If an ADDE command specifies a group name that is not in the client routing table, the data request is sent to the local server.

You cannot assign the same group name to more than one server because the ADD option overwrites an existing group name without warning. You may, however, assign more than one group name to a server.

The SSHADD, SSHDEL and SSHLIST options should be used only if you access datasets on a remote server (specified in the *address* parameter) that uses SSH tunneling. Both the client and remote server must have an ssh key file set up for automatic login (with saved password) on the remote server, and its ssh daemon must be listening for requests.

Examples

DATALOC

This entry lists the client routing table.

DATALOC ADD MYDATA LOCAL-DATA

This entry modifies the client routing table so that commands requesting data from a dataset in the group MYDATA are routed to the local server.

DATALOC ADD RTSERV WEATHER.SSEC.WISC.EDU

This entry modifies the client routing table so that commands requesting data from a dataset in the group RTSERV are routed to the server at the IP address WEATHER.SSEC.WISC.EDU.

DATALOC ADD RADAR 123.456.789.012

This entry modifies the client routing table so that commands requesting data from a dataset in the group RADAR are routed to the server at the IP address 123.456.789.012.

DATALOC LIST RADAR

This entry lists the client routing table entry for the group RADAR.

DATALOC DEL RADAR

This entry deletes the client routing table entry for the group RADAR.

DATELIST

Lists the specified date in several formats.

Format

DATELIST *date offset* **FORM=**

Parameters

date date to list; valid formats are YYYYDDD, YYYY/MM/DD and DD/MON/YYYY where YYYY, DDD, MM and DD are integers and MON is three or more letters, for example, JAN or JANUARY (default=current)

offset number of days to offset the date; a positive integer increments the specified date, while a negative integer decrements the date (default=0)

Keyword

FORm= '*text*' customize output format using text and/or predefined place holders; see the Remarks for a list of place holders default= '(YYYY)(DDD) = (DD) (SMON) (YYYY) = (YYYY)/(MM)/(DD)'

Remarks

The minimum date that can be listed is 1 January 1970. The maximum date is system dependent.

The FORM keyword lets you customize the output format. You can mix regular text with the predefined place holders listed below.

Place Holder	Description	Example
(DOW)	day of week	Thursday
(SDOW)	3-character day of week	Thu
(DD)	2-digit day of month	31
(DDD)	3-digit Julian day	365
(MM)	2-digit month of year	12
(MON)	full month name	December
(SMON)	3-character month name	Dec
(YY)	2-digit year	98
(YYYY)	4-digit year	1998

Examples

DATELIST

This entry lists the current day in the formats YYYYDDD, DD MON YYYY and YYYY/MM/DD.

DATELIST X 1

This entry lists tomorrow's date in the formats YYYYDDD, DD MON YYYY and YYYY/MM/DD.

DATELIST 97264

This entry lists day 264 of 1997 in the formats shown below.

```
1997264 = 21 Sep 1997 = 1997/09/21
```

DATELIST 2012/05/23

This entry lists 23 May 2012 in the formats shown below.

```
2012144 = 23 May 2012 = 2012/05/23
```

DATELIST 1/MAR/2004 -1

This entry lists the day before 1 March 2004 in the formats shown below.

```
2004060 = 29 Feb 2004 = 2004/02/29
```

DATELIST 1998365 FORM='The date is: (DOW) (SDOW) (DD) (DDD) (MM) (MON) (SMON) (YY) (YYYY)'

This entry lists 31 December 1998 in the formats shown below.

```
The date is: Thursday Thu 31 365 12 December Dec 98 1998
```

DEFLOOP

Lists, saves and restores loops.

Formats

DEFLOOP LIST *bloop eloop*
DEFLOOP SAVE *loop "label*
DEFLOOP REST *loop*

Parameters

LIST	lists the directory of saved loops
SAVE	saves the current loop
REST	restores a previously saved loop
bloop	beginning loop to list (default=all defined loops)
eloop	ending loop to list (default=*bloop*)
loop	loop number to save or restore; the range is 1 to 99
"label	label to save with the loop state; 48 characters maximum including blanks

---

Remarks

The DEFLOOP SAVE option saves the state of the workstation display by storing the display parameters in User Common. The display parameters include the loop bounds, dwell rates, image and graphics on/off state, cursor state, nortel key (N), and loop state. The REST option restores the loop with its saved settings to the Image Window.

You can use either DEFLOOP or the F Key Menu pseudocommand *S to save the state of your workstation display. They use the same filing system to save loops.

You can restore loops 1 through 9 to a briefing window created with the BRIEF command. All the loop settings except the cursor position, image on/off, and graphic on/off are restored. To restore the loop to a briefing window, use the MOVIE command with keyword WIN or make the briefing window active and press the keyboard key corresponding to the loop number between 1 and 9. Press **L** to manually start or stop looping, **A** to advance a frame, and **B** to back up a frame in the loop.

Always add new loops consecutively to save disk space since you will need 16 Kbytes on disk to store the context for a loop.

Examples

DEFLOOP SAVE 3 "East US visible hourly

This entry saves the current loop as loop number 3 with the label *East US visible hourly*.

DEFLOOP REST 27

This entry restores loop 27 and displays its label if one was specified. You can use the LS, DR and F commands to interrogate the state.

DEFLOOP LIST

This entry lists the loop numbers and labels for loops saved with a label.

DIST

Finds the distance on a navigated frame.

Formats

DIST CIR *angle inc distance [keywords]*
DIST ROT *angle inc distance [keywords]*
DIST STE *angle din distance [keywords]*
DIST E *lat lon [keywords]*
DIST I *line ele [keywords]*
DIST T *raster pictel [keywords]*

Parameters

CIR	draws a circle around the initial cursor location beginning at a specified distance and angle
ROT	draws all or part of a circle around the initial cursor location beginning at a specified distance and angle
STE	steps the cursor away from the initial cursor location at a specified distance and angle
E	steps the cursor away from the initial cursor location by the specified latitude and longitude increments
I	steps the cursor away from the initial cursor location by the specified line and element increments
T	steps the cursor away from the initial cursor location by the specified raster and pictel increments
angle	moves the cursor away from the original location by this angle, in degrees (default=0)
inc	increment (in degrees) to shift the cursor along its circular path
distance	distance to move the cursor from its original location
din	increment to step the cursor along its straight-line path
lat	degrees of latitude to move the cursor; northward is positive
lon	degrees of longitude to move the cursor; westward is positive
line	number of image lines to move the cursor; downward is positive
ele	number of image elements to move the cursor; rightward is positive
raster	number of TV lines (rasters) to move the cursor; downward is positive
pictel	number of TV elements (pictels) to move the cursor; rightward is positive

---

Keywords

Color=	graphics color level of the characters (default=2)	
SIze=	height of the plotted characters, in pixels (default=5)	
SYm=	**.**	a period marks off the distances (default)
	+	a plus sign marks off the distances
	*	an asterisk marks off the distances

	o	a lowercase letter O marks off the distances
Unit=	**KM**	measures distance in kilometers (default)
	MI	measures distance in statute miles
	NMI	measures distance in nautical miles

Remarks

Command DIST will not run unless the satellite image is navigated or a MAP command is run.

Each cursor location marked on the graphics frame is indicated by a period (default) or a symbol chosen with keyword SYM. A symbol is plotted on the graphics frame at the new cursor location each time you press the middle mouse button.

DIST runs interactively when specified with no parameters, or with the ROT, STE, E, I or T parameter.

Examples

Position the cursor at the point on a navigated frame that is the starting point of the calculation. Then,

Type: **DIST**

The program replies:

```
Press MIDDLE mouse button or ALT+G to measure point. Press RIGHT button to reset
the starting point. Press BOTH mouse buttons or ALT+Q to exit DIST.
```

Move the cursor to the starting point from which you want to measure one or more other points, then press the middle mouse button or **Alt G**. Next move the cursor to the first point you want to measure and press the middle mouse button or **Alt G**. Move the cursor to all subsequent points you want to measure, pressing the middle mouse button or **Alt G** at each point. Press the middle and right mouse buttons or **Alt Q** after measuring all the points.

Each time you measure a point, the SYM keyword symbol and the distance to the starting point are plotted on the graphics frame. Also, the latitude and longitude of the point and its angle and distance to the starting and previous points are listed on the text frame.

DIST CIR 45 30 100 SYM=*

This entry draws a circle of asterisks on the graphics frame 100 km away from the original cursor location beginning at angle 45. The circle contains 12 asterisks since the increment is 30°.

DIST ROT 90 15 200 UNIT=NMI SYM=* SIZE=20 COLOR=4

This entry draws an arc or circle of asterisks on the graphics frame 200 nautical miles away from the original cursor location, beginning at an angle of 90°. The asterisks are 20 pixels high in graphics color level 4.

DMAP

Lists information about files.

Formats

DMAP *string [keyword]*
DMAP *bletter eletter [keyword]*

Parameters

string	character string in the file name; you can include the wild card characters asterisk (*) and question mark (?) (default=lists all files)
bletter	beginning letter of the file name
eletter	end of the range of letters beginning the file name (default=*bletter*)

Keyword

Form=	**ALL**	lists all versions of the specified file names; the unreachable versions are flagged by a pound sign (#)

Remarks

DMAP lists file information in the ASCII collating sequence. The permissions, size, date, file name and directory are given for each entry in the list.

If you specify string without a wild card character, DMAP lists all files that begin with the specified character string. If you specify *bletter* and *eletter*, all files with names that begin with a letter in the range *bletter-eletter* are listed.

The output's PERM field indicates the user's permissions. A *d* in the first field position indicates the entry is a directory; a hyphen (-) indicates a file. The letters *r*, *w* or *x* in the next three positions indicate read, write or execute permissions. A hyphen (-) in any of the positions indicates that the user does not have that permission.

The FORM=ALL list also includes unreachable files in the REDIRECT entries and the **MCPATH** directories. Unreachable files are flagged by a pound sign (#) before the fully qualified path name in the PATHNAME field. If the file is in a REDIRECT directory, the FROM field contains an *R* followed by the REDIRECT entry. If the file is in a **MCPATH** directory, the FROM field is an *M*.

Examples

DMAP

This entry lists information about all files.

DMAP M

This entry lists information about all files with names beginning with M.

DMAP MDX

This entry lists information about all files with names that begin with the character string MDX.

DMAP ?MDX

This entry lists information about all files with names beginning with a single character followed by the character string MDX.

DMAP *.PIX

This entry lists information about all files with the .PIX extension.

DMAP F J

This entry lists information about files with names beginning with F, G, H, I and J.

DOQTL

Selectively deletes MD files.

Format

DOQTL *bfile efile cutoff*

Parameters

bfile	beginning MD file in the range (no default)
efile	ending MD file in the range (no default)
cutoff	maximum number of MD files kept on the system, in days (no default)

Remarks

Command DOQTL allows you to delete unwanted MD files on the workstation according to the specified cutoff. If *cutoff* =4, only the MD files for the last four days are retained on the workstation; the rest are deleted. This assumes that the MD files 1 to 10 are for ISFC schema data, MD files 11 to 20 are for IRAB schema data, etc.

Example

DOQTL 1 30 2

If the current day is 91223, this entry saves MD files 2, 3, 12, 13, 22 and 23, and deletes the rest.

DOSTOLW

Converts ASCII text format files to LW format text files.

Format

DOSTOLW *sfile* *dfile* **R**

Parameters

sfile	source ASCII text file name and extension
dfile	destination LW file name; it must be different than *sfile*
R	replaces the destination file if it already exists

Remarks

The *sfile* must be an ASCII text file with a new line character at the end of each line.

A line in LW file format is an 80-character card image, blank padded as needed, with no carriage returns or linefeeds.

Examples

DOSTOLW SFCPLT.RUN SFCPLT

This entry converts the ASCII text file named **SFCPLT.RUN** to an 80-character card image format LW file named SFCPLT.

DR

Defines the dwell rates for frame looping.

Formats

DR *rate1 rate2 . . raten*
DR *repeat*rate repeat*rate . . .*
DR INI
DR AUTO

Parameters

rate1 . . n	dwell rate for each frame (no default)
repeat	number of frames the rate applies to (no default)
INI	initializes the dwell rates to nine units for the first frame and six units for all subsequent frames
AUTO	assigns the longest dwell to the latest image in the current loop.

Remarks

The rate is in units of 1/15 second. The last frame assigned a rate determines the rate for all subsequent frames in that loop. See the examples.

Set the dwell rates to compensate for irregularly spaced image times. Graphics frames in loops have dwell rates identical to image frames.

You can mix the command formats. For example, **DR 3 3 5*8 3** is the same as **DR 3 3 8 8 8 8 8 3**.

Examples

DR

This entry lists the current dwell rates in the loop.

DR 7

This entry sets all frames to dwell rate 7.

DR 12 6 6 12

This entry sets dwell rates to six for the second and third frames, and 12 for the first, fourth and all subsequent frames of the loop.

DR AUTO

This entry rotates the dwell rates so the longest dwell is assigned to the most recent image in the loop. If the dwell rates of all the frames are equal, this entry has no effect.

DSINFO

Lists datasets on local and remote servers.

Format

DSINFO *type group*

Parameters

type	data type; valid types are GRID, IMAGE, NAV, POINT, TEXT or ALL (default=ALL)
group	group name (default=all groups)

Remarks

DSINFO scans the client routing table to find the server with the data type and group name specified in the command. It then lists information about the server's datasets with that data type and group name.

Use the DATALOC LIST command to list the groups currently defined in the client routing table.

If the ADDE remote servers you access require a user name and project number for accounting, you must logon to your workstation (using the LOGON command) before requesting data with DSINFO or other ADDE commands.

Examples

DSINFO

This entry lists all datasets in the groups defined in the client routing table.

DSINFO ALL RTSERV

This entry lists all datasets in the group RTSERV.

DSINFO IMAGE RTSERV

This entry lists all datasets of data type IMAGE in the group RTSERV.

DSINFO IMAGE

This entry lists all datasets of data type IMAGE in the groups defined in the client routing table.

DSSERVE

Manages and lists dataset names on the local server.

Formats

DSSERVE ADD *dataset format bfile efile [keywords] "comment*
DSSERVE DEL *dataset*
DSSERVE LIST *group format*

Parameters

ADD	adds a new ADDE dataset name to the local server
DEL	deletes an ADDE dataset name from the local server
LIST	lists the ADDE dataset names on the local server (default)
dataset	ADDE dataset name to add or delete; must be specified in the group/descriptor format; see the Remarks for restrictions
format	file format; valid formats include AIRS, AREA, COMS, GEOT, GRIB, GRID, LV1B, MD, MOD4, MOD8, MODR, MODS, MODX, MSGT, MTST, NCDF, SYSN, and TEXT (no default for ADD; default=all formats for LIST)
bfile	beginning file number assigned to the ADDE dataset; *bfile* and *efile* are required for AREA, GRID, MD, NCDF and SYSN formats
efile	ending file number assigned to the ADDE dataset
"*comment*	description of the ADDE dataset; up to 80 characters; the description cannot include an equal sign (=), apostrophe ('), or comma (,); the text entered in this field is visible to the client when they list the dataset information with the DSINFO command
group	group name (default=all groups)

Keywords

DIRfile=	'*mask*'	directory and/or file mask to locate files for the non-McIDAS format servers; can also be used for the McIDAS area, grid and MD file servers if the files are not in the **MCPATH** directories, REDIRECT directory, or have names other than **AREA***, **GRID*** and **MDXX***; do not specify *bfile* and *efile* when using this keyword; see the Remarks
INFo=	'*info*'	additional information for the server about the data; required by some servers but not needed or used with McIDAS file formats AREA, GRID, MD, and SYSN; 120 characters maximum; the beginning and ending single quotes are mandatory; see the Remarks
TYPe=		data type of the dataset created with the ADD option; valid types are IMAGE, GRID, NAV, POINT and TEXT; must be specified for all non-McIDAS file formats except TEXT (default=IMAGE for AREA *format*, GRID for GRID *format*, NAV for SYSN *format*, POINT for MD *format*, TEXT for TEXT *format*)

General Remarks

The *dataset* parameter must be specified in the group/descriptor format. The total length of *dataset*, including the slash, must be 21 characters or less. The group name is limited to eight characters or less and the descriptor name is limited to 12 characters or less. The group name cannot be entirely numeric. The following characters are not valid in the group or descriptor name: slash (/), period (.), blank ( ), left bracket ([ ), right bracket ( ]).

The ADD option overwrites an existing ADDE dataset name without warning. This means you cannot assign the same ADDE dataset name to more than one dataset, even if the datasets have different file formats.

After adding the dataset name, the ADD option maps the dataset's group name to the local server in the DATALOC table. If the group name is

already mapped to a remote server, the DATALOC table entry is not changed.

DIRFILE Keyword Remarks

Use the DIRFILE='*mask*' keyword to access non-McIDAS format files or McIDAS area, grid or MD files with names other than **AREA***nnnn*, **GRID***nnnn* and **MDXX***nnnn* (where *nnnn* is a 4-digit number) and/or files outside the **MCPATH** and REDIRECT directories. The *mask* value contains the location and names of the files (i.e., the directory and file masks), and is specified using standard Unix substitution rules and wildcards. The file mask consists of all characters after the last slash; everything before and including the last slash is the directory mask. For example, if you specify DIRFILE='/home/user/goes/images/199?/Conus*', then /home/user/goes/images/199?/ is the directory mask, Conus* is the file mask, and the entry matches all files beginning with "Conus" that reside in any /home/user/goes/images/ subdirectory whose name is four characters long and begins with "199". If you don't specify any slashes in *mask*, there is no directory mask so the entry is treated as a file mask only and its file(s) must be located in a **MCPATH** directory.

Absolute position numbers in datasets created using the DIRFILE keyword are determined in the order that the files are found by the server, beginning with position 1. Thus, the absolute position number of a file can change if another file is added to or removed from the dataset. Relative position numbers work as expected, so position 0 is the most recent image, -1 is the next most recent, etc.

AIRS HDF Dataset Remarks

You can access Aqua AIRS Level 1b HDF files with the IMG* commands by using DSSERVE to assign a dataset name to the files. When doing so, specify "AIRS" in the *format* parameter, TYPE=IMAGE, and the directory and file masks in the DIRFILE keyword. The file names must match the standard naming convention that includes the 5-minute granule number (JPL format).

COMS HRIT Dataset Remarks

You can access COMS HRIT files with the IMG* commands by using DSSERVE to assign a dataset name to the files. When doing so, specify "COMS" in the *format* parameter, TYPE=IMAGE, and the directory and file masks in the DIRFILE keyword. The file names must match the KMA segmented image file naming convention, **IMG_ppp_08_cccc_YYYYMMDD_hhmmss_nn.hrit** or **IMG_ppp_08_cccc_YYYYMMDD_hhmmss_nn.hrit.xrit**, where **ppp** is the projection (FD, ENH, LSH, APNH or LA), **cccc** is the spectral channel (VIS, SWIR, WV, IR1 or IR2), **YYYYMMDD_hhmmss** is the date and time, and **nn** is the segment file number (01-10).

GeoTIFF Dataset Remarks

GeoTIFF images are similar to TIFF images, but include calibration and navigation information. They are used by the GIS community and can be viewed with *ARCVIEW* or other GIS software packages.

You can create GeoTIFF image files from other TYPE=IMAGE datasets. To do so, make a GeoTIFF image dataset with a DSSERVE command similar to: **DSSERVE ADD GEOTIFF/GOES GEOT 1 9999 TYPE=IMAGE "GeoTIFF GOES image files**. Then specify GEOTIFF/GOES as the destination dataset with a positive position number in an IMGCOPY, IMGFILT, IMGOPER or IMGREMAP command. If you use IMGCOPY, IMGFILT or IMGOPER, the source image must be in a rectilinear projection. If you use IMGREMAP, the source image can be in any projection, but you must specify PRO=RECT to make the destination GeoTIFF image a rectilinear projection. IMGFILT, IMGOPER, and IMGREMAP save the image with BRIT calibration. IMGCOPY can save it with other calibrations by using the UNIT keyword.

The output file name consists of the descriptor name and position number followed by **.tif**. For example, the command **IMGCOPY GOESEAST/VIS GEOTIFF/GOES.300 SIZE=ALL** creates an output file named **GOES0300.tif**.

GRIB Dataset Remarks

You can access GRIB files in GRIB1 or GRIB2 format with the GRD* commands by using DSSERVE to assign a dataset name to the files. When doing so, specify "GRIB" in the *format* parameter, TYPE=GRID, and the directory and file masks in the DIRFILE keyword. Only GRIB1 and GRIB2 files from the NOAAPORT data stream have been tested and are fully supported.

MODIS HDF Dataset Remarks

You can access MODIS Level 1b HDF files with the IMG* commands by using DSSERVE to assign a dataset name to the files. When doing so, specify "MODS" in the *format* parameter, TYPE=IMAGE, and the directory and file masks in the DIRFILE keyword.

You can access MODIS Level 2 MOD06 (cloud top properties), MOD07 (atmospheric profile) and MOD35 (cloud mask) products in HDF format with the IMG* commands by using DSSERVE to assign a dataset name to the files. When doing so, specify "MODX" in the *format* parameter, TYPE=IMAGE, and the directory and file masks in the DIRFILE keyword.

You can access MODIS Level 2 MOD04 (aerosol), MOD28 (sea surface temperature), and MODR (corrected reflectance) products in HDF format with the IMG* commands by using DSSERVE to assign a dataset name to the files. When doing so, specify "MOD4" (for MOD04 aerosol products), "MOD8" (for MOD28 sea surface temperature products), or "MODR" (for MODR corrected reflectance products) in the *format* parameter, TYPE=IMAGE, and the directory and file masks in the DIRFILE keyword.

You can access MODIS (and potentially other) HDF files as point data in MODS-*format* datasets created using DSSERVE with the TYPE, DIRFILE and INFO keywords. The TYPE keyword specifies the data type (POINT) of the HDF files in the dataset. The DIRFILE keyword specifies the directory and file masks to locate the HDF files. The INFO keyword specifies the name of the configuration file that maps the HDF file parameter names to McIDAS parameter names. The configuration file must be tailored specifically to the HDF files in the dataset. Two sample configuration files, named **MOD06.cfg** and **MOD07.cfg**, are available in the **~mcidas/data** directory. Refer to those files when creating configuration files for your HDF point datasets.

When creating MODIS HDF image or point datasets, the file names must match the standard Goddard DAAC (Distributed Active Archive System) or MODIS direct broadcast naming conventions. In the DAAC convention, for example, 1 km Terra MODIS Level 1b HDF image files must follow the format **MOD021KM.Ayyyyddd.hhmm.002.yyyyddd.hdf**. Aqua files must begin with **MYD** instead of **MOD**, and 1/2 km and 1/4 km resolution files are denoted as **HKM** and **QKM** instead of **1KM**. In the MODIS direct broadcast convention, for example, the same 1 km Terra image files must follow the format **t1.yyddd.hhmm.1000m.hdf**. Aqua files must begin with **a1** instead of **t1**, and 1/2 km and 1/4 km files are denoted as **500m** and **250m** instead of **1000m**. Similarly, these two naming conventions can be applied to the MODIS Level 2 products in HDF format.

MSG Dataset Remarks

You can access real-time segmented Meteosat Second Generation Level 1.5 files transmitted by EUMETSAT with the IMG* commands by using DSSERVE to assign a dataset name to the files. When doing so, specify "MSGT" in the *format* parameter, TYPE=IMAGE, and the directory and file masks in the DIRFILE keyword. Also, the dataset descriptor must be "HRV" if the dataset contains high resolution visible channel (band 12) data, and either "FD" or "VISIR" if the dataset contains full disk low resolution visible and infrared channel (bands 1-11) data.

MTSAT HRIT Dataset Remarks

You can access MTSAT HRIT files with the IMG* commands by using DSSERVE to assign a dataset name to the files. When doing so, specify "MTST" in the *format* parameter, TYPE=IMAGE, and the directory and file masks in the DIRFILE keyword. The file names must match the JMA segmented image file naming convention described at http://www.ssec.wisc.edu/mcidas/doc/sdi_man/current/mtsat_hrit-3.html.

---

netCDF Dataset Remarks

You can access netCDF grid and point files in NCDF-*format* datasets created using DSSERVE with the TYPE, DIRFILE and INFO keywords. The TYPE keyword specifies the data type (GRID or POINT) of the netCDF files in the dataset. The DIRFILE keyword specifies the directory and file masks to locate the netCDF files. The INFO keyword specifies the name of the configuration file that maps the netCDF file parameter names to McIDAS parameter names. The configuration file must be tailored specifically to the netCDF files in the dataset. Two sample configuration files, named **netcdfgrid.cfg** and **netcdfpoint.cfg**, are available in the **~mcidas/data** directory. Refer to those files when creating configuration files for your netCDF grid and point datasets.

You can create netCDF image files from other TYPE=IMAGE datasets. To do so, make a netCDF image dataset with a DSSERVE command similar to: **DSSERVE ADD NETCDF/GOES NCDF 1 9999 TYPE=IMAGE "netCDF GOES files**. Then specify NETCDF/GOES as the destination dataset with a positive position number in an IMGCOPY, IMGFILT, IMGOPER or IMGREMAP command. The output file name consists of the descriptor name and position number followed by **.nc**. For example, the command **IMGCOPY GOESEAST/VIS NETCDF/GOES.300 SIZE=ALL** creates an output file named **GOES0300.nc** that's mapped to position 300 of the dataset NETCDF/GOES.

POES and Metop Level 1b Dataset Remarks

You can access AVHRR Level 1b files containing Metop FRAC or POES/NOAA imagery from NOAA CLASS (http://www.class.noaa.gov) with the IMG* commands by using DSSERVE to assign a dataset name to the files. When doing so, specify "LV1B" in the *format* parameter, TYPE=IMAGE, and the directory and file masks in the DIRFILE keyword. To process the data (during client command data requests) using the file's line-by-line lat/lon navigation (rather than the orbital navigation at the beginning of the file), also include the keyword INFO=LALO. You can compare the two navigation types by using DSSERVE to make two datasets that are identical except that one includes INFO=LALO.

The two types of AVHRR Level 1b files that can be used are 10-bit files (which inherently have all channels) and 16-bit files that contain all channels. You can obtain 16-bit files with only one channel from CLASS, but they won't work in McIDAS ADDE servers. In either case (whether you choose 10-bit or 16-bit), the file must contain the entire pass, not a subset of it.

You can copy POES/NOAA AVHRR data in McIDAS readable format (McIDAS Area with 192 byte doc section) to the CLASS Level 1b file format with IMGCOPY by using DSSERVE to assign a dataset name to the files. When doing so, specify "LV1B" in the *format* parameter, TYPE=IMAGE, and the directory and file masks in the DIRFILE keyword. The output file name consists of the descriptor name and position

number followed by the extension **.l1b**. For example, the command **IMGCOPY WALLOPS/N17HRPT LV1B/17HRPT.2 SIZE=SAME DOC=YES BAND=ALL** creates an output file named **17HRPT0002.l1b** in the directory specified in the DIRFILE keyword. The IMGCOPY command must include SIZE=ALL or DOC=YES with BAND=ALL and SIZE=SAME or SIZE= with any value for its *lines* parameter and either 2048 (for HRPT or LAC data) or 409 (for GAC data) for its *ele* parameter.

System Navigation Dataset Remarks

When adding a SYSN-*format* dataset, the descriptor portion of the dataset name must exactly match a satellite name in the text file **~mcidas/data/CORE.SAT**. The server uses this file to map the descriptor to the correct sensor source number and locate the needed information in the dataset's system navigation files. For example, NAV/GOES9, NAV/GOES-9, NAV/G9, NAV/G-9 and NAV/WEST are all valid dataset names because all the descriptors are entries in the **CORE.SAT** file. You cannot add new satellite names or edit existing names in **CORE.SAT**; you must use a satellite name that exists in the SSEC-supplied version of the file.

Text Dataset Remarks

When adding a TEXT-*format* dataset, you must specify the text file name with the INFO keyword. The file name is case-sensitive. If you do not include the full path name, the file must be located in a **MCPATH** or REDIRECT directory. For example, you can create a TEXT-*format* dataset containing a NORAD Two-Line Element (TLE) file, then list and display its data with the NAVCALC and NAVDISP commands. To do so, place the TLE file into a MCPATH or REDIRECT directory, then make the dataset with a command similar to: **DSSERVE ADD NAV/MODIS TEXT INFO='TLE.TXT' "TLE file with TERRA and AQUA orbits**.

Examples

DSSERVE

This entry lists information about all datasets on the local server.

DSSERVE ADD LOCALDATA/AREAS AREA 1 9999 "Areas in my data directory

This entry assigns the dataset name LOCALDATA/AREAS to areas 1 through 9999 on the local server.

DSSERVE ADD NAV/NOAA-14 SYSN 1 6 "NOAA-14 System Navigation

This entry assigns the dataset name NAV/NOAA-14 to system navigation files 1 through 6 on the local server.

DSSERVE ADD HURRICANE/GOES-WV-8KM AREA 6001 6100 "Hurricane Bob images

This entry assigns the dataset name HURRICANE/GOES-WV-8KM to areas 6001 through 6100 on the local server.

DSSERVE ADD ADMIN/SATDATA TEXT INFO='/u/oper/mcidas/info/satdata.txt' "Listing of all available satellite data

This entry assigns the dataset name ADMIN/SATDATA to the text file **/u/oper/mcidas/info/satdata.txt**. Use the READ command to list the file.

DSSERVE ADD GOES/VIS_4KM HDF TYPE=IMAGE INFO='/users/ingest/hdfdata' "GOES-8 HDF data

This entry assigns the dataset name GOES/VIS_4KM to HDF image data on the local server. The directory specified with the INFO keyword is used by the locally-developed HDF server to locate the data.

DSSERVE LIST HURRICANE

This entry lists information about all datasets in the group HURRICANE on the local server.

DSSERVE DEL HURRICANE/GOES-WV-8K

This entry deletes the dataset name HURRICANE/GOES-WV-8K from the local server, but does not affect the data to which the dataset name was assigned.

DSSERVE ADD GRID/NETCDF NCDF TYPE=GRID INFO='NCDF_ETA.CFG' DIRFILE='/home/user/netcdf/grids/ETA*'

This entry assigns the dataset name GRID/NETCDF to netCDF grid files whose names begin with ETA and are located in the **/home/user/netcdf/grids/** directory. The INFO keyword specifies the configuration file (NCDF_ETA.CFG) that maps the netCDF file parameter names to McIDAS parameter names.

E

Lists the earth coordinates at the cursor center.

Remarks

To list the earth coordinates at the cursor center,

Press: **Alt E**

or

Type: **E**

Press: **Enter**

Command E lists:

- the latitude and longitude of a pixel on a navigated frame in the format HH:MM:SS
- satellite image line and element coordinates
- TV raster and pictel coordinates; position raster=1 and pictel=1 is the upper-left corner of the frame

The output format of E is shown below. The E key lists the coordinates of the current image frame if it's toggled on, and the coordinates of the current graphics frame if the image frame is toggled off. Press **K** to toggle the image frame.

```
Frame Latitude Longitude Tvline Tvelem Line Elem nn DDD:MM:SS DDD:MM:SS nnnn nnnn
nnnn nnnn
```

EB

Black and white contrast stretching.

Format

EB *inlo inhi outlo outhi*

Parameters

inlo	lowest input brightness value (default=0)
inhi	highest input brightness value (default=255)
outlo	new low brightness value (default=0)
outhi	new high brightness value (default=255)

Remarks

The input brightness values correspond to the brightness values on the image frame. Input brightness values below *inlo* and above *inhi* are not changed. *Inlo* is output at value *outlo* and *inhi* at value *outhi*. Brightness values between *inlo* and *inhi* are output at values determined by linear interpolation from *outlo* and *outhi*.

When using the mouse-controlled version, moving the cursor to the right along the x-axis decreases the upper brightness bound (from level 255) to be enhanced as white (255). Moving the cursor upward along the y-axis increases the lower brightness bound (from level 0) to be enhanced as black (0).

Occasionally, a brightness range will contain a value that will not need to change as the new values are assigned by EB. In the example below, the data's brightness value of 104 happens to match the value assigned during the EB stretch. Since value 104 did not need to change, it becomes a breakpoint in the EU TABLE listing and is not shown on the output below. Command **EB 62 164 2 250** produces output from EU TABLE like this:

```
Brightness Blue Green Red min max min max min max min max --- --- --- --- --- --- -
-- --- 62 103 2 102 2 102 2 102 105 164 107 250 107 250 107 250
```

See Also

Command EU SAVE will save the current enhancement. Command EU MAKE will apply color enhancements to any frame.

Examples

EB

This entry invokes interactive mouse-controlled contrast stretching. Press the right mouse button to end the process, leaving the current enhancement.

EB 50 150 0 255

This entry rescales input brightness level 50 as 0 (black) and input brightness level 150 as 255 (white). The 101 counts from 50 to 150 are linearly stretched along the 256 counts from 0 to 255.

EB 30 255 50 200

This entry rescales input brightness level 30 as 50, and input brightness level 255 as 200. The 226 counts from 30 to 255 are linearly stretched along the 151 counts from 50 to 200.

ECHO

Displays a line of text on a text frame.

Format

ECHO *"text*

Parameter

"text text to display on the text frame; the double quote is mandatory

---

Remarks

ECHO is useful in a RUN command.

Examples

ECHO

This entry echoes a blank line.

ECHO "This is a test

This entry prints *This is a test* on the text frame.

EDITMODE

Lists and changes the McIDAS command input editor.

Formats

EDITMODE LIST
EDITMODE *editor*

Parameters

LIST list the available editors, with the current setting denoted (default)

editor editor to use for entering McIDAS commands

- **MCIDAS** standard McIDAS input editor (default)
- **VI** vi input editor
- **EMACS** emacs input editor
- **DEBUG** keyboard input debugging editor; press any key three times to exit to the previous editor

---

Remarks

You can also enter the EDITMODE command as EMODE.

The EDITMODE vi and emacs editors support a small subset of the commands available in the actual editors. See the list of the supported commands below.

The EDITMODE vi editor supports the following commands.

Command	Function
ESC key	Switch to the vi command editor from the McIDAS editor.
l	Move the cursor forward (right) one character.
w or W	Move the cursor forward one word.
h	Move the cursor back (left) one character.
b or B	Move the cursor back one word.
0	Move the cursor to beginning of the line.
$	Move the cursor to end of the line.
k or -	Recall the previous command.
j or +	Recall the next command.
a	Return to the McIDAS editor and enter text after the current character.
A	Append text to the end of the line. Equivalent to $a.
c*motion*	Delete the current character through the character defined by *motion*, and return to the McIDAS editor. For example, cw deletes a word starting at the cursor position and returns to the McIDAS editor. If *motion* is c, the entire line is deleted.
C	Delete the current character through the end of line and return to the McIDAS editor. Equivalent to c$.
s	Delete the current character and return to McIDAS editor.
S	Delete the current line and return to the McIDAS editor. Equivalent to cc.
D	Delete the current character through the end of line. Equivalent to d$.
d*motion*	Delete current character through the character defined by *motion*. For example, dw

	deletes a word starting at the cursor position. If *motion* is d, the entire line is deleted.
i	Return to McIDAS editor and insert text before the current character.
I	Insert text before the beginning of the line. Equivalent to 0i.
R	Return to McIDAS editor and replace characters on the screen with characters you type overlay fashion.
x	Delete the current character.
X	Delete the preceding character.

EDITMODE emacs editor supports the following emacs-like commands:

Command	**Function**
^F	Move the cursor forward (right) one character.
^B	Move the cursor back (left) one character.
^A	Move the cursor to beginning of the line.
^E	Move the cursor to end of the line.
^H	Delete the previous character.
^D	Delete the current character.
^K	Delete the current line starting at the cursor.
^G	Delete the current line.
^U	Delete the current line. Same as ^G.
^P	Recall the previous command.
^N	Recall the next command line.

Examples

EDITMODE

This entry lists the available McIDAS command input editors, noting the current setting with *[cur]*.

EDITMODE VI

This entry switches the McIDAS command input editor to the vi editor allowing you to edit the command line using the vi-like commands listed in the Remarks.

EG

Erases graphics and images.

Formats

EG *bframe eframe [keyword]*
EG I *bframe eframe*

Parameters

bframe	beginning frame in the range to erase (default=current)
eframe	ending frame in the range to erase (default=*bframe*)
I	erases the image and the frame directory

Keyword

LEV=	graphics color level to erase; only the graphics drawn in this color are erased (default=all levels)

Remarks

The EG *bframe eframe* format erases only graphics; the image and the frame directory (information about what's displayed in the frame) are not erased. The EG I *bframe eframe* format erases the image and the frame directory (but not the graphics). The EG command does not change the current image and graphic enhancement tables.

If you have multiple panels on a frame, you can erase a range of panels by specifying beginning and ending panel numbers with the global keyword PAN. For example, **EG 1 5 PAN=2 4** erases the graphics on panels 2, 3, and 4 on frames 1 through 5.

See Also

Command ERASE will also erase graphics and images, and in addition, it can also erase entire frames.

Examples

EG

This entry erases graphics on the current frame.

EG 3 5

This entry erases the graphics on frames 3, 4 and 5.

EG I 2 3

This entry erases the images and the frame directory information on frames 2 and 3.

EG LEV=3

This entry erases graphics drawn in color level 3 on the current frame.

ERASE

Erases graphics, images or entire frames.

Formats

ERASE F *bframe eframe*
ERASE G *bframe eframe* **LEV=**
ERASE I *bframe eframe*

Parameters

bframe	number of beginning frame in the range to erase (default=current)
eframe	number of ending frame in the range to erase (default=*bframe*)
F	erase entire frame: image, graphics and frame directory (default)
G	erase graphics only
I	erase image and frame directory

Keyword

LEVel=	graphics color level to erase; only the graphics drawn in this color are erased; valid only with "ERASE G" format (default=all levels)

Remarks

The ERASE command does not change the current image and graphic enhancement tables.

If you have multiple panels on a frame, you can erase a range of panels by specifying beginning and ending panel numbers with the global keyword PAN. For example, **ERASE G 1 5 PAN=2 4** erases the graphics on panels 2, 3, and 4 on frames 1 through 5.

Examples

ERASE

This entry erases the current frame's image, graphics, and frame directory (information about what's displayed in the frame).

ERASE F 3 6

This entry erases the images, graphics, and frame directories on frames 3 through 6.

ERASE G 3 6 LEVEL=3

This entry erases the graphics drawn in color level 3 on frames 3 through 6.

ERASE I 21 22

This entry erases the images and the frame directories on frames 21 and 22.

EU

Enhancement utility.

Formats

EU COLORS
EU DEL *name*
EU LIST *string*
EU MAKE *inlo inhi bcolor ecolor bframe eframe [keywords]*
EU MAKE *inlo inhi 1lo 1hi 2lo 2hi 3lo 3hi bframe eframe [keywords]*
EU REST *name bframe eframe [keywords]*
EU SAVE *name*
EU TABLE *name [keywords]*

Parameters

COLORS	lists the predefined colors in the **MCRGB.TXT** data file for use in the *bcolor* and *ecolor* parameters
DEL	deletes a saved enhancement
LIST	lists the saved enhancements
MAKE	sets the enhancement
REST	restores a saved enhancement (default)
SAVE	saves the current enhancement
TABLE	lists the values of an enhancement
name	enhancement name; 512 alphanumeric characters maximum (REST option default=default grayscale enhancement; TABLE option default=current; no default for DEL and SAVE options)
string	lists only the enhancements beginning with this character string (no default; list all enhancements)
inlo	lowest brightness value of displayed image to apply the MAKE enhancement change to; the range is 0 to 255 (default=0); the resultant changes in color from *inlo* to *inhi* are done by interpolation from *bcolor* to *ecolor* or from *1lo*, *2lo*, *3lo* to *1hi*, *2hi*, *3hi*
inhi	highest brightness value of displayed image to apply the MAKE enhancement change to; the range is 0 to 255 (default=0)
bcolor	predefined color to apply to *inlo*; use EU COLORS to list the valid colors
ecolor	predefined color to apply to *inhi* (default=*bcolor*); use EU COLORS to list the valid colors
bframe	beginning image frame on which to apply the enhancement (default=current)
eframe	ending image frame on which to apply the enhancement (default=*bframe*)
1lo	intensity of the first color component to apply to *inlo*
1hi	intensity of the first color component to apply to *inhi*
2lo	intensity of the second color component to apply to *inlo*
2hi	intensity of the second color component to apply to *inhi*
3lo	intensity of the third color component to apply to *inlo*
3hi	intensity of the third color component to apply to *inhi*; the valid range for parameters *1lo*, *1hi*, *2lo*, *2hi*, *3lo* and *3hi* is 0 to 255 (default=0); the first color component is either blue or red, the second color component is green, the third color component is either

red or blue; see the Remarks for more information

Keywords

FORm=	**ALL**	formats the TABLE option output to list each brightness level and its coresponding enhancement values on a separate line
	STN	formats the TABLE option output to list sets of brightness levels and their coresponding enhancement values (default)
LOOP=	**YES**	applies the enhancement to all frames in the loop; overrides the *bframe* and *eframe* values
	NO	does not apply the enhancement to all frames in the loop (default)
RGB=	**YES**	changes the order of the color components from the default (blue-green-red) to red-green-blue in the MAKE option's *1lo*, *1hi*, *2lo*, *2hi*, *3lo* and *3hi* input parameters and the TABLE option's output listing
	NO	keeps the default order of the color components (default)

Remarks

Parameters *1lo*, *2lo*, *3lo* and *1hi*, *2hi*, *3hi* are the intensities (0-255) of the three color components that make up the final color applied to the *inlo* and *inhi* input brightness values. The RGB keyword determines the order in which the three color components are specified:

- If RGB=YES is specified, then the color order is red, green, blue (*1lo* and *1hi* are RED intensities, *2lo* and *2hi* are GREEN intensities, and *3lo* and *3hi* are BLUE intensities).
- If RGB=NO (the default), then the color order is blue, green, red (*1lo* and *1hi* are BLUE intensities, *2lo* and *2hi* are GREEN intensities, and *3lo* and *3hi* are RED intensities).

Brightness values are only listed with EU TAB when they vary from the default table, i.e., brightness values 0 to 255 have intensities 0 to 255 for blue, green and red. For example, **EU MAKE 20 120 20 120 20 120 20 120** lists nothing because the blue, green and red intensities have not altered the default range.

Occasionally, a brightness range will contain a value that will not need to change as the new values are assigned by command EB. In the example below, the data's brightness value of 104 happens to match the value assigned during the EB stretch. Since value 104 did not need to change, it becomes a breakpoint in the EU TABLE listing and is not shown on the output below. Command EB 62 164 2 250 produces the following output from EU TABLE.

```
Brightness Blue Green Red min max min max min max min max --- --- --- --- --- --- -
-- --- 62 103 2 102 2 102 2 102 105 164 107 250 107 250 107 250
```

Multipanel frames (created with the PANEL command) use a single color table. So when you apply an enhancement to a frame (with the EU or EB command), the enhancement is applied to all panels on the frame. That is, you cannot have EU or EB apply one enhancement to one panel, and a different enhancement to another panel.

However, you can simulate different enhancements on each panel by displaying the image on each panel with its own set of brightness values. To do so, use the SU command to make the stretch tables, then display the images using IMGDISP with the SU= and PAN= keywords to select the appropriate stretch table for each image/panel. Then use EU to create separate enhancements specifically for each panel so that only its colors change when the enhancement is applied.

For example, if you have a four-panel frame you can make four SU tables, each stretching a different ranges of brightness values with commands like these:
SU INI PAN1; SU MAKE PAN1 0 255 0 63
SU INI PAN2; SU MAKE PAN2 0 255 64 127
SU INI PAN3; SU MAKE PAN3 0 255 128 191
SU INI PAN4; SU MAKE PAN4 0 255 192 255
Then display the imagery with commands like IMGDISP *dataset* PAN=1 SU=PAN1, IMGDISP *dataset* PAN=2 SU=PAN2, etc. You can then use EU to make an enhancement table for each of the images/panels. Enhancements for brightnesses 0-63 (e.g., EU MAKE 0 63 BLUE PURPLE) will apply to panel 1, enhancements for brightnesses 64-127 will apply to panel 2, enhancements for brightnesses 128-191 will apply to panel 3, and enhancements for brightnesses 192-255 will apply to panel 4.

See Also

Use command EB for interactive, mouse-controlled contrast stretching.

Examples

EU

This entry restores the default enhancement to the current frame.

EU LIST

This entry lists all the saved enhancement tables on the workstation.

EU LIST VAP

This entry lists all the saved enhancement tables beginning with VAP.

EU MAKE 20 75 255 255 0 0 255 255

This entry color enhances all image pixels with brightness values between 20 and 75 to solid magenta.

EU MAKE 100 200 RED YELLOW 1 3

This entry color enhances the image pixel brightness values between 100 and 200 on image frames 1 to 3 from red to yellow.

EU SAVE VAPOR

This entry saves the enhancement on the current frame to a file named VAPOR.

EU REST VAPOR LOOP=YES

This entry restores an enhancement called VAPOR to all frames defined in a loop.

EU TABLE

This entry lists the values of the current enhancement.

EXIT

Ends a McIDAS-X session and associated windows.

Remarks

Use the EXIT command to end the McIDAS-X session and close its associated windows. This method is preferred over manually closing down the windows with the mouse buttons.

To restart McIDAS-X after using EXIT, type the following command from the Unix command prompt and press **Enter**.

Type: **mcidas**

F

Displays the workstation's state.

Remarks

To list the workstation's state,

Press: **Alt F**

or

Type: **F**

Press: **Enter**

The output format of F is workstation dependent; a sample output is shown below.

```
Video Status for Your Workstation Image Frame(s) Graphics Frame(s) Number Available
nn nn Current nn (Opp=nnn) nn Loop Bounds n to n / Random n to n Visible (K & W
toggle) Yes/No Yes/No Connected (Y & J toggle) Yes/No Yes/No Looping (L toggle)
Yes/No Yes/No Cursor parameters: Size= nnn / nnn Type=cursor type Center position=
nnn / nnn Color=color Image frames nn - nn and Graphics frames nn - nn are nnnn BY
nnnn.
```

If you use command LB to set image or graphics loop bounds, the table indicates the loop bound set. If you use command LS to set the loop bounds, random looping is in effect. In this case, you can enter LS or LS G to list the current loop definition.

The final lines of the output information describe the number and size of the image and graphics frames. Add new frames with the MAKFRM command.

Use commands K, W, J, Y and L to toggle elements of the loop control system on and off. Command K toggles the image. W toggles the graphics. J toggles the graphics loop control. Y toggles the image loop control. L toggles the image and graphics looping.

FIND

Locates U. S. towns by name and state.

Format

FIND *state* '*string*' **E**

Parameters

state	two-letter state Post Office abbreviation
'*string*'	character string; lists all towns beginning with this string
E	specify E after '*string*' to list only exact matches

Remarks

The parameters are not case sensitive.

If only one match is found, the cursor is moved to its location.

The town name, county, latitude, longitude and elevation, if available, are listed on the text frame.

This command uses the files **towns.nd1**, **towns.nd2** and **towns.txt**, which are derived from data provided by the U.S. Geological Survey.

Examples

FIND NY 'NEW YORK'

This entry lists all New York towns beginning with NEW YORK. They are New York and New York Mills.

FIND NY 'NEW YORK' E

This entry lists only towns in New York that exactly match the string NEW YORK. The cursor is moved to its location if it's on the current frame.

FIND WI 'MAD'

This entry lists all Wisconsin towns beginning with MAD. They are Madge, Madison and Madsen.

FONT

McIDAS graphics font utility.

Formats

FONT LIST
FONT RESET
FONT SET *type*

Parameters

LIST	lists the available font types (default)		
RESET	resets the font type to the default McIDAS font		
SET	sets the font type for McIDAS graphics; this option lists the current setting if you don't specify *type*		
type	font type to use, in the format *fffws*		
	fff	three-character family name	
	w	**B**	bold weight
		M	medium weight
	s	**I**	italic slant
		O	oblique slant
		R	Roman slant

Remarks

You can use the global keyword FONT with any graphics command to override the current font setting. Each font type has an index file with the extension .FFD. The index file lists the font used when a specific size is requested in a McIDAS command. If the size you request is not available, another size is substituted. For example, if you run this command:

ZA 3 20 FONT=COUBO "Courier Bold Font

The text is plotted in 18 point Courier bold oblique rather than 20 point. The index file **COUBO.FFD** specifies that the file **courbo18.bdf** is used when a command requests 20 point text.

The default McIDAS font and those in the Courier family are non-proportional. Use these fonts to display evenly-spaced characters.

See Also

See *Appendix A, Global Keywords* for more information about the global keyword FONT.

Examples

FONT

This entry lists all the available font types.

FONT SET TIMMR

This entry sets the McIDAS graphics font type to Times Roman, medium weight.

FRMLABEL

Displays an image label on a frame.

Format

FRMLABEL *[keywords] "text*

Parameters

"text text to display on the frame's image label; all text is displayed in uppercase; 250 characters maximum; the following placeholders can be specified for text substitution of information from the frame, dataset or displayed image; the output format is indicated in italics, e.g., *nnn*

(FRAME)	frame number, *nnn*
(DATASET)	ADDE dataset name, 24 characters maximum
(POS)	ADDE dataset position, *nnnn*
(BAND)	band number, *nn*
(WL)	wavelength, *nn.nn unit*
(WN)	wave number, *nn.nn unit*
(FREQ)	frequency, *nn.nn unit*
(ID)	NEXRAD station ID, 3 characters
(MEMO)	memo field, 32 characters maximum
(SS)	sensor source number, *nnn*
(STYPE)	image type, 12 characters maximum
(ERES)	element resolution, *nn.nn*
(LRES)	line resolution, *nn.nn*
(RES)	displayed resolution of the image, *nn.nn*
(EGEORES)	geographical element resolution in km, *n.nn*; see the Remarks
(LGEORES)	geographical line resolution in km, *n.nn*; see the Remarks
(GEORES)	geographical resolution in km, (*n.nn* X *n.nn* KM); see the Remarks
(DAY)	day, *dd mon yy*
(MDY)	day, *mon dd yy*
(JDAY)	Julian day, *ccyyddd*
(TIME)	image time, *hhmmss*
(HH)	image time, *hh*
(HHMM)	image time, *hh:mm*
(HHMMSS)	image time, *hh:mm:ss*

Keywords

CRAwl=	**FOREVER**	crawls the text across the frame continuously until stopped with **Alt Q**
	N *num*	number of times the entire text string crawls across the frame; the range is 0 to 100 (default=10); when completed, the text is repositioned to the location specified with keyword JUST
	T *sec*	number of seconds the text crawls across the frame; the range is 0 to 600 (default=120); when the specified number of seconds expires, the text is repositioned to the location specified with keyword JUST
IMA=	image frame number (default=current)	
JUSt=	**C**	centers the text on the frame (default)
	L	left justifies the text on the frame
	R	right justifies the text on the frame
LEVel=	*text bg*	brightness level for the text and background (default=255 0)
LINe=	starting frame line (default=12 lines from bottom of frame)	
NLIne=	number of lines blanked out for text (default=12)	

Remarks

The image label is not a graphic; it is written to the image plane of the frame. The EG command will not erase the label. You must reload the image to erase the image label.

Each character is eight pixels square with a 1-pixel separation. Thus, the maximum number of characters you can display on a frame is the number of elements divided by nine.

The following five characters are not part of FRMLABEL's character set and are thus displayed as a blank (like a space character): tilde (~), grave accent (`), exclamation point (!), left square bracket ([ ), right square bracket ( ]). Also, the left and right curly brackets ({ and }) are displayed as left and right square brackets ([and]).

If you specify keyword CRAWL, the text moves slowly across the screen from right to left. The crawl continues even if a new image is loaded onto the frame. Press **Alt Q** to end the text crawl and reposition the text to the location specified in keyword JUST.

The (GEORES), (LGEORES) and (EGEORES) placeholders can be used only if GEORES=YES was specified in the IMGDISP command that displayed the image. The resolution values, listed as *n.nn* in the Parameters section above, are in kilometers and are displayed in the format *0.nn*, *n.nn*, *nn.n* or *nnn* (depending on the magnitude of the resolution). The two resolution values displayed with the (GEORES) placeholder are the same as (LGEORES) and (EGEORES), in that order.

Examples

FRMLABEL "GOES-9 INFRARED IMAGE

This entry displays an image label with the text *GOES-9 INFRARED IMAGE* on the bottom 12 lines of the frame.

FRMLABEL LINE=1 CRAWL=FOREVER "SEVERE THUNDERSTORM WARNING FOR DANE COUNTY

This entry displays an image label with the specified text on the top 12 lines of the frame. The text continuously crawls across the frame from right to left until **Alt Q** is pressed.

FRMLABEL JUST=L "GOES-8 BAND (BAND) FROM (DAY) AT (HHMM) UTC

This entry displays an image label with the specified text left justified on the bottom 12 lines of the frame. The placeholders (BAND), (DAY) and (HHMM) are replaced with the corresponding values from the displayed image.

FRMLABEL LINE=22 LEVEL=0 255 "IMAGE FROM DATASET (DATASET), POSITION (POS)

This entry displays an image label with the specified text in black (brightness level 0) with a white (brightness level 255) background beginning at

line 22. The placeholders (DATASET) and (POS) are replaced with the corresponding values from the displayed image.

FRMLABEL LEVEL=232 2 "(ID) RADAR - BASE REFLECTIVITY TILT 1 AT (HHMM) ON (DAY)

This entry is very similar to the FRMLABEL command in the MKX_BREF1 predefined image product in the **IMGDISP.CORE** file. The product displays a NEXRAD Base Reflectivity image and uses this FRMLABEL command to display an image label with the (ID), (HHMM) and (DAY) placeholders replaced by the NEXRAD station ID, time and date of the displayed image.

FRMLIST

Lists frame directory information.

Format

FRMLIST *bframe eframe*

Parameters

bframe beginning frame number to list (default=current)

eframe ending frame number to list (default=*bframe*)

Keywords

Form=	**ALL**	lists the entire frame directory
	EXP	provides an expanded listing containing the sensor source number and description, Julian date, time, image coordinates of the upper-left pixel, and the data source
	PANEL	provides a standard listing for all panels in each frame, plus panel size
	STN	provides a standard listing containing the sensor source description, date, time and data source (default)

Remarks

FORM=PANEL is useful with multipanel frames. It lists frame directory information for all panels on each frame; FORM=ALL, EXP, and STN list information only for the current panel or the panel specified with the global keyword PAN.

Examples

FRMLIST

This entry lists frame directory information for the current frame.

FRMLIST 2 6

This entry lists frame directory information for frames 2 through 6.

FRMLIST 2 6 FORM=EXP

This entry lists expanded frame directory information for frames 2 through 6.

FRMSAVE

Saves a McIDAS frame to one of several image formats.

Formats

FRMSAVE *frame name [keywords]*
FRMSAVE *window name* **TYPE=COMBINE** *[keywords]*

Parameters

frame image frame number to save (def=current)

window combined window number (created with COMBINE command); the range is 0-9; you must also specify TYPE=COMBINE with this option

name name of the file in which to save the frame; if you don't specify an extension, .GIF is appended if FORM=GIF, .PPM is appended if FORM=PPM, .BMP is appended if FORM=BMP, .JPG is appended if FORM=JPG, .PS is appended if FORM=PS, and .CPS is appended if FORM=CPS

Keywords

GRA= graphics frame to save with the image; valid only for TYPE=FRAME; see the Remarks (default=number specified in *frame* if the session does not have independent graphics; default=last-associated graphics frame if the session has independent graphics)

FORm= format of saved file; valid entries are GIF, PPM, BMP, JPG, PS and CPS for graphics interchange, portable pixmap, bitmap, joint photographic experts group, PostScript, and color PostScript formats, respectively (default=GIF for TYPE=FRAME; default=JPG for TYPE=COMBINE)

QUAlity= JPEG image quality; the range is 0 - 100 with higher numbers representing better quality; you must also specify FORM=JPG with this keyword (default=100)

TYPe=

FRAME save the frame specified in *frame* (default)

COMBINE save the combined window specified in *window*; you must specify this keyword when using the **FRMSAVE** *window name* format

---

Remarks

You can view files created with FRMSAVE with a Unix viewer program like **xv** or **xloadimage**.

FRMSAVE can be used with single- or multi-panel frames, but it always saves the entire frame (all panels) even if you specify the global keyword PAN.

Image frames and graphics frames are "associated" once they have been viewed together. Sessions without independent graphics automatically associate each image frame with the graphics frame of the same number once the frame has been displayed. However, sessions with independent graphics have separate control (and display) of image and graphics frames so image frames are not necessarily associated with the graphics frames of the same number. For example, if you loop through image frames 1-10 while displaying graphics frame 1, all ten of the image frames are associated with graphics frame 1. This is relevant because the GRA keyword defaults to the last-associated graphics frame for the image frame specified in the *frame* parameter. In this example, command entry **FRMSAVE 7 GOESIR** would save graphics frame 1 with image frame 7 in the file GOESIR.GIF.

Examples

FRMSAVE X IMAGE

This entry saves the current image and graphics frame into the file **IMAGE.GIF**.

FRMSAVE 5 PICTURE FORM=BMP

This entry saves image frame 5 and graphics frame 5 into the bitmap file **PICTURE.BMP** in sessions without independent graphics. In sessions with independent graphics, the entry saves image frame 5 and its last-associated graphics frame into the bitmap file **PICTURE.BMP**.

FRMSAVE 11 GOESVIS GRA=1

This entry saves image frame 11 and graphics frame 1 into the file **GOESVIS.GIF**.

FRMSAVE 1 trucolor.jpg TYPE=COMBINE

This entry saves the contents of combined window 1 into the JPEG file **trucolor.jpg**. The combined window was created using the COMBINE command.

FRNTDISP

Displays fronts from NCEP frontal positions products.

Format

FRNTDISP *map time [keywords]*

Parameters

map map for the fronts; use any predefined map in the MAP command, e.g., MID, USA, NA (default=do not draw map; use navigation of frame specified in NAV keyword)

time frontal analysis time or model run time; specify as an hour, **LATEST** or **FRAME** (default=**LATEST**), where:
hour = 0, 3, 6, etc.
LATEST = uses the latest analysis within previous 6 hours
FRAME = uses the nearest analysis within 3 hours of the day and time of the image displayed in the frame

---

Search Keywords

DAY= analysis day or day of model run (default=current)

FHOur= forecast hour for forecast fronts; valid entries are 0, 12, 24, 36, or 48 (default=0 - analysis); not valid with **FRAME** time

FILE= file to read from if not using weather text dataset; the file must contain only frontal positions products (default=none - use weather text files)

GROup= ADDE group to search; the group name must be mapped with the DATALOC command to a server with weather text data (default=RTWXTEXT for weather text; FRONTS for reading from files)

Navigation Keywords

NAV= frame to use for navigation; specify a frame number or **C** for the current frame (default=**C**); no map is drawn; you cannot use this keyword unless the default is used for the *map* parameter

---

Display Keywords

BARbsiz= *cold warm* height of cold and warm barbs (default=plot size dependent)

BASe= length of barb base on front (default=plot size dependent)

COLor= *hi low cold warm ocld stnry trof*
graphics color level for each component (default=6 5 *hi low* 1 *hi* 3), where:

hi	High labels and pressure values
low	Low labels and pressure values
cold	cold front lines and barbs
warm	warm front lines and barbs
ocld	occluded front lines
stnry	stationary front lines

trof dashed trough lines

GRA= graphics frame number to display output (default=current)

LABel= *'text' color line ele size*
text, color, location and size for label, where:

`text' label's text; 72 characters maximum; specify an **X** without quotes for default text (default=computed)

color graphics color level of label (default=3)

line beginning TV line to write label (default=top of frame)

ele beginning TV element to write label (default=left side of the frame minus the length of the label)

size size of label text (default=plot dependent)

LSIZE= *hilo pres* height of High and Low labels, and pressure values (default=plot size dependent)

LWIDth= *front hilo pres* line width of front lines, High and Low labels, and pressure values (default=2 2 1)

ORIent= *cold warm ocld stnry*
side of the cold, warm, occluded, and stationary front lines that the barbs are drawn, as seen moving out from the Low; specify **L** for the left side or **R** for the right side (default=**L L L L**)

OUTput= **PLOT** plot the frontal positions (default)

RAW list the original text product

TAB list the decoded front point and High/Low positions in a tabular format

SPAce= distance between barbs along fronts (default=plot size dependent)

Remarks

FRNTDISP draws fronts using the NCEP frontal positions products issued under WMO headers ASUS1 and FSUS2. It accesses these products through the ADDE server mapped to the group name specified in the GROUP keyword.

The current frontal time is defined to be that of the most recent front product received if it falls within six hours of the current clock time.

The direction that the cold and warm barbs face is defined by the front product. If the product does not follow standard conventions, the barbs may face the incorrect way. You change the direction that they face with the ORIENT keyword.

Examples

FRNTDISP

This entry draws the fronts in the most recent NCEP frontal positions analysis product (WMO header ASUS1) on the current frame. No map is drawn; the current frame's navigation is used.

FRNTDISP USA

This entry draws the fronts in the most recent NCEP frontal positions analysis product (WMO header ASUS1) on a map of the United States.

FRNTDISP X FRAME NAV=8 GRA=8

This entry draws, on graphics frame 8, the fronts in the NCEP frontal positions analysis product (WMO header ASUS1) that is nearest in date and time to the image displayed in frame 8.

FRNTDISP USA 0 FHOUR=24

This entry draws the 24-hour forecast fronts in today's 00 UTC NCEP frontal positions forecast product (WMO header FSUS2) on a map of the United States.

FRNTDISP OUT=RAW

This entry lists the most recent NCEP frontal positions analysis product (WMO header ASUS1) on the text window. No fronts are drawn on the graphics frame. If OUT=TAB is specified instead of OUT=RAW, the data is listed in a decoded tabular format that labels the latitude, longitude and pressure of each High and Low, and the latitude and longitude for each point of the troughs and cold, warm, stationary and occluded fronts.

GD

Defines graphics display parameters.

Formats

GD INI
GD *width dlength gcolor glength*

Parameters

INI	initializes the graphics parameters to their logon values
width	graphics line width in pixels; the range is 1 to 64 (default=last value set; at logon, it is 1)
dlength	length of the dashes in a line; the range is 0 to 64 (default=last value set; at logon, it is 10 pixels)
gcolor	graphics color level of the gaps in dashed lines (default=last value set; at logon, it is 255, which is transparent)
glength	length of the gaps in dashed lines; the range is 0 to 64 (default=last value set; at logon, it is 10 pixels)

Examples

GD

This entry lists the current graphics parameters.

GD 3 5 3 5

This entry defines the line width as three pixels, the length of the dashes and gaps in the dashed lines as five pixels, and the color of the gaps as graphics color level 3.

GEO

Finds and displays a geostationary satellite image over a point or region.

Format

GEO *map channel [keywords]*
GEO *station channel [keywords]*
GEO *lat lon channel [keywords]*
GEO *`city,st,co' channel [keywords]*

Parameters

map map name that defines the geographic region for which to display the image; see the MAP command for a list of valid map names, e.g., FL, MID, USA, EUR (default=value specified for the LOCATION keyword in the **GEO.USER /.SITE /.CORE** text file; the value is set to USA in **GEO.CORE**)

station 3- to 5-character station ID or 5-digit ID number at which to center the image

lat latitude at which to center the image

lon longitude at which to center the image

`city,st,co' city or town at which to center the image, where *city* is the city or town name, *st* is the two-letter state or province code for the specified city and *co* is the two-letter country code for the specified city; if *city* is a U.S. city or town and appears more than once in the state, you can identify the one you want by following the city/town with the county name in brackets, e.g., 'GERMANTOWN[RICHLAND],WI'

channel satellite channel to display; valid options are VIS, SW, WV, IR or SST, where:
VIS = visible cloud and surface features (~ 0.6μm),
SW = shortwave infrared low cloud/fog, fire (~ 3.9 μm),
WV = infrared mid level water vapor (~ 7.0 μm),
IR = infrared surface/cloud top temperature (~11.0 μm),
SST = infrared surface/cloud temperature, low level water vapor (~12.0 μm)
(default=VIS if the image time is between one hour after sunrise and one hour before sunset; otherwise default=IR)

Keywords

DAY= day in which to begin search for image to display (default=current)

GRAphic= graphics frame number to display the map (default=current)

IMAge= image frame number to display the image (default=current)

MCOlor= *pol cnty* graphics color levels for map's political boundaries and county boundaries (in U.S.); use positive numbers for solid lines, negative numbers for dashed lines, and 0 to suppress drawing (default=value specified for the MCOLOR keyword in the **GEO.USER /.SITE /.CORE** text file; the value is set to **7 8** in **GEO.CORE**)

OFF= *op1..opN* set specified options to OFF; see the ON keyword for the list of options and their defaults

ON= *op1..opN* set specified options to ON; the default for all options is the value specified for the corresponding keyword in the **GEO.USER /.SITE /.CORE** text file; the options for the OFF= and ON= keywords and their default setting in **GEO.CORE** are:

- **DEBUG** list output from generated commands in text frame (default=OFF)
- **ECHO** list generated commands in text frame (default=ON)

FILTER apply filter or stretch to image (default=OFF)

MAPS display map over image (default=ON)

RES= resolution, in kilometers, at which to display image (default=resolution of the primary dataset if the *station*, *lat lon*, or *'city,st,co'* command format is used; default=calculated to match the geographic region if the *map* command format is used)

TIMe= time in which to begin search for image to display (default=current)

Remarks

The GEO command uses the **GEO.***, **MAP.*** and **GEODATA.*** context files (where * = **.CORE**, **.SITE** and **.USER**) to determine which data to use and how to display the data. Default parameter and keyword values are defined in the **GEO.*** files. Regional map definitions are defined in the **MAP.*** files. Satellite coverage, instrument channel and resolution information and bordering image datasets are defined in the **GEODATA.*** files. The datasets in **GEODATA.CORE** may not be appropriate for users who do not access their satellite data through the SSEC Data Center. Those users may need to create a **GEODATA.USER** or **.SITE** file in order to use GEO.

The **.CORE** files are supplied with McIDAS-X and should not be modified. You can, however, change their settings to your preferences. To do so, copy the **.CORE** file from **~mcidas/data** to your **$HOME/mcidas/data** or another **MCPATH** directory, change its extension from **.CORE** to **.USER** or **.SITE**, then edit the settings as needed. The **.USER** file overrides the **.SITE** file, which overrides the **.CORE** file. This order of precedence ensures that your local settings (in the **.USER** and/or **.SITE** file) override the settings in the **.CORE** file.

GEO cannot produce a world composite image because the process that merges data from multiple datasets (to create a remapped composite image) is limited to one bordering dataset to the north, south, east and west of the dataset containing the image center point.

All images are remapped based on a derived projection and resolution. The projection is determined by the center point latitude of the image region requested by the command. For center point latitudes -30° to 30°, the projection is Mercator (MERC). For center point latitudes -50° to -30° and 30° to 50°, the projection is Lambert Conformal (LAMB). For all other center point latitudes, the projection is Polar Sterographic (PS).

Examples

GEO

This entry finds the geostationary satellite whose coverage includes the point or region specified for the LOCATION keyword in the **GEO.USER /.SITE /.CORE** file. It then displays the satellite's most recent image at the base resolution of the channel if a point was specified, or at the resolution needed to cover the domain if a region was specified. For most users, the region will be the United States because the LOCATION keyword is set to USA in **GEO.CORE**. For this and all examples below, unless the *channel* parameter is specified, GEO displays the visible channel during daylight hours and displays the infrared channel during nighttime hours.

GEO FL

This entry finds the geostationary satellite whose coverage includes the region defined by the map named FL in the **MAP.USER /.SITE /.CORE** file. It then displays the satellite's most recent image at the resolution needed to exactly cover the region. For most users, the region will be the state of Florida because that's how FL is defined in the **MAP.CORE** file.

GEO EUR WV

This entry finds the geostationary satellite that covers the region defined by the map named EUR in the **MAP.USER /.SITE /.CORE** file. It then displays the satellite's most recent water vapor image at the resolution needed to exactly cover the region. For most users, the region will be Europe because that's how EUR is defined in the **MAP.CORE** file.

GEO YSSY ON=FILTER

This entry finds the geostationary satellite that covers the Sydney, Australia region. It then displays the satellite's most recent image centered on Sydney. A filter is applied to provide optimum contrast in the displayed image.

GEO 27 -122

This entry finds the geostationary satellite that covers the region including 27° North, 122° East. It then displays the satellite's most recent image centered on that point.

GEO `NEW YORK,NY'

This entry finds the most recent GOES-East image, then displays it centered on New York, New York.

GEO `MADRID„ES' IR RES=3

This entry finds the most recent Meteosat infrared image, then displays it at 3 km resolution, centered on Madrid, Spain. Note that two commas appear between the city (Madrid) and country code (ES) because if only one comma is specified, it's assumed to be a state or province code.

GLOBKEY

Lists global keyword parameters and examples.

Format

GLOBKEY *keyword*

Parameter

keyword global keyword to list; valid options are DEV, FONT, MCCOMPRESS, PAN, TCOL, TWIN, VIRT and ALL (default=ALL)

Remarks

You can also refer to *Appendix A, Global Keywords* for global keyword information.

Examples

GLOBKEY

This entry lists information about all global keywords.

GLOBKEY VIRT

This entry lists information about the global keyword VIRT.

GRDCOPY

Copies grids from one dataset to another.

Format

GRDCOPY *sdataset ddataset [keywords]*

Parameters

sdataset source ADDE dataset name and absolute position; specify one of the following formats:

group/descriptor.position

alias.position

only positive integers or ALL (all positions) are valid for *position* (no default for group/descriptor or alias; default=ALL for *position*)

ddataset destination ADDE dataset name and absolute position; specify one of the following formats:

group/descriptor.position

alias.position

use only positive integers for *position*

Search Keywords

DAY=	*d1 . . dn*	copies grids with the specified days
DRAnge=	*bday eday inc*	copies grids in the range of days *bday* through *eday,* incremented by *inc* days (no default for *bday*; *eday* default=*bday*; *inc* default=1)
ENSemble=	*e1 . . en*	copies grids with the specified ensemble numbers; each ensemble number must include a + or - sign, e.g., -1 -0 +0 +1
FDAy=	copies grids with the specified forecast day; a grid's forecast day is determined by adding the forecast hour to the day and time; for example, a 12 UTC grid from day 95300 with a forecast hour of 72 has a forecast day of 95303; you cannot use FDAY with the FHOUR, FRANGE or MATH keyword	
FHOur=	*h1 . . hn*	copies grids with the specified forecast hours; you cannot use FHOUR with the FRANGE, FDAY or FTIME keyword
FRANGe=	*bhr ehr inc*	copies grids in the range of forecast hours *bhr* through *ehr,* incremented by *inc* hours; you cannot use FRANGE with the FHOUR, FDAY or FTIME keyword (no default for *bhr*; *ehr* default=*bhr*; *inc* default=1)
FTIme=	copies grids with this forecast time; a grid's forecast time is determined by adding the forecast hour to the time; for example, a 12 UTC grid with a forecast hour of 18 has a forecast time of 6 on the following day; you cannot use FTIME with the FHOUR, FRANGE or MATH keyword	
GPRo=	*g1 . . gn*	copies grids with the specified projection, for example, MERC, PS, LAMB, EQUI
GRIB=	*geo par model level*	copies grids with the specified GRIB codes; the four values are the geographic, parameter, model and level codes that can be listed with **GRDLIST FORM=ALL**; values specified as an X (a placeholder) or not specified at all will match any GRIB code
GRId=	*bgrid egrid*	copies grids *bgrid* through *egrid*; enter a grid number or LAST (last grid) for *bgrid* and *egrid*; see the Remarks (no default for *bgrid*; *egrid* default=*bgrid*)

LEV= *l1[u1] . . ln[un]* copies grids with the specified levels and units, e.g., SFC, 850[MB], 5000[M]; units are optional but must be in square brackets if specified

PARam= *p1 . . pn* copies grids with the specified parameters, for example, T, Z, RH; see the Remarks

SRC= *s1 . . sn* copies grids with the specified sources, for example, MDX, GFS, ETA

TIMe= *t1 . . tn* copies grids with the specified times

TRAnge= *btim etim inc* copies grids in the time range of *btim* through *etim,* incremented by *inc* (no default for *btim*; *etim* default=*btim*; *inc* default=1, meaning one hour)

Derived Grid Keywords

DERive= derives a grid of the specified parameter, for example, DST or VOR; searches for the component grids using the search keywords, derives the grid, and copies it to the *ddataset*; do not use with the PARAM keyword; see the Remarks

MERidional= **YES** adds a correction term to account for the convergence of longitude lines at the poles

NO does not add a correction term; only valid with DERIVE=ABV, DSH, DST, DVG, or VOR; see the Remarks (default)

PLAnet= planet for which derive calculations are done; valid options are MERCURY, VENUS, EARTH, MARS, JUPITER, SATURN, NEPTUNE, URANUS (default=EARTH)

Mathematical Operations Keywords

G*n*= '*clause1*; *clause2*; . .; *clausen*'
grids to be used with the MATH keyword; separate select clauses with semicolons; single quotes are mandatory; specify *clause* using the following format:

searchkeyword value

valid options for *searchkeyword* are DAY, FHOUR, GPRO, GRID, LEV, PARAM, SRC, TIME; you can specify numbers between 1 and 99 in the keyword name (G1, G2, . . , G99); see the Remarks

INFo= '*text*' string placed in the header of the copied grid when the MATH keyword is used; maximum of 48 characters

MATh= '*expression*' mathematical operation to perform on the grids specified with the G*n* keywords; single quotes are mandatory; see the Remarks

NEWpar= *param punit level lunit src*
header information for the grid copied with the MATH keyword; parameter and level values, and associated units

param grid parameter; four characters maximum (default=MATH)

punit units of grid parameter; four characters maximum (default=NONE)

level grid level (default=from grid)

lunit units of grid level; two characters maximum (default=from grid)

src grid source; four characters maximum (default=from grid)

Output Keywords

BYTecount= **YES** lists the number of bytes received from the server

NO does not list the number of bytes (default)

DEL= **YES** delete the destination grid file before copying; see the Remarks

NO do not delete the destination grid file (default)

DGRid= beginning grid number in the destination grid file to place the copied grids; this keyword may overwrite existing grids; see the Remarks (default=append after the last grid in the grid file)

MAXgrd= maximum number of grids the newly created destination grid file can store; see the Remarks (default=1000)

NUM= number of grids matching the criteria defined with the search keywords to copy; for example, if you specify NUM=50, the first 50 grids matching the search keywords are copied (default=1)

ALL copies all grids matching the search keywords

SUBsect= *slat nlat elon wlon incrow inccol*

geographic region to subsect from the source grid, and row and column reduction intervals to use within the subsect region; see the Remarks

slat, nlat southern and northern latitudes of the subsect region

elon, wlon eastern and western longitudes of the subsect region (*slat*, *nlat*, *elon*, *wlon* defaults=source grid bounds)

incrow, inccol row and column reduction intervals; 1 means copy every row/column of grid points in the subsect region, 2 means copy every other row/column, etc.; when subsecting a conformal projection grid, the *incrow* and *inccol* values must be the same (default=1 for both)

Note: If the source grid is a McIDAS grid, you can subsect it by row and column instead of latitude and longitude. To do so, specify row and column numbers instead of latitudes and longitudes in the first four parameters, and specify ROWCOL as the seventh parameter. For example, SUBSECT=10 40 80 160 1 2 ROWCOL uses rows 10, 11, 12, ..., 40 and columns 80, 82, 84, ..., 160 for the subsect region. (If you don't include the ROWCOL at the end, it will treat 10 and 40 as latitudes, and 80 and 160 as longitudes.)

TITLE= *'text'* title of the newly created destination grid file; 32 characters maximum; see the Remarks

---

Remarks

Individual grids are stored in grid files. Each dataset position points to a single grid file. GRDCOPY copies grids from one or more grid files in the source dataset to a single grid file in the destination dataset.

If you specify DEL=YES or a position number of a nonexistent grid file in the *ddataset* parameter, the grids are copied into a newly created grid file with the attributes specified in keywords MAXGRD and TITLE.

Using the DGRID keyword may result in grids being overwritten in the destination grid file. For example, if the destination grid file has 50 grids (in grids 1 to 50) and you specify NUM=10 DGRID=25, the new grids will overwrite grids 25 to 34 in the destination grid file.

The GRID keyword copies grids from the grid file specified with the *sdataset position* number; the default *position* value, ALL, is not valid. When using the GRID keyword, do not specify additional search keywords, PARAM, LEV, DAY, etc., or NUM.

If you specify PARAM=STREAML, WINDB or WINDV, GRDCOPY does not search for grids with that parameter. Instead, it locates the u-component grids that match the other search keywords, then locates the matching v-component grids, and copies both sets of grids to the destination dataset. These grids may be used to draw streamlines, plot wind barbs or plot wind vectors with the GRDDISP command. If you specify any other parameter with the PARAM keyword, the value entered is the value searched for in the grid's parameter field.

Use command GRDLIST to preview the grids to be copied with GRDCOPY.

Use the SUBSECT keyword to copy a portion of the source grid. This keyword is useful because the subsected grid is smaller and can be copied more quickly than the source grid.

Use the DERIVE keyword to create grids of a variety of common meteorological parameters. The valid options are defined in the table below. In these equations, the following variables appear often:

u = u-component of the wind

v = v-component of the wind

x = grid point distance in the east-west direction

y = grid point distance in the north-south direction

DERIVE=	Description	Equation
ABV	absolute vorticity	$\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} + f$ *f* =coriolis parameter: 2ΩsinΦ (see COR)
BETA	beta parameter	$\frac{\partial f}{\partial y} = \frac{2\Omega \cos\phi}{a}$ *f* =coriolis parameter: 2ΩsinΦ (see COR) Ω=angular speed of the rotation of the planet (7.292×10^{-5} radians/second for Earth) Φ=latitude in degrees *a*=radius of the planet
COR	coriolis parameter	2ΩsinΦ Ω=angular speed of the rotation of the planet (7.292×10^{-5} radians/second for Earth) Φ=latitude in degrees
DIR	wind direction	atan2(-*u*,-*v*)
DSH	shearing deformation	$\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}$
DST	stretching deformation	$\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}$
DVG	divergence	$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$
SPD	wind speed	$\sqrt{u^2 + v^2}$
TD	dew point temperature	$\left[\frac{1}{T} - \frac{Rv \times \ln RH}{2500800 - 2397.5 \times (T - 273.15)}\right]^{-1}$ *T* =temperature in Kelvin *RH* =relative humidity *Rv*=moist gas constant: 461.5 Joules per kilogram per degree Kelvin
VOR	relative vorticity	$\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$

Use the MERIDIONAL keyword to correct for the convergence of longitude lines at the poles. When deriving divergence and stretching deformation grids (DVG and DST), specify MERIDIONAL=YES to subtract the following correction term from the calculation.

$$v \times \tan\left(\frac{\Phi}{r}\right)$$

When deriving vorticity, absolute vorticity, or shear deformation grids (VOR, ABV, or DSH), specify MERIDIONAL=YES to add the following correction term to the calculation.

$$u \times \tan\left(\frac{\Phi}{r}\right)$$

For both the correction terms above:

u = u-component of the wind

v = v-component of the wind

Φ = latitude in degrees

r = radius, in kilometers, of the planet at Φ latitude

The MERIDIONAL keyword has no effect with other DERIVE keyword options.

Use the G*n* keywords to specify the grids to be used with the MATH keyword. Each G*n* keyword specifies a single grid. G*n* is followed by a list of select clauses in single quotes. By default, subsequent G*n* keywords have the same select clauses, unless specified differently. For example, to request 850 and 1000 mb height grids from the 0:00 UTC GFS run, specify

G1='LEV 850;PARAM Z;TIME 0;SRC GFS' G2='LEV 1000'

The MATH keyword defines the operation to perform on the specified grids, for example, MATH='G1**(SQRT(G2))'. The table below shows the operations available with the MATH keyword.

MATH= option	Function
+	addition
–	subtraction
*****	multiplication
/	division
****	power
SQRT	square root
EXP	exponential (the number e raised to a power)
LOG	natural logarithm
LOG10	base 10 logarithm
SIN	sine
COS	cosine
TAN	tangent
ASIN	arcsine
ACOS	arccosine
ATAN	arctangent
ABS	absolute value
MIN	minimum of two grids
MAX	maximum of two grids
DDX	derivative with respect to *x*
DDY	derivative with respect to *y*
DELSQ	laplacian
COR	coriolis parameter ($2\Omega\sin\Phi$)
BETA	beta parameter $\left(\frac{\partial f}{\partial y}\right)$
LAT	creates a grid of latitude
LON	creates a grid of longitude

When specifying powers of ten, such as 10^2 or 10^{-3} in the MATH keyword, use one of the two methods shown in the examples below.

MATH='2*7.292*(1e-5)*(SIN(G1))'

MATH='2*7.292*(10(-5))*(SIN(G1))'**

In these entries both 1e-5 and 10**(-5) represent 10^{-5}. You must include the parentheses with 10**(-5); 10**-5 will not be interpreted correctly.

When using the MATH keyword and the range of output data values is larger than five orders of magnitude, the output grid is scaled based on the maximum end of the range. This means that small data values may be replaced by zeros in the output grid.

Grids created using GRDCOPY and those created using IGG MAKE with the DERIVE or MATH keyword may not yield numerically identical results. However, the values for both are correct. The differences are the result of one of the following:

- Grids created using GRDCOPY will be in Standard International units, whereas grids created using IGG MAKE may not be. For example, a grid created using the TADV (temperature advection) function in IGG MAKE will be in degrees per day, while a similar grid created using GRDCOPY with the MATH keyword will be in degrees per second unless a factor is added to the equation to compute it in degrees per day.
- By default, IGG MAKE applies a smoothing function and, if applicable, a meridional correction term to the data. GRDCOPY does not do this by default. You can override the IGG MAKE defaults by specifying SMO=NO and ADD=NO.

Examples

GRDCOPY NCEP/GFS MYDATA/GRIDS.1

This entry copies the first grid in the first grid file in dataset NCEP/GFS to the grid after the last grid in position 1 of dataset MYDATA/GRIDS.

GRDCOPY NCEP/GFS.5 MYDATA/GRIDS.2 LEV=850 NUM=10 DGRID=5

This entry copies the first 10 grids with a level of 850 mb in position 5 of dataset NCEP/GFS to grids 5 through 14 in position 2 of dataset MYDATA/GRIDS.

GRDCOPY NCEP/GFS.5 MYDATA/GRIDS.1 NUM=ALL DEL=YES TITLE='All grids in NCEP/GFS position 5'

This entry copies all grids in position 5 of dataset NCEP/GFS to position 1 of dataset MYDATA/GRIDS. The keyword DEL=YES deletes the existing destination grid file and creates a new file with the title *All grids in NCEP/GFS position 5* for the copied grids. Since the destination grid file is new, the copied grids begin in grid 1. Be careful when using NUM=ALL; grid files can contain thousands of grids. Use the search keywords to copy only the desired grids, as shown in the following examples.

GRDCOPY NMC/ETA.ALL G/SCRATCH.10 LEV=500 PARAM=Z T TD RH ABV DAY=#Y TIME=0 NUM=ALL

This entry copies all 500 mb height, temperature, dew point temperature, relative humidity and absolute vorticity grids from today's 00 UTC run in all grid files in dataset NMC/ETA to position 10 of dataset G/SCRATCH. The grids are appended after the last grid in the destination grid file.

GRDCOPY NMC/ETA.1 G/MODEL.2 LEV=TRO PARAM=Z T FHOUR=12 36 NUM=ALL DGRID=1

This entry copies all 12- and 36-hour forecast tropopause height and temperature grids in position 1 of dataset NMC/ETA to position 2 of dataset G/MODEL. The copied grids begin in grid 1 of the destination grid file.

GRDCOPY NMC/ETA.1 G/MODEL.2 LEV=TRO PARAM=Z T FRANGE=12 36 6 NUM=ALL DGRID=100

This entry copies all 12-, 18-, 24-, 30-, and 36-hour forecast tropopause height and temperature grids in position 1 of dataset NMC/ETA to position 2 of dataset G/MODEL. The copied grids begin in grid 100 of the destination grid file.

GRDCOPY NMC/ETA.1 G/MODEL.2 LEV=TRO PARAM=Z T FDAY=#Y FTIME=18 NUM=ALL

This entry copies all forecast tropopause height and temperature grids valid at 18 UTC today in position 1 of dataset NMC/ETA to position 2 of dataset G/MODEL. The grids are appended after the last grid in the destination grid file.

GRDCOPY MODEL.5 GRD.1 GRID=1 20 DGRID=1

This entry copies grids 1 through 20 in position 5 of the dataset with the alias name MODEL to grids 1 through 20 in position 1 of the dataset with the alias name GRD.

GRDCOPY GRIDS/GFS CASE/STUDY.1 DAY=#Y TIME=12 PARAM=U V LEV=200 SUBSECT=-20 20 -180 -120 1 2 NUM=ALL

This entry copies all 200 mb height u- and v-component grids from today's 12 UTC run in dataset GRIDS/GFS to position 1 of dataset CASE/STUDY. The copied grids contain every source grid point in the row direction and every other source grid point in the column direction in the region 20°S to 20°N and 180°E to 120°E.

GRDCOPY GG.1 GRD.4000 DERIVE=SPD LEV=500 FHOUR=48

The entry creates a 500 mb wind speed grid using component grids in position 1 of the dataset with the alias GG. The grids used to derive wind speed are the u- and v-components of the wind with 48-hour forecast times. The derived grid is copied to position 4000 of the dataset with the alias GRD.

GRDCOPY RTGRIDS/GFS LOCAL/MYGRIDS.10 DERIVE=VOR LEV=500 DAY=96284 TIME=12 FHOUR=12 24 36 48 NUM=ALL

This entry creates 500 mb relative vorticity grids using component grids from dataset RTGRIDS/GFS. The grids used to derive relative vorticity are the u- and v-components of the wind from day 96284 at 12:00 UTC with 12-, 24-, 36-, and 48-hour forecast times. The derived grids are copied to position 10 of the dataset LOCAL/MYGRIDS.

GRDCOPY GRD.8000 GRD.4000 G1='PARAM Z;LEV 850;FHOUR 48' G2='LEV 1000' MATH='G1-G2' NEWPAR=THCK

This entry finds two 48-hour forecast geopotential height grids in position 8000 of the dataset with the alias GRD. The 1000 mb grid is subtracted from the 850 mb grid and the result is copied to position 4000 of the dataset with the alias GRD. The copied grid is labeled with the new parameter name THCK and units NONE.

GRDCOPY GRD.4004 GRD.4004 G1='PARAM T;LEV 1000' G2='LEV 850' G3='PARAM U;LEV 1000' G4='LEV 850' NEWPAR=M DEG MATH='(COR(G1))/9.8*((G1+G2)/2)*((G3-G4)/(G1-G2))'

This entry creates a grid of frontal slope (Margules equation) from the 1000 and 850 mb temperature grids, and the 1000 and 850 mb u-component grids in position 4004 of the dataset with the alias GRD. The grid is copied into the same dataset, and labeled M with units DEG.

GRDCOPY G/G.4000 G/G.4000 G1='PAR T;LEV 1000;TIME 12; DAY 93070;SRC GFS' G2='PAR U;SRC ROI' G3='PAR V; SRC ROI'

MATH='-(G2*(DDX(G1))+G3*(DDY(G1)))' INFO='Temperature Advection at 1000 MB' NEWPAR=TADV K/S

This entry creates a grid of 1000 mb temperature advection using the 1000 millibar GFS model temperature grid, and the 1000 mb ROI model u- and v-component grids in position 4000 of the dataset G/G. The grid is copied into the same dataset, and labeled TADV with units of degrees Kelvin per second. The string *Temperature Advection at 1000 MB* is placed in the header of the new grid.

GRDDISP

Generates contours, streamlines or plots from grid data.

Format

GRDDISP *dataset frame [keywords]*

Parameters

dataset ADDE dataset name and absolute position; specify one of the following formats:

group/descriptor.position

alias.position

only positive integers or ALL (all positions) are valid for *position*; when a grid matching the conditions specified in the search keywords is found, the search stops (no default for *group/descriptor* or *alias*; default=ALL for *position*)

frame frame number to display the output; specify OPP to display the output on the frame opposite the current frame (default=current)

Search Keywords

DAY= displays grids with the specified day

ENSemble= *e1 . . en* displays grids with the specified ensemble numbers; each ensemble number must include a + or - sign, e.g., -1 -0 +0 +1

FDAy= displays grids with the specified forecast day; a grid's forecast day is determined by adding the forecast hour to the day and time; for example, a 12 UTC grid from day 95300 with a forecast hour of 72 has a forecast day of 95303; you cannot use FDAY with the FHOUR, FRANGE or MATH keyword

FHOur= *h1 . . hn* displays grids with the specified forecast hours; use with the keyword GRA to display the *h1* grid in the first frame, *h2* grid in the second frame, etc.; you cannot use FHOUR with the FRANGE, FDAY or FTIME keyword

FRANge= *bhr ehr inc* displays grids in the range of forecast hours *bhr* through *ehr*, incremented by *inc* hours; use with the keyword GRA to display the grids in forecast hour order (*bhr* to *ehr*); you cannot use FRANGE with the FHOUR, FDAY or FTIME keyword (no default for *bhr*; *ehr* default=*bhr*; *inc* default=1)

FTIme= displays grids with the specified forecast time; a grid's forecast time is determined by adding the forecast hour to the time; for example, a 12 UTC grid with a forecast hour of 18 has a forecast time of 6 on the following day; you cannot use FTIME with the FHOUR, FRANGE or MATH keyword

GPRo= displays grids with the specified projection, for example, MERC, PS, LAMB, EQUI

GRIB= *geo par model level*

displays grids with the specified GRIB codes; the four values are the geographic, parameter, model and level codes that can be listed with **GRDLIST FORM=ALL**; values specified as an X (a placeholder) or not specified at all will match any GRIB code

GRId= displays the specified grid or last grid (LAST) in the grid file specified in the dataset position number; when using this keyword, the default *position* value in *dataset* (ALL) is not valid and all other search keywords (PARAM, LEV, DAY, etc.) are ignored

LEV= *lev[units]* displays grids with the specified level and units, e.g., **SFC, 850[MB], 5000[M]**; units are optional but must be in square brackets if specified

PARam= *param* displays grids with the specified grid parameter, for example, T, Z, RH; see the Remarks

MOVIE draws streamlines with colored line segments that can be animated using the CM command

STREAML draws streamlines using u- and v-component grids

WINDB plots wind barbs using u- and v-component grids

WINDV plots wind vectors using u- and v-component grids

SRC= *s1 . . sn* displays grids with the specified sources, for example, MDX, GFS, ETA

TIMe= displays grids with the specified time

Multiple Contour/Plot Keywords

GRA= *bfra efra* generates contours or plots for the grids matching the criteria defined with search keywords in frames *bfra* through *efra*; the grids are displayed by forecast hour in the order specified with the FHOUR keyword or sequentially from the range *bhr* to *ehr* in FRANGE; you cannot specify both GRA and NUM (*bfra* default=current; *efra* default=*bfra*)

NUM= number of grids matching the criteria defined with the search keywords to contour or plot, beginning on the current frame; the grids are not sorted, they are displayed in the order they are found in the grid files; you must specify FHOUR or FRANGE with NUM; you cannot specify both NUM and GRA

Derived Grid Keywords

DERive= displays a derived grid of the specified parameter, for example, DST or VOR; searches for the component grids using the search keywords; do not use with the PARAM keyword; see the Remarks

MERidional= **YES** adds a correction term to account for the convergence of longitude lines at the poles

NO does not add a correction term; only valid with DERIVE=ABV, DSH, DST, DVG, or VOR; see the Remarks (default)

PLAnet= planet for which derive calculations are done; valid options are MERCURY, VENUS, EARTH, MARS, JUPITER, SATURN, NEPTUNE, URANUS (default=EARTH)

Mathematical Operations Keywords

G*n*= '*clause1*; *clause2*; . .; *clausen*'

grids to be used with the MATH keyword; separate select clauses with semicolons; single quotes are mandatory; specify *clause* using the following format:

searchkeyword value

valid options for *searchkeyword* are DAY, FHOUR, GPRO, GRID, LEV, PARAM, SRC, TIME; you can specify numbers between 1 and 99 in the keyword name (G1, G2, . . , G99); see the Remarks

MATh= '*expression*' mathematical operation to perform on the grids specified with the G*n* keywords; single quotes are mandatory; see the Remarks

NEWpar= *param punit level lunit*

title information for the plot or contour displayed when the MATH keyword is used

param grid parameter; four characters maximum (default=MATH)

punit units of grid parameter; four characters maximum (default=NONE)

level grid level (default=from grid)

lunit units of grid level; two characters maximum (default=from grid)

Navigation Keywords

LAT= *lat1 lat2* latitudes of the points at the lower-right and upper-left corners of the map; you cannot use LAT with the NAV or MAP keyword

LON= *lon1 lon2* longitudes of the points at the lower-right and upper-left corners of the map; you cannot use LON with the NAV or MAP keyword

MAP= map for the contour or plot; see the MAP command for a list of valid predefined maps, for example, MID, USA, WORL; you cannot use MAP with the NAV, LAT or LON keyword (default=grid extents)

MCOlor= graphics color level of map (default = 1)

NAV= draw the contour or plot using the navigation of the specified frame; no map is drawn; you cannot use NAV with the MAP, LAT or LON keyword

num frame number to use for navigation

C use the current frame's navigation

F use the navigation of the frame where the contour or plot is drawn

PRO= **CONF** polar stereographic or Lambert conformal projection

MERC Mercator projection (default=grid's projection)

SLAt= *lat1 lat2* standard latitudes for a CONF projection; two values for a general Lambert conformal projection; one value for a polar stereographic projection (default=from grid header when the LAT and LON, or MAP keywords are not used; 60 60 for northern hemisphere maps defined with the LAT and LON, or MAP keywords; -60 -60 for southern hemisphere maps defined with the LAT and LON, or MAP keywords)

SLOn= standard longitude for a CONF projection (default=computed)

Contour and Plot Keywords

COLor= graphics color level (default=2 for contour, 3 for plot); when plotting data (with OUT=PLOT or PAR=WINDB or PAR=WINDV), the value can be specified as a single number, e.g., COLOR=5, or in the format *color*[*min-max*B*inc*] to assign multiple colors for the parameter, based on its values; see the Remarks for details

ELE= *bele eele* beginning and ending frame elements for output; specify MAX for maximum element number (default=1 MAX when using the NAV keyword; default=20 MAX-20 when using the MAP keyword)

FORMAT= *designator 'text'*

output format for grid point values or contour labels, and associated text; valid formats are I, F, or E; if multiple parameters are requested, FORMAT cannot be specified; see the Remarks (*designator* default=from *dataset*; no *text* default)

LINE= *blin elin* beginning and ending frame lines for output; specify MAX for maximum line number (default=1 MAX-57 when using the NAV keyword; default=20 MAX-57 when using the MAP keyword)

LSIze= height, in pixels, of the plot characters or contour labels; the range is 0-50 where 0 omits the labels (default=8)

OUT= **CON** draws contours using the grid data (default)

PLOT plots the grid points

POWer= power of 10 to multiply the grid point values by for display (default=0)

TDAy= **CAL** display calendar day on the default title; not valid if the ELE, LINE or PAN keyword is used

JUL display the Julian day on the default title (default)

TITLE= *'text' col line ele size*

text, color, location and size for the contour or plot title

'text' 72 characters maximum; specify X without quotes for the default title (default=computed)

col graphics color level of the title (default=1 if PARAM=MOVIE, otherwise default=value from the COLOR keyword)

line beginning TV line to write the title (default=below contour or plot)

ele beginning TV element to write the title (default=below contour or plot)

size height of the title text in pixels (default=8)

UNIt= units for the contour or plot; see the Remarks (default=from the grid header)

Contour-Specific Keywords

CINt= *num* contour interval; specify 0 to let GRDDISP compute a reasonable interval; specify your own interval as an integer or decimal value greater than zero; see the Remarks for the default value

string contour values; specify a string name that contains up to 64 values; e.g., CINT=SNOWDEPTH will draw contours at the values specified in the string named SNOWDEPTH

DASh= **ALL** dash all contours

NEG dash negative contours only

POS dash positive contours only

(default=solid contours)

LINT= contour label interval (default=1, meaning label every contour)

MOVie= *bcol ecol bspd espd blen elen*

change the defaults for the PARAM=MOVIE output

bcol beginning graphics color level for line segments (default=2)

ecol ending graphics color level for line segments (default=maximum color level)

bspd beginning wind speed represented by *blen*, units are from the grid data (default=computed)

espd ending wind speed represented by *elen*, units are from the grid data (default=computed)

blen beginning length of line segments (default=2)

elen ending length of line segments (default=40)

SMOoth= contour smoothing factor; the range is 0 through 50; larger numbers produce smoother contours; not valid with PARAM=MOVIE or PARAM=STREAML (default=20)

Plot-Specific Keyword

PINt= *row col* row and column intervals of grid points to plot with OUT=PLOT, PARAM=WINDB or PARAM=WINDV; for example, PINT=4 3 plots every fourth grid point in the row direction and every third grid point in the column direction (*row* default=1; *col* default=*row*)

Output Keywords

BYTecount= **YES** lists the number of bytes received from the server

NO does not list the number of bytes (default)

SUBsect= *slat nlat elon wlon incrow inccol*

geographic region to subsect from the source grid, and row and column reduction intervals to use within the subsect region; see the Remarks

slat, nlat southern and northern latitudes of the subsect region

elon, wlon eastern and western longitudes of the subsect region (*slat*, *nlat*, *elon*, *wlon* defaults=region slightly larger than that defined by the frame navigation or map specified with MAP or LAT and LON keywords; if NAV=F is specified, the grid is not subsected)

incrow, inccol row and column reduction intervals; 1 means use every row/column of grid points in the subsect region, 2 means use every other row/column, etc.; when subsecting a conformal projection grid, the *incrow* and *inccol* values must be the same (default=1 for both)

Note: If the source grid is a McIDAS grid, you can subsect it by row and column instead of latitude and longitude. To do so, specify row and column numbers instead of latitudes and longitudes in the first four parameters, and specify ROWCOL as the seventh parameter. For example, SUBSECT=10 40 80 160 1 2 ROWCOL uses rows 10, 11, 12, ..., 40 and columns 80, 82, 84, ..., 160 for the subsect region. (If you don't include the ROWCOL at the end, it will treat 10 and 40 as latitudes, and 80 and 160 as longitudes.)

Remarks

Individual grids are stored in grid files. Each dataset position points to a single grid file. If you know which grid file contains the grid you want to display, specify its position in the *dataset* parameter to reduce the searching time.

The GRDDISP command draws contours or streamlines, or plots data from the first grid or set of grids that matches the conditions specified with the search keywords. If you don't specify any search keywords, GRDDISP displays the first grid in the dataset. If, for example, you specify PARAM=T LEV=850 TIME=12 SRC=ETA, GRIDDISP displays the first grid with T in its parameter field, 850 in its level field, 12 in its time field, and ETA in its source field. To choose the exact grid you want, you may need to specify several search keywords.

Use the keyword FHOUR or FRANGE with GRA or NUM to generate a contour or plot using more than one grid with a single GRDDISP command. The keyword GRA displays the grids in the order entered with FHOUR or sequentially from the range *bhr* to *ehr* in FRANGE. For example, if you specify SRC=GFS TIME=12 PARAM=Z LEV=500 FHOUR=0 36 24 60 GRA=11 14, the 12 UTC GFS 500 mb height grids with forecast hours 0, 36, 24 and 60 are displayed in frames 11, 12, 13 and 14.

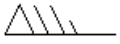
The default contour interval for a grid is taken from the text file **CONTOUR.DEF**. If no file entry matches the grid's parameter, units and level, the default is calculated to produce a reasonable number of contours, usually between 10 and 20. To change, add or delete entries in **CONTOUR.DEF**, first back up the current version, then modify it using the format described near the top of the file.

If you specify PARAM=MOVIE, STREAML, WINDB or WINDV, GRDDISP does not search for a grid with that parameter. Instead, it locates the first u-component grid that matches the other search keywords, then it locates the matching v-component grid and uses those grids to contour streamlines, plot wind barbs, or plot wind vectors. In all other cases, the value entered in the keyword PARAM is the value searched for in the grid's parameter field.

When plotting data (with OUT=PLOT or PAR=WINDB or PAR=WINDV), you can assign output plot color based on the parameter's values by using the COLOR keyword format *color*[*min*-*max*B*inc*], where *color* is the beginning graphics color level, *min* and *max* indicate the range of parameter values to color-code, and *inc* is the increment (the dash means "to" and the B means "by"). For example, when plotting integer values, COLOR=2[30-59B10] plots values 30-39 in level 2, values 40-49 in level 3, and values 50-59 in level 4. Values less than 30 are in level 1, and values greater than 59 are in level 5. When plotting floating point values (e.g., FORMAT=F5.2), COLOR=2[30-59.99B10] plots values 30.00-39.99 in level 2, values 40.00-49.99 in level 3, and values 50.00-59.99 in level 4. Values less than 30.00 are in level 1, and values greater than 59.99 are in level 5. Also, when plotting color-coded wind barbs (PAR=WINDB), the units of the 'min' and 'max' values are knots, even if you specify UNIT=MPS.

Use the SUBSECT keyword to analyze a portion of the original grid. The analysis can be performed slightly faster because there is less data to transfer and analyze.

Wind barb symbols use triangular flags, and long and short lines, or barbs, to represent wind speed. For example, the wind barb shown here represents a wind speed of 75 knots or 37.5 meters/second.



- a flag represents 50 knots or 25 meters/second

- a long barb represents 10 knots or 5 meters/second
- a short barb represents 5 knots or 2.5 meters/second.

When using the UNIT keyword, you can specify the following units:

Unit type	UNIT=	Definition
Distance	**CM**	centimeters
	DM	decameters
	FT	feet
	GPM	geopotential meters
	IN	inches
	KM	kilometers
	M	meters
	MI	miles
	MM	millimeters
	NMI	nautical miles
	YD	yards
Pressure	**HPA**	hectopascals
	INHG	inches of mercury
	MB	millibars
Speed	**KT or KTS**	knots
	MPH	miles per hour
	MPS	meters per second
Temperature	**C**	Celsius
	F	Fahrenheit
	K	Kelvin

Use the DERIVE keyword to display grids of a variety of common meteorological parameters. The valid options are defined in the table below. In these equations, the following variables appear often:

u = u-component of the wind

v = v-component of the wind

x = grid point distance in the east-west direction

y = grid point distance in the north-south direction

DERIVE=	Description	Equation
ABV	absolute vorticity	$\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} + f$ f =coriolis parameter: 2ΩsinΦ (see COR)
BETA	beta parameter	$\frac{\partial f}{\partial y} = \frac{2\Omega cos\phi}{a}$ f =coriolis parameter: 2ΩsinΦ (see COR) Ω=angular speed of the rotation of the planet (7.292×10^{-5} radians/second for Earth) Φ=latitude in degrees a=radius of the planet
COR	coriolis parameter	2ΩsinΦ Ω=angular speed of the rotation of the planet (7.292×10^{-5} radians/second for Earth) Φ=latitude in degrees
DIR	wind direction	atan2($-u$,$-v$)
DSH	shearing deformation	$\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}$
DST	stretching deformation	

		$\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}$
DVG	divergence	$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$
SPD	wind speed	$\sqrt{u^2 + v^2}$
TD	dew point temperature	$\left[\frac{1}{T} - \frac{Rv \times \ln RH}{2500800 - 2397.5 \times (T - 273.15)}\right]^{-1}$ T =temperature in Kelvin RH =relative humidity Rv=moist gas constant: 461.5 Joules per kilogram per degree Kelvin
VOR	relative vorticity	$\frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}$

Use the MERIDIONAL keyword to correct for the convergence of longitude lines at the poles. When deriving divergence and stretching deformation grids (DVG and DST), specify MERIDIONAL=YES to subtract the following correction term from the calculation.

$$v \times \tan\left(\frac{\Phi}{r}\right)$$

When deriving vorticity, absolute vorticity, or shear deformation grids (VOR, ABV, or DSH), specify MERIDIONAL=YES to add the following correction term to the calculation.

$$u \times \tan\left(\frac{\Phi}{r}\right)$$

For both the correction terms above:

u = u-component of the wind

v = v-component of the wind

Φ = latitude in degrees

r = radius, in kilometers, of the planet at Φ latitude

The MERIDIONAL keyword has no effect with other DERIVE keyword options.

Use the G*n* keywords to specify the grids to be used with the MATH keyword. Each G*n* keyword specifies a single grid. G*n* is followed by a list of select clauses in single quotes. By default, subsequent G*n* keywords have the same select clauses, unless specified differently. For example, to request 850 and 1000 mb height grids from the 0:00 UTC GFS run, specify

G1='LEV 850;PARAM Z;TIME 0;SRC GFS' G2='LEV 1000'

The MATH keyword defines the operation to perform on the specified grids, for example, MATH='G1**(SQRT(G2))'. The table below shows the operations available with the MATH keyword.

MATH= option	**Function**
+	addition
-	subtraction
*	multiplication
/	division
**	power
SQRT	square root
EXP	exponential (the number e raised to a power)
LOG	natural logarithm
LOG10	base 10 logarithm
SIN	sine
COS	cosine
TAN	tangent

ASIN	arcsine
ACOS	arccosine
ATAN	arctangent
ABS	absolute value
MIN	minimum of two grids
MAX	maximum of two grids
DDX	derivative with respect to *x*
DDY	derivative with respect to *y*
DELSQ	laplacian
COR	coriolis parameter ($2\Omega\sin\Phi$)
BETA	beta parameter
LAT	creates a grid of latitude
LON	creates a grid of longitude

When specifying powers of ten, such as 10^2 or 10^{-3} in the MATH keyword, use one of the two methods shown in the examples below.

MATH='2*7.292*(1e-5)*(SIN(G1))'

MATH='2*7.292*(10(-5))*(SIN(G1))'**

In these entries both 1e-5 and 10**(-5) represent 10^{-5}. You must include the parentheses with 10**(-5); 10**-5 will not be interpreted correctly.

When using the MATH keyword and the range of output data values is larger than five orders of magnitude, the output grid is scaled based on the maximum end of the range. This means that small data values may be replaced by zeros in the output grid.

Use the FORMAT keyword options below to format output based on FORTRAN designators for data output. To add text to the numeric output, follow the designator with a text in single quotes.

Designator	**Description**
I	integer
F	floating point
E	exponential

To specify a parameter's output format, use a designator followed by a positive number. For example, I4 specifies an integer of up to four digits; F10.1 specifies a floating point decimal of up to 10 digits (including the decimal point) with one digit to the right of the decimal; E10.3 specifies an exponential number of up to 10 characters (including the decimal point and the four-character exponential flag) with three digits to the right of the decimal. Specify an X to use the parameter's default format. If the specified integer or floating point format is too small for the parameter's values, a series of asterisks (***) is displayed.

Examples

GRDDISP G/LOCAL

This entry draws a contour using the first grid in the first grid file in dataset G/LOCAL. The map and contours cover the grid's entire domain. The contour interval is taken from the text file **CONTOUR.DEF** if it has an entry that matches the grid's parameter, units and level. If **CONTOUR.DEF** does not have a matching entry, GRDDISP calculates the contour interval.

GRDDISP G/LOCAL.9 GRID=5 OUT=PLOT PINT=1 3 NAV=C

This entry plots grid point values from grid 5 in the grid file in position 9 of dataset G/LOCAL. Every grid point in the row direction and every third grid point in the column direction is plotted using the navigation of the current frame.

GRDDISP NMC/ETA LEV=1000 PARAM=WINDV DAY=#Y TIME=0 FDAY=#Y FTIME=18 MAP=MID PRO=MERC TITLE='ETA 1000 mb wind vectors, valid at 18 UTC today'

This entry plots the 1000 mb wind vector forecast valid at 18 UTC today over the Midwest. The vectors are created using the u- and v-component grids from today's 00 UTC model run in dataset NMC/ETA. The default title below the plot is replaced with the text *ETA 1000 mb wind vectors, valid at 18 UTC today*.

GRDDISP NCEP/GFS LEV=500 PARAM=Z CINT=60 FHOUR=24 LAT=-40 -40 LON=-90 90 SLAT=-40 PRO=CONF

This entry draws a contour of the 24-hour forecast 500 mb heights with an interval of 60 over the South Pole. The data is taken from the first 24-hour forecast 500 mb height grid in dataset NCEP/GFS.

GRDDISP NCEP/GFS LEV=500 PARAM=Z CINT=60 DAY=#Y TIME=12 FRANGE=0 120 24 GRA=1 6

This entry draws a contour of the 0-, 24-, 48-, 72-, 96- and 120-hour forecast 500 mb heights with an interval of 60. The contours are displayed over a map of the grids' domain in frames 1, 2, 3, 4, 5 and 6. The data is taken from today's 12 UTC model run grids in dataset NCEP/GFS.

GRDDISP MODEL.2 LEV=850 PARAM=T UNIT=C CINT=5 SRC=GFS DASH=NEG COLOR=5

This entry draws a contour of the first 850 mb GFS temperature grid in the grid file in position 2 of the dataset with the alias name MODEL. The map and contours cover the grid's entire domain. The contours are drawn in graphics color level 5 with an interval of 5 degrees Celsius. Negative contours are dashed.

TE SNOW "0 10 20 32 40 60
GRDDISP NMC/ETA LEV=850 PARAM=T UNIT=F CINT=SNOW MAP=USA PRO=CONF

The TE command entry creates a string called SNOW containing contour values for the GRDDISP entry. The GRDDISP entry draws a contour of the first 850 mb temperature grid in dataset NMC/ETA. The map and contours cover the USA, and only the 0, 10, 20, 32, 40 and 60 degree Fahrenheit contours are drawn. Do not specify a pound sign (#) before the string name in the CINT keyword.

GRDDISP GRIDS/GFS LEVEL=SFC PARAM=T DAY=97300 FHOUR=24 MAP=USA SUBSECT=25 50 60 100

This entry draws 24-hour forecast surface temperature contours from the dataset GRIDS/GFS. A map of the United States is drawn, but the contours are drawn only for the geographic region specified in the SUBSECT keyword (25-50°N and 60-100°W). If the SUBSECT keyword was not specified, GRDDISP would subsect a region slightly larger than the map and draws contours in the entire frame.

GRDDISP GRD.8000 4 LAT=20 50 LON=75 145 PRO=CONF DERIVE=SPD LEV=500 FHOUR=48

This entry draws a 500 mb wind speed contour with a 48-hour forecast time. The component grids used to derive wind speed are the u- and v-components of the wind from position 8000 in the dataset with the alias GRD. The contour is displayed on frame 4 over a polar stereographic map between 20° and 50° latitude and 75° and 145° longitude.

GRDDISP RTGRIDS/GFS MAP=USA PRO=MERC DERIVE=VOR LEV=500 DAY=96284 TIME=12 FHOUR=12 24 36 48 GRA=1 4 DASH=NEG

This entry draws contours for a sequence of 500 mb relative vorticity grids from day 96284 at 12:00 UTC. The component grids are from dataset RTGRIDS/GFS. The 12-, 24-, 36-, and 48-hour forecast contours are displayed on frames 1, 2, 3, and 4, and displayed over the USA in a Mercator projection. All negative contours are dashed.

GRDDISP RTGRIDS/GFS PARAM=MOVIE LEV=500 DAY=#Y TIME=0 FHOUR=18 MAP=USA MOVIE=2 8
TE MOVIECOLORS "GU MAKE !1 0 !2 0
REPEAT MOVIECOLORS 2 TO 8 BY 1 0 BY 40
CM 2 8 GO=YES

The GRDDISP entry draws the 18-hour GFS forecast 500 mb streamlines with colored line segments on a map of the U.S. The TE and REPEAT entries set graphics color levels 2 to 8 to increase linearly from black (level 2) to bright green (level 8). The CM entry then circulates the graphics color levels to animate the wind flow.

GRDDISP RTGRIDS/NAM-USLC DAY=#Y TIME=12 FHOUR=12 PAR=T LEV=SFC UNIT=F MAP=USA COLOR=3[10-69B10] OUT=PLOT FORMAT=I4 PINT=2 3

This entry plots the 12-hour forecast surface temperatures over the United States from today's 12 UTC NAM model run. The temperatures are color coded: less than 10°F in graphics color level 2, 10-19°F in level 3, 20-29°F in level 4, 30-39°F in level 5, 40-49°F in level 6, 50-59°F in level 7, 60-69°F in level 8, and 70°F and greater in level 9.

GRDDISP RTGRIDS/GFS-GLME DAY=#Y TIME=0 FHOUR=24 PAR=WINDB LEV=700 UNIT=KTS MAP=NA COLOR=2[1-20B5] PINT=3 3

This entry plots the 24-hour forecast 700 mb wind barbs over North America from today's 00 UTC GFS model run. The flags are color coded by speed: less than 1 knot in graphics color level 1, 1-5 knots in level 2, 6-10 knots in level 3, 11-15 knots in level 4, 16-20 knots in level 5, and greater than 20 knots in level 6. Note that each flag symbol may appear in two different colors because the speeds are rounded to the closest five knot increment (i.e., 2.5-7.4 knot speeds are displayed as 5 knot flags, 7.5-12.4 knot speeds are displayed as 10 knot flags, 12.5-17.4 knot speeds are displayed as 15 knot flags, etc.). Thus, 5 knot flags can be in graphics color level 2 or 3, 10 knot flags can be in level 3 or 4, 15 knot flags can be in level 4 or 5, etc.

GRDIMG

Converts a grid to a digital image.

Format

GRDIMG *sdataset ddataset [keywords]*

Parameters

sdataset source ADDE grid dataset name and absolute position; specify one of the following formats:

group/descriptor.position
alias.position

use only positive integers or ALL (all positions) for *position* (no default for *group/descriptor* or *alias*; default=ALL for *position*)

ddataset destination ADDE image dataset

Search Keywords

DAY= selects first grid with this day

DRAnge= *bday eday inc*

selects the first grid in the range of days *bday* through *eday* incremented by *inc* days (no default for *bday*; *eday* default=*bday*; *inc* default=1)

ENSemble= *e1 . . en* selects grids with the specified ensemble numbers; each ensemble number must include a + or - sign, e.g., -1 -0 +0 +1

FDAy= selects the first grid with this forecast day; a grid's forecast day is determined by adding the forecast hour to the day and time, e.g., a 12 UTC grid from day 2000100 with a forecast hour of 72 has a forecast day of 2000103; you cannot use FDAY with the FHOUR or FRANGE keyword

FHOur= selects the first grid with this forecast hour; you cannot use FHOUR with the FRANGE, FDAY or FTIME keyword

FRANGe= *bhr ehr inc*

selects the first grid in the range of forecast hours *bhr* through *ehr* incremented by *inc* hours (no default for *bhr*; *ehr* default=*bhr*; *inc* default=1); you cannot use FRANGE with the FHOUR, FDAY or FTIME keyword

FTIme= selects the first grid with this forecast time; a grid's forecast time is determined by adding the forecast hour to the time, e.g., a 12 UTC grid with a forecast hour of 18 has a forecast time of 6 on the following day; you cannot use FTIME with the FHOUR or FRANGE keyword

GPRo= selects the first grid with this projection; valid entries include MERC, PS, LAMB and EQUI

GRIB= *geo par model level*

selects the first grid with the specified GRIB codes; the four values are the geographic, parameter, model and level codes that can be listed with GRDLIST FORM=ALL; values specified as an X (a placeholder) or not specified at all will match any GRIB code

GRId= selects the grid with this grid number or LAST (for last grid); when using this keyword, all other search keywords are ignored and the default *position* value in *sdataset* (ALL) is invalid so you must specify a position number

LEV= *lev* **[***unit***]** selects the first grid with this level and *units*, e.g., SFC, 1000[MB], 5000[M]; units are optional but must be in square brackets if specified

PARam= selects the first grid with this parameter, e.g., T, Z, RH

SRC= *s1 . . sn* selects the first grid with any of these sources, e.g., MDX, GFS, ETA

TIMe= selects the first grid with this time

TRAnge= *btim etim inc*

selects the first grid in the range of times *btim* through *etim* incremented by *inc* hours (no default for *btim*; *etim* default=*btim*; *inc* default=1)

Output Keywords

MAGnify= *rmag cmag* grid row and column magnification factors; both values must be greater than 0; these values determine the number of lines and elements in the destination image; see the Remarks (default=1 *rmag*)

SCAle= *prodlo prodhi britlo brithi*

source grid's product value range to retain in the destination image, and the corresponding brightness value range to assign to the product value range; see the Remarks (default=*mingridvalue maxgridvalue* 0 255)

SUBsect= *slat nlat elon wlon incrow inccol*

geographic region to subsect from the source grid, and row and column reduction intervals to use within the subsect region, where:

slat, nlat southern and northern latitudes of the subsect region

elon, wlon eastern and western longitudes of the subsect region (*slat*, *nlat*, *elon*, *wlon* defaults=source grid bounds)

incrow, inccol row and column reduction intervals; 1 means use every row/column of grid points in the subsect region, 2 means use every other row/column, etc.; when subsecting a conformal projection grid, the *incrow* and *inccol* values must be the same (default=1 for both)

Note: If the source grid is a McIDAS grid, you can subsect it by row and column instead of latitude and longitude. To do so, specify row and column numbers instead of latitudes and longitudes in the first four parameters, and specify ROWCOL as the seventh parameter. For example, SUBSECT=10 40 80 160 1 2 ROWCOL uses rows 10, 11, 12, ..., 40 and columns 80, 82, 84, ..., 160 for the subsect region. (If you don't include the ROWCOL at the end, it will treat 10 and 40 as latitudes, and 80 and 160 as longitudes.)

UNIt= output units for the selected parameter (default=from grid header)

Remarks

By default, GRDIMG converts each source grid point to a single pixel in the destination image. Use the MAGNIFY keyword to make the image larger. The equations below calculate *rmag* and *cmag* values that will create an image large enough to fill the frame. Round the values up to the nearest integer since MAGNIFY accepts only integer values.

rmag = (number of lines in frame - 1) / (number of rows in grid - 1)
cmag = (number of elements in frame - 1) / (number of columns in grid - 1)

The values specified in the *prodlo*, *prodhi*, *britlo*, and *brithi* parameters in the SCALE keyword define a linear product-to-brightness calibration for displaying the destination image. Source grid product values outside of the specified *prodlo-prodhi* range are assigned a missing data value if the *britlo* value is greater than 0 and/or the *brithi* value is less than 255. They are assigned the appropriate *prodlo* or *prodhi* value if *britlo* is 0 and *brithi* is 255. Brightness levels are assigned for missing values using linear interpolation above and below the *prodlo* and *prodhi* values.

Examples

GRDIMG LOCAL/GRIDS.100 LOCAL/AREAS.10

This entry uses the data in the first grid in position 100 of dataset LOCAL/GRIDS to create the image in position 10 of dataset LOCAL/AREAS. The minimum and maximum grid values are assigned brightness values 0 and 255, respectively. Grid values between the minimum and maximum are assigned brightness values linearly interpolated between 0 and 255.

GRDIMG GRIDS/ETA IMAGES/ETA.1 PARAM=PRE LEV=SFC SCALE=1010 1020

This entry uses the data in the surface pressure grid in dataset GRIDS/ETA to create the image in position 1 of dataset IMAGES/ETA. Grid pressures between 1010 and 1020 are retained in their image product values and are assigned brightness levels linearly interpolated between 0 and 255. Grid pressures less than 1010 are assigned an image product value of 1010 and a brightness value of 0. Grid pressures greater than 1020 are assigned an image product value of 1020 and a brightness value of 255.

GRDIMG GRIDS/ETA IMAGES/ETA.1 PARAM=PRE LEV=SFC SCALE=1010 1020 100 200

This entry uses the data in the surface pressure grid in dataset GRIDS/ETA to create the image in position 1 of dataset IMAGES/ETA. Grid pressures between 1010 and 1020 are retained in their image product values and are assigned brightness levels linearly interpolated between 100 and 200. Grid pressures less than 1010 are assigned missing product values and brightness values linearly interpolated between 0 and 100. Grid pressures greater than 1020 are assigned missing product values and brightness values linearly interpolated between 200 and 255.

GRDIMG GRIDS/GLOBALGFS IMAGES/GFSUSA.4000 PARAM=T LEV=500 SUBSECT=25 50 65 130

This entry uses the data in the 500-mb temperature grid in dataset GRIDS/GLOBALGFS to create the image in position 4000 of dataset IMAGES/GFSUSA. Only the grid points within the latitude and longitude ranges specified in the SUBSECT keyword are used, so the destination image contains only that subset of the data in the source grid.

GRDINFO

Lists, modifies, or statistically analyzes grid point values.

Formats

GRDINFO *sdataset* **EDIT** *ddataset [keywords]*
GRDINFO *sdataset* **HIST** *[keywords]*
GRDINFO *sdataset* **LIST** *[keywords]*
GRDINFO *sdataset* **STAT** *[keywords]*

Parameters

sdataset source ADDE dataset name and absolute position; specify one of the following formats:

group/descriptor.position

alias.position

use only positive integers or ALL (all positions) for *position* (no default for *group/descriptor* or *alias*; default=ALL for *position*)

EDIT modifies grid point values and copies the new grid to *ddataset*

HIST displays histogram for grid point values

LIST lists grid point values

STAT displays statistical analysis for grid point values; statistics include the minimum and maximum values, mean, and standard deviation (default)

ddataset destination ADDE dataset name and absolute position; specify one of the following formats:

group/descriptor.position

alias.position

use only positive integers for *position*

Search Keywords

DAY= *d1 . . dn* selects grids with the specified days

DRAnge= *bday eday inc* selects grids in the range of days *bday* through *eday,* incremented by *inc* days (no default for *bday*; *eday* default=*bday*; *inc* default=1)

ENSemble= *e1 . . en* selects grids with the specified ensemble numbers; each ensemble number must include a + or - sign, e.g., -1 -0 +0 +1

FDAy= selects grids with the specified forecast day; a grid's forecast day is determined by adding the forecast hour to the day and time; for example, a 12 UTC grid from day 2000100 with a forecast hour of 72 has a forecast day of 2000103; you cannot use FDAY with the FHOUR, FRANGE or MATH keyword

FHOur= *h1 . . hn* selects grids with the specified forecast hours; you cannot use FHOUR with the FRANGE, FDAY or FTIME keyword

FRANGe= *bhr ehr inc* selects grids in the range of forecast hours *bhr* through *ehr,* incremented by *inc* hours; you cannot use FRANGE with the FHOUR, FDAY or FTIME keyword (no default for *bhr*; *ehr* default=*bhr*; *inc* default=1)

FTIme= selects grids with the specified forecast time; a grid's forecast time is determined by adding the forecast hour to the time; for example, a 12 UTC grid with a forecast hour of 18 has a forecast time of 6 on the following day; you cannot use FTIME with the FHOUR, FRANGE or MATH keyword

GPRo= *g1 . . gn* selects grids with the specified projection, for example, MERC, PS, LAMB, EQUI

GRIB= *geo par model level*

selects grids with the specified GRIB codes; the four values are the geographic, parameter, model and level codes that can be listed with GRDLIST FORM=ALL; values specified as an X (a placeholder) or not specified at all will match any GRIB code

GRId= *bgrid egrid* selects grids *bgrid* through *egrid* in the grid file specified in the *dataset* parameter; enter a grid number or LAST (last grid) for *bgrid* and *egrid*; see the Remarks (no default for *bgrid*; *egrid* default=*bgrid*)

LEV= *l1[u1] . . ln[un]* selects grids with the specified levels and units, e.g., SFC, 850[MB], 5000[M]; units are optional but must be in square brackets if specified

PARam= *p1 . . pn* selects grids with the specified parameters, for example, T, Z, RH; do not use with the DERIVE keyword; see the Remarks

SRC= *s1 . . sn* selects grids with the specified sources, for example, MDX, GFS, ETA

TIMe= *t1 . . tn* selects grids with the specified times

TRAnge= *btim etim inc* selects grids in the range of times *btim* through *etim,* incremented by *inc* (no default for *btim*; *etim* default=*btim*; *inc* default=1, meaning one hour)

Derived Grid Keywords

DERive= selects the component grids needed to derive a grid with the specified parameter, for example, DST or VOR; the grid listed with the HIST, LIST, or STAT parameters does not exist, but the components needed to derive such a grid do; do not use with the PARAM keyword; see the Remarks

MERidional= **YES** adds a correction term to account for the convergence of longitude lines at the poles

NO does not add a correction term; only valid with DERIVE=ABV, DSH, DST, DVG, or VOR; see the Remarks (default)

PLAnet= planet for which derive calculations are done; valid options are MERCURY, VENUS, EARTH, MARS, JUPITER, SATURN, NEPTUNE, URANUS (default=EARTH)

Mathematical Operations Keywords

G*n*= '*clause1*; *clause2*; . .; *clausen*'

grids to be used with the MATH keyword; separate select clauses with semicolons; single quotes are mandatory; specify *clause* using the following format:

searchkeyword value

valid options for *searchkeyword* are DAY, FHOUR, GPRO, GRID, LEV, PARAM, SRC, TIME; you can specify numbers between 1 and 99 in the keyword name (G1, G2, . . , G99); see the Remarks

INFo= '*text*' string placed in the header of the copied grid when the MATH keyword is used; maximum of 48 characters

MATh= '*expression*' mathematical operation to perform on the grids specified with the G*n* keywords; single quotes are mandatory; see the Remarks

NEWpar= *param punit level lunit src*
header information for the grid copied with the MATH keyword; parameter and level values, and associated units

param grid parameter; four characters maximum (default=MATH)

punit units of grid parameter; four characters maximum (default=NONE)

level grid level (default=from grid)

lunit units of grid level; two characters maximum (default=from grid)

src grid source; four characters maximum (default=from grid)

Grid Point Selection Keywords

COL= *bcol ecol* range of columns to list or edit; specify MAX for the last column in the grid (default=1 MAX for the STAT and HIST options; default=1 *bcol* for the LIST and EDIT options)

LAT= *lat1 lat2* range of latitudes to list or edit; range must be south to north (default=grid latitude extents)

LON= *lon1 lon2* range of longitudes to list or edit; range must be east to west (default=grid longitude extents)

RANge= *bval eval* range of grid point values to list or edit; *bval* can be MISS for missing values (default=all values)

MISS list or edit grid points with missing values

ROW= *brow erow* range of rows to list or edit; specify MAX for the last row in the grid (default=1 MAX for the STAT and HIST options; default=1 *brow* for the LIST and EDIT options)

EDIT Option Keywords

DEL= **YES** delete the destination grid file before copying the edited grid; see the Remarks

NO do not delete the destination grid file (default)

DGRid= beginning grid number in the destination grid file to place the edited grids; this keyword may overwrite existing grids; see the Remarks (default=append after the last grid in the grid file)

MAXgrd= maximum number of grids the newly created destination grid file can store; see the Remarks (default=1000)

TITLE= *'text'* title of the newly created destination grid file; 32 characters maximum; see the Remarks

VALue= *num* value to assign to grid points specified with the grid point selection keywords

MISS assigns a missing data value to the specified grid points

---

Output Keywords

BINsize= increment, or bin width, for the histogram; valid only with the HIST option (default=from *sdataset*)

BYTecount= **YES** lists the number of bytes received from the server

NO does not list the number of bytes (default)

FORmat= output format for grid point values; if multiple parameters are requested, FORMAT cannot be specified; see the Remarks

NUM= number of grids matching the criteria defined with the search keywords to list or edit; for example, if you specify NUM=50, the first 50 grids matching the search keywords are used (default=1)

ALL use all grids matching the search keywords

SUBsect= *slat nlat elon wlon incrow inccol*

geographic region to subsect from the source grid, and row and column reduction intervals to use within the subsect region, where:

slat, nlat southern and northern latitudes of the subsect region

elon, wlon eastern and western longitudes of the subsect region (*slat*, *nlat*, *elon*, *wlon* defaults=source grid bounds)

incrow, inccol row and column reduction intervals; 1 means use every row/column of grid points in the subsect region, 2 means use every other row/column, etc.; when subsecting a conformal projection grid, the *incrow* and *inccol* values must be the same (default=1 for both)

Note: If the source grid is a McIDAS grid, you can subsect it by row and column instead of latitude and longitude. To do so, specify row and column numbers instead of latitudes and longitudes in the first four parameters, and specify ROWCOL as the seventh parameter. For example, SUBSECT=10 40 80 160 1 2 ROWCOL uses rows 10, 11, 12, ..., 40 and columns 80, 82, 84, ..., 160 for the subsect region. (If you don't include the ROWCOL at the end, it will treat 10 and 40 as latitudes, and 80 and 160 as longitudes.)

UNIt= units to list for grid point values; not valid if multiple parameters are requested with the PARAM keyword (default=from grid header)

Remarks

Individual grids are stored in grid files. Each dataset position points to a single grid file. GRDINFO lists or edits individual grid point values in one or more grid files in a dataset. Use the search keywords to limit grid selection.

Use the RANGE, LAT and LON, and ROW and COL keywords to limit the selection of grid points. Using a combination of these grid point selection keywords limits grid points to those that meet all the specified criteria.

More than one grid can be altered using the EDIT option; however, only one value can be assigned to modify all the grids.

If you specify DEL=YES or a position number of a nonexistent grid file in the *ddataset* parameter, the edited grids are copied into a newly created grid file with the attributes specified in keywords MAXGRD and TITLE.

Using the DGRID keyword may result in grids being overwritten in the destination grid file. For example, if the destination grid file has 50 grids (in grids 1 to 50) and you specify NUM=10 DGRID=25, the edited grids will overwrite grids 25 to 34 in the destination grid file.

The GRID keyword copies grids from the grid file specified with the *sdataset position* number; the default *position* value, ALL, is not valid. When using the GRID keyword, do not specify additional search keywords, PARAM, LEV, DAY, etc., or NUM.

When using the DERIVE keyword with the LIST, HIST, and STAT options, GRDINFO lists information for a virtual grid: a grid that could be derived. The command searches for grids with components needed to derive a grid with the specified parameter. Only one derived parameter can be specified at a time, but you can request multiple grids, by specifying multiple levels or times, for example. Use GRDCOPY with the DERIVE keyword to create derived grids.

Use the FORMAT keyword options below to format output based on FORTRAN designators for data output:

Designator	Description
I	integer
F	floating point
E	exponential

To specify a parameter's output format, use a designator followed by a number. For example, I4 specifies an integer of up to four digits; F10.1 specifies a floating point decimal of up to 10 digits (including the decimal point) with one digit to the right of the decimal; E10.3 specifies an exponential number of up to 10 characters (including the decimal point and the four-character exponential flag) with three digits to the right of the decimal. Specify an X to use the parameter's default format. If the specified integer or floating point format is too small for the parameter's values, a series of asterisks (***) is displayed.

See Also

See the Remarks and Examples in the GRDCOPY or GRDDISP command for additional information on DERIVE, MERIDIONAL, and MATH keywords.

Examples

GRDINFO GFS/12

This entry lists the statistics for the first grid in the first grid file in dataset GFS/12. The statistics include the minimum and maximum values, mean, and standard deviation.

GRDINFO RT/GRID.10 HIST PARAM=T LEV=850 FHOUR=12

This entry displays the histogram of the 12-hour forecast, 850 mb temperature grid in position 10 of dataset RT/GRID.

GRDINFO GRIDS/ETA.1 DERIVE=SPD LEV=500 FHOUR=48

This entry lists the statistics for a wind speed grid that could be derived from 500 mb, 48-hour forecast time component grids in position 1 of dataset GRIDS/ETA.

GRDINFO CASEA.21 LIST PARAM=T LEV=700 FHOUR=24 ROW=15 COL=30 40 UNIT=C FORMAT=F5.1

This entry lists the temperature values stored in columns 30 through 40 of row 15 of a 24-hour forecast, 700 mb temperature grid in position 21 of the dataset with the alias name CASEA. The temperatures are listed in degrees Celsius. The temperature output is formatted as real numbers containing up to five digits (including the decimal point) with one digit to the right of the decimal point.

GRDINFO BLIZZARD/GFS.8000 EDIT BLIZZARD/EDIT.4003 VALUE=MISS LAT=40 45 LON=90 95 DERIVE=VOR LEV=500 FHOUR=48 DGRID=15

This entry creates a derived grid of relative vorticity using the 48-hour forecast, 500 mb u- and v- wind component grids in position 8000 of the dataset BLIZZARD/GFS. The new grid is placed in grid number 15 in position 4003 of the dataset BLIZZARD/EDIT, overwriting all pre-existing data. The values of grid points between 40° and 45° latitude and between 90° and 95° longitude are changed to missing data values (MISS).

GRDLIST

Lists grids in a dataset.

Format

GRDLIST *dataset [keywords]*

Parameter

dataset ADDE dataset name and absolute position; specify one of the following formats:

group/descriptor.position

alias.position

use only positive integers or ALL (all positions) for *position* (no default for *group/descriptor* or *alias*; default=ALL for *position*)

Search Keywords

DAY= *d1 . . dn* lists grids with the specified days

DERive= lists derived grids with the specified parameter, for example, DST or VOR; the grid listed does not exist, but the components needed to derive such a grid do; do not use with the PARAM keyword; see the Remarks

DRAnge= *bday eday inc* lists grids in the range of days *bday* through *eday*, incremented by *inc* days (no default for *bday*; *eday* default=*bday*; *inc* default=1)

ENSemble= *e1 . . en* lists grids with the specified ensemble numbers; each ensemble number must include a + or - sign, e.g., -1 -0 +0 +1

FDAy= lists grids with the specified forecast day; a grid's forecast day is determined by adding the forecast hour to the day and time; for example, a 12 UTC grid from day 95300 with a forecast hour of 72 has a forecast day of 95303; you cannot use FDAY with the FHOUR, FRANGE or MATH keyword

FHOur= *h1 . . hn* lists grids with the specified forecast hours; you cannot use FHOUR with the FRANGE, FDAY or FTIME keyword

FRANge= *bhr ehr inc* lists grids in the range of forecast hours *bhr* through *ehr*, incremented by *inc* hours; you cannot use FRANGE with the FHOUR, FDAY or FTIME keyword (no default for *bhr*; *ehr* default=*bhr*; *inc* default=1)

FTIme= lists grids with the specified forecast time; a grid's forecast time is determined by adding the forecast hour to the time; for example, a 12 UTC grid with a forecast hour of 18 has a forecast time of 6 on the following day; you cannot use FTIME with the FHOUR, FRANGE or MATH keyword

GPRo= *g1 . . gn* lists grids with the specified projection, for example, MERC, PS, LAMB, EQUI

GRIB= *geo par model level*

lists grids with the specified GRIB codes; the four values are the geographic, parameter, model and level codes that can be listed with the FORM=ALL keyword; values specified as an X (a placeholder) or not specified at all will match any GRIB code

GRId= *bgrid egrid* lists grids *bgrid* through *egrid* in the grid file specified in the *dataset* parameter; enter a grid number or LAST (last grid) for *bgrid* and *egrid*; see the Remarks (no default for *bgrid*; *egrid* default=*bgrid*)

LEV= *l1[u1] . . ln[un]* lists grids with the specified levels and units, e.g., SFC, 850[MB], 5000[M]; units are optional but must be in square brackets if specified

PARam= *p1 . . pn* lists grids with the specified parameters, for example, T, Z, RH; do not use with the DERIVE keyword; see the Remarks

SRC= *s1 . . sn* lists grids with the specified sources, for example, MDX, GFS, ETA

TIMe= *t1 . . tn* lists grids with the specified times

TRAnge= *btim etim inc* lists grids in the range of times *btim* through *etim,* incremented by *inc* (no default for *btim*; *etim* default=*btim*; *inc* default=1, meaning one hour)

Mathematical Operations Keywords

G*n*= '*clause1*; *clause2*; . .; *clausen*'

lists grids to be used with the MATH keyword; separate select clauses with semicolons; single quotes are mandatory; specify *clause* using the following format:

searchkeyword value

valid options for *searchkeyword* are DAY, FHOUR, GPRO, GRID, LEV, PARAM, SRC, TIME; you can specify numbers between 1 and 99 in the keyword name (G1, G2, . . , G99); see the Remarks

MATh= '*expression*' mathematical operation to be performed on the grids specified with the G*n* keywords; single quotes are mandatory; see the Remarks

---

Output Keywords

BYTecount= **YES** lists the number of bytes received from the server

NO does not list the number of bytes (default)

FORm= **SHORT** lists a one-line summary for each grid (default)

ALL lists all information for each grid

FILE *bpos epos*

lists only the grid file directories; none of the grids in the grid files are listed; all other keywords are ignored; *bpos* is the beginning position number or ALL; *epos* is the ending position number or LAST; *bpos* and *epos* override the position specified in the *dataset* parameter (*bpos* default=position specified in the *dataset* parameter, *epos* default=*bpos*)

NUM= number of grids matching the criteria defined with the search keywords to list; for example, if you specify NUM=50, the first 50 grids matching the search keywords are listed (default=1)

ALL lists all grids matching the search keywords

Remarks

Individual grids are stored in grid files. Each dataset position points to a single grid file. GRDLIST lists grids in one or more grid files in a dataset. The listing includes the parameter, level, day, time, source, forecast hour, forecast day, forecast time, grid number and projection. Use the search keywords to limit the output to the desired grids.

The GRID keyword lists grids in the grid file specified in the *dataset position* number; the default *position* value, ALL, is not valid. When using the GRID keyword, do not specify additional search keywords, PARAM, LEV, DAY, etc., or NUM.

If you specify PARAM=STREAML, WINDB or WINDV, GRDLIST does not search for grids with that parameter. Instead, it locates the u-component grids that match the other search keywords, then locates the matching v-component grids, and lists both sets of grids. These grids may be used to draw streamlines, plot wind barbs or plot wind vectors with the GRDDISP command. If you specify any other parameter with the PARAM keyword, the value entered is the value searched for in the grid's parameter field.

The DERIVE keyword lists a virtual grid: a grid that could be derived. GRDLIST searches for grids with the components needed to derive a grid with the specified parameter. The listing is for the grid that would be created using the GRDCOPY command. Use GRDCOPY with the DERIVE keyword to create derived grids. Once the derived grid is created, use GRDLIST with the PARAM keyword to list the output grid.

Use the G*n* keywords to specify the grids to be used with the MATH keyword. GRDLIST lists the grids but the mathematical operation is not performed. Each G*n* keyword specifies a single grid. G*n* is followed by a list of select clauses in single quotes. By default, subsequent G*n* keywords have the same select clauses, unless specified differently. For example, to request 850 and 1000 mb height grids from the 0:00 UTC GFS run, specify **G1='LEV 850;PARAM Z;TIME 0;SRC GFS' G2='LEV 1000'**.

The MATH keyword defines the operation to be performed on the specified grids, for example, MATH='G1**(SQRT(G2))'. Even though the math is not done, the MATH keyword must be used with the G*n* keywords.

See Also

See the GRDCOPY command for additional information on MATH keyword parameters. See the GRDCOPY or GRDDISP command for additional information on DERIVE keyword parameters.

Examples

GRDLIST NCEP/GFS

This entry lists the first grid in the first grid file in dataset NCEP/GFS.

GRDLIST NCEP/GFS.5

This entry lists the first grid in the grid file in position 5 of dataset NCEP/GFS.

GRDLIST NCEP/GFS.5 LEV=850 NUM=10

This entry lists the first 10 grids with a level of 850 mb in the grid file in position 5 of dataset NCEP/GFS.

GRDLIST NCEP/GFS.5 NUM=ALL

This entry lists all grids in the grid file in position 5 of dataset NCEP/GFS. Be careful when using **NUM=ALL**; grid files can contain thousands of grids. Use the search keywords to limit the output to the desired grids, as shown in the following examples.

GRDLIST NMC/ETA.ALL LEV=500 PARAM=Z T TD RH ABV DAY=#Y TIME=0 NUM=ALL

This entry lists all 500 mb height, temperature, dew point temperature, relative humidity and absolute vorticity grids from today's 00 UTC run in all grid files in dataset NMC/ETA.

GRDLIST NMC/ETA.1 LEV=TRO PARAM=Z T FHOUR=12 36 NUM=ALL

This entry lists all 12- and 36-hour forecast tropopause height and temperature grids in the grid file in position 1 of dataset NMC/ETA.

GRDLIST NMC/ETA.1 LEV=TRO PARAM=Z T FRANGE=12 36 6 NUM=ALL

This entry lists all 12-, 18-, 24-, 30- and 36-hour forecast tropopause height and temperature grids in the grid file in position 1 of dataset NMC/ETA.

GRDLIST NMC/ETA.10 LEV=TRO PARAM=Z T FDAY=#Y FTIME=18 NUM=ALL

This entry lists all forecast tropopause height and temperature grids valid at 18 UTC today in the grid file in position 10 of dataset NMC/ETA.

GRDLIST LOCAL/GRIDS.2 GRID=1 20

This entry lists grids 1 through 20 in the grid file in position 2 of dataset LOCAL/GRIDS.

GRDLIST MYGRIDS.2 GRID=LAST

This entry lists the last grid in the grid file in position 2 of the dataset with the alias name MYGRIDS. This information may be useful for subsequent GRDLIST or GRDCOPY commands, as it usually indicates about how many grids are in the grid file. For example, if this command shows that grid 800 is the last grid, then the grid file contains 800 or less grids. There are less than 800 grids only if there are "holes" in the grid file. For example, if grid 800 is the last grid, but there are no grids between 600 and 649, the grid file contains 750 grids.

GRDLIST STORM/FORECAST.ALL FORM=FILE

This entry lists the directories for all grid files in dataset STORM/FORECAST. None of the grids in the dataset are listed. If the grid file directories contain useful information such as day, time and contents, you can determine which grid file has the grids you want to list, copy or display. Then you can specify that grid file in the dataset parameter's position in your GRDLIST, GRDCOPY and GRDDISP commands to limit the search to that grid file.

GRDLIST MYGRIDS.1 DERIVE=SPD LEV=500 FHOUR=48

This command lists the wind speed grids that could be derived from 500 millibar, 48 hour forecast time component grids in position 1 of the dataset with the alias MYGRIDS.

GRDLIST MYGRIDS.1 G1='SRC GFS;PARAM Z;LEV 500;FHOUR 12' G2='FHOUR 24' G3='FHOUR 36' MATH='(G1+G2+G3)/3'

This command lists the three 500 millibar geopotential height grids from the GFS model needed for an average of the 12, 24, and 36 forecast hours. The grids listed are from position 1 of the dataset with the alias MYGRIDS.

GU

Graphics utility.

Formats

GU COLORS
GU DEL *name*
GU LIST *string*
GU MAKE *level color bframe eframe [keyword]*
GU MAKE *level c1 c2 c3 bframe eframe [keyword]*
GU OFF *level bframe eframe*
GU ON *level bframe eframe*
GU REST *name bframe eframe*
GU SAVE *name frame*
GU TABLE *name [keyword]*

Parameters

COLORS	lists the predefined colors in the **MCRGB.TXT** data file for use in the *color* parameter
DEL	deletes a graphics file
LIST	lists the saved graphics files
MAKE	sets the graphics color levels
OFF	turns a graphics color level off
ON	turns a graphics color level on
REST	restores a saved graphics file (default)
SAVE	saves the current graphics table to a file
TABLE	lists the values in a graphics file
name	graphics file name; do not specify an extension, for example, .GRX (REST option default=default table; TABLE option default=current; no default for DEL and SAVE options)
string	lists only the graphics files beginning with this character string (no default; list all graphics files)
level	graphics color level (OFF and ON options default=all levels; MAKE option default=0)
color	predefined color to apply to *level* (default=BLACK); use GU COLORS to list the valid colors
bframe	beginning graphics frame number on which to apply the graphics color level changes (default=current)
eframe	ending graphics frame number on which to apply the graphics color level changes (default=*bframe*)
c1	intensity of the first color component to apply to *level*
c2	intensity of the second color component to apply to *level*
c3	intensity of the third color component to apply to *level*; the valid range for parameters *c1*, *c2* and *c3* is 0 to 255 (default=0); the first color component is either blue or red, the second color component is green, the third color component is either red or blue; see the Remarks for more information
frame	graphics frame number displaying the graphics table to save (default=current)

Keyword

RGB=	YES	changes the order of the color components from the default (blue-green-red) to red-green-blue in the MAKE option's *c1*, *c2* and *c3* input parameters and the TABLE option's output listing
	NO	keeps the default order of the color components (default)

Remarks

The default file extension for the LIST, REST, SAVE and TABLE options is .GRX.

Parameters *c1*, *c2* and *c3* are the intensities (0-255) of the three color components that make up the final color applied to *level*. The RGB keyword determines the order in which the three color components are specified:

- If RGB=YES is specified, then the color order is red, green, blue (*c1* is RED intensity, *c2* is GREEN intensity, and *c3* is BLUE intensity).
- If RGB=NO (the default), then the color order is blue, green, red (*c1* is BLUE intensity, *c2* is GREEN intensity, and *c3* is RED intensity).

Examples

GU

This entry restores the default graphics table to the current graphics frame.

GU MAKE 4 WHITE 1 3

This entry sets graphics color level 4 to white for graphics frames 1, 2 and 3.

GU OFF 4 1 3

This entry turns graphics color level 4 off on graphics frames 1, 2 and 3.

GU TABLE

This entry lists the values in the current graphics table.

GUI

Starts the McIDAS-X Graphical User Interface.

Remarks

When you start the McIDAS-X GUI, a menu bar and frame control buttons are added to the Image Window. The Text and Command Window remains displayed and can be used to view text output or run McIDAS-X commands outside the GUI.

For more information, see the Help menu in the GUI or the section titled *Using the McIDAS-X Graphical User Interface*.

HEAT

Calculates the heat index and relative humidity.

Format

HEAT *temp dewpt units*

Parameters

temp	temperature for calculating the heat index
dewpt	dew point temperature for calculating the heat index; must be less than or equal to the value in *temp*
units	units of the specified temperature, dew point and output
F	Fahrenheit (default)
C	Celsius
K	Kelvin

Remarks

The heat index is calculated using an empirical algorithm developed by NOAA.

Valid temperature ranges are 80-110°F, 26.6-43.3°C, or 299.8-316.5°K.

The valid dew point temperature range is 55-85°F, 12.7-29.4°C, or 285.9-302.6°K.

Examples

HEAT 97 72

This entry lists the heat index and relative humidity for a temperature of 97° F and a dew point temperature of 72° F.

HEAT 38 29 C

This entry lists the heat index and relative humidity for a temperature of 38° C and a dew point temperature of 29° C.

HELP

Lists online helps.

Formats

HELP *command*
HELP *McBASI program*

Parameters

command	McIDAS command name
McBASI program	McBASI program name

Remarks

HELP lists the contents of *command*'s help file. The help file must have the extension .hlp and must reside in a directory in the REDIRECT list or in a directory in the **MCPATH** environment variable.

To display help for a McBASI program with a name identical to a McIDAS command, you must include the .MCB extension with the program name.

You can list an abbreviated HELP (the command syntax) while entering a command from the keyboard. After typing the command name or more,

Press: **ALT ?**

Examples

HELP

This entry lists the first line of all help files in the REDIRECT list and **MCPATH** directories.

HELP CUR

This entry lists the online documentation for the McIDAS command CUR.

HELP WHO.MCB

This entry lists the help for the McBASI program **WHO.MCB**.

HODO

Plots a hodograph of upper-air data.

Format

HODO *station time day [keywords]*

Parameters

station 4-character station ID, or 5-digit station ID number, e.g., KGRB or 72645

time observation time (default=0 if current time is between 0 UTC and 11:59 UTC, otherwise default=12)

day observation date; valid formats include YYYYDDD, YYYY/MM/DD and DD/MON/YYYY where YYYY, DDD, MM and DD are integers and MON is three or more letters, e.g., JAN or JANUARY (default=current)

Keywords

ANAlysis= *text gra* controls display of sounding/parcel analysis values in the text frame and graphics frame; specify YES in *text* to list all the values in the text frame; specify YES in *gra* to plot selected values in the graphics frame below the hodograph; specify NO in either or both to suppress the values (default=NO YES)

COLor= *low mid hi lab bg title par*
graphics color level for each component;specify 0 to omit the component (default=2 3 7 1 8 3 1)

- *low* line segments depicting low (surface to 699 mb) winds
- *mid* line segments depicting middle (700 mb to 400 mb) winds
- *hi* line segments depicting high (399 mb to 1 mb) winds
- *lab* wind speed labels, sounding table labels, and wind points
- *bg* background grid (wind speed circles)
- *title* plot title and sounding/parcel analysis values
- *par* sounding/parcel analysis labels

DATaset= *mand sig* source ADDE point dataset names for mandatory- and significant-level data; specify as *group*/*descriptor* or *alias*; you must specify NONE in the *sig* parameter or SIG=NO keyword to use only mandatory data, i.e., ignore the significant data (*mand* default=RTPTSRC/UPPERMAND; *sig* default=RTPTSRC/UPPERSIG if *mand* =RTPTSRC/UPPERMAND, otherwise *sig* default=NONE)

DEPth= depth in mb from the surface to include in the parcel analysis; the range is 0 - 400 where 0 means use only surface data (default=100)

GRA= graphic frame number to display the hodograph (default=current)

INT= interval for wind speed circles in the units specified in the UNIT keyword (default=calculated, based on the maximum wind speed)

OUTlonly=
- **YES** draws a hodograph without any data
- **NO** draws a hodograph and plots the upper-air data for the specified station (default)

PRAnge= *min max* mimimum and maximum pressure in mb; all winds in this range are plotted (default=all pressures)

SIG=	**YES**	include data from the significant-level dataset specified in the DATASET keyword (default)
	NO	do not include data from the significant-level dataset specified in the DATASET keyword; use only its mandatory-level dataset
UNit=	units for wind speed circles; run CU LIST to list the valid units and abbreviations (default=KTS)	

Remarks

HODO automatically erases the graphics before plotting the data.

A hodograph is a profile of a wind sounding representing the changing wind shear with height. Wind direction is the azimuth from the origin and wind speed is the radial distance from the origin. Northerly winds are plotted below the origin, and westerly winds are plotted to the right of the origin. The curve, or hodogram, between any two plotted points gives the wind shear between the two levels.

The table to the left of the wind speed circles lists pressure, height, wind speed and wind direction. The table includes all mandatory levels, plus significant wind and pressure levels whose pressure difference (from the adjacent levels in the table) is at least 30 mb.

Use the ANA keyword to list all available sounding/parcel analysis values in the text frame and plot selected values in the graphics frame below the diagram. To obtain the sounding/parcel analysis values, the reported heights are recomputed for hydrostatic consistency, though the adjustments are generally less than 10 meters. Also, the computed parcel temperature of a 12 UTC sounding is warmed 2 K to account for daytime heating. No adjustment is made to the computed parcel temperature of a 0 UTC sounding.

The following abbreviations are used for the sounding/parcel analysis values below the diagram: KI=K Index, LI=Lifted Index, SwI=Sweat Index, TTI=Total Totals Index, CVT=Convective Temperature in C, FMAX=Forecast Maximum Temperature in C, ThE=Equivalent Potential Temperature in K, EL=Equilibrium Level in mb, MIX=Mixing Ratio in g/kg, PW=Precipitable Water in mm, CAPE=Convective Available Potential Energy in J/kg, and HELI=Helicity in m**2/s**2. See *Appendix E, Stability Parameters* for definitions of these parameters.

Examples

HODO OUTL=YES

This entry plots a blank hodograph without data from any station.

HODO KOUN 0

This entry plots a hodograph of mandatory- and significant-level data for Norman, Oklahoma at 0 UTC of the current day. The sounding table to the left of the hodograph lists pressure, height, wind speed and wind direction at each level. Note that you can also specify the station using Norman's 5-digit station ID number, 72357.

HODO KEYW 12 PRANGE=400 1000 INT=10 UNIT=MPH ANA=YES

This entry plots a hodograph of mandatory- and significant-level data for Key West, Florida at 12 UTC of the current day. Only winds between 1000 and 400 mb are plotted. Since there are only five wind speed circles and an interval of 10 miles per hour was specified with the INT and UNIT keywords, the circles represent 10, 20, 30, 40 and 50 MPH winds respectively. The ANA=YES keyword outputs all parcel/sounding analysis values in the text frame and selected values below the hodograph in the graphics frame.

HODO KMFR 12 06/JUNE/1998 DATASET=LOCAL/MAND.1017

This entry plots a hodograph of mandatory-level data for Medford, Oregon at 12 UTC on 6 June 1998. The data is obtained from the file in position 1017 of the dataset LOCAL/MAND.

IGG

Grid utility.

Formats

IGG DEL *bgrid egrid* **GRIDF=**
IGG GET *sgridf bgrid egrid dgridf dgrid*
IGG LIST *bgrid egrid* **FORM= GRIDF=**
IGG MAKE *grid1 doption grid2 [keywords]*
IGG MAKE *grid1 option1 . . gridn optionn [keywords]*

Parameters

DEL	deletes grids	
GET	copies grids from one file to another or within the same file	
LIST	lists grid headers; see the Remarks	
MAKE	arithmetic operations performed on the grids; see *doption* and *optionn* below	
bgrid	beginning grid number; specify LAST for the last grid in the grid file (default=1)	
egrid	ending grid number; specify LAST for the last grid in the grid file (default=option dependent; see the Remarks)	
dgrid	destination grid number; if more than one grid is copied, *dgrid* is the beginning destination grid number (default=first available)	
sgridf	source grid file number (default=from IGU SET *gridf*)	
dgridf	destination grid file number (default=*sgridf*)	
grid1 . . n	a grid number	
doption	**ABV**	creates a grid of absolute vorticity with units $10^{-5}sec^{-1}$; specify the u-component grid as *grid1* and the v-component grid with the keyword V; *grid2* is ignored
	ADV	creates an advection grid of the *grid2* parameter in units/day; specify the u-component grid as *grid1* and the v-component grid with the keyword V
	DSH	creates a deformation shear grid; specify the u-component grid as *grid1* and the v-component grid with the keyword V; *grid2* is ignored
	DST	creates a deformation stretch; specify the u-component grid as *grid1* and the v-component grid with the keyword V; *grid2* is ignored
	DVG	creates a divergence grid with units $10^{-6}sec^{-1}$; specify the u-component grid as *grid1* and the v-component grid with the keyword V; if *grid2* is specified, the divergence of its parameter is computed
	GEO	creates u- and v-component grids of the geostrophic wind; specify a height grid as *grid1*; *grid2* is ignored
	SPD	creates a grid of wind speed; specify the u-component grid as *grid1* and the v-component grid with the keyword V; *grid2* is ignored
	VOR	creates a grid of vorticity with units $10^{-6}sec^{-1}$; specify the u-component grid as *grid1* and the v-component grid in the keyword V; *grid2* is ignored
option1 . . n	**ADD**	adds grids on a point-by-point basis, or adds a constant to a grid; see the Remarks
	AVG	averages grids on a point-by-point basis

CHGT changes all input grid points with values greater than the specified constant to that constant in the output grid; grid points with values less than the constant are not changed; see the Remarks

CHLT changes all input grid point values less than the specified constant to that constant in the output grid; grid points with values greater than the constant are not changed; see the Remarks

COM compares two grids for overlapping values; for a threat score, the second grid should be followed by two numbers that specify the range of values to include, for example, MAK 12 COM 18 -500 500

COR creates a grid of the coriolis force using the *grid1* boundaries

DIV divides a grid by another grid or constant; see the Remarks

FIL creates a grid with the specifications of the first grid but fills it with the value specified for the second grid, for example, MAK 12

MPY multiplies a grid by another grid or constant; see the Remarks

RMS computes RMS and bias differences of two named grids

SQR square root; applies to the first grid only; it can be used with any two successive grids to reproduce a grid of the square root of the sum of the squares

SUB subtracts a grid from another grid or a constant; see the Remarks

---

Keywords

ADD=	**YES**	corrects for meridional flow; valid only with the derived options ABV, DSH, DVG, VOR, and DST (default)
	NO	does not correct for meridional flow
DGRID=	destination grid number for the MAKE parameter (default=next available grid)	
FORm=	**SHORT**	lists a one-line summary for each specified grid (default)
	ALL	lists an expanded summary for each grid
GRIDF=	grid file number (default=from IGU SET *gridf*)	
LAB=	labels the resulting grid (default depends on operation)	
PLANET=	planet name (default=earth)	
SMO=	**YES**	smooths the resulting grid; only works with the derived options ABV, DVG, VOR, ADB, DST, and DSH (default)
	NO	does not smooth the resulting grid
V=	number of the v-component grid; used with the MAKE option when *grid1* is the u-component grid (default=u-grid +1)	

Remarks

When using LIST, *egrid*'s default is *bgrid*. If you do not specify *bgrid* and *egrid*, *egrid*'s default is LAST if FORM=SHORT and *bgrid* if FORM=ALL. When using DEL or GET, *egrid*'s default is *bgrid*.

To apply a constant to an arithmetic function (ADD, DIV, MPY, SUB), enter a negative number for the second named *grid*. The absolute value of this number is used as a constant.

In the CHLT and CHGT options, the number specified for the second named parameter (*gridn*) is used as a constant. Input grid point values less than (CHLT option) or greater than (CHGT option) this constant are changed to the constant in the output grid. See the last example.

Grids created using IGG MAKE and those created using GRDCOPY with the DERIVE or MATH keyword may not yield numerically identical results. However, the values for both are correct. The differences are the result of one of the following:

- By default, IGG MAKE applies a smoothing function and, if applicable, a meridional correction term to the data. GRDCOPY does not do this by default. You can override the IGG MAKE defaults by specifying SMO=NO and ADD=NO.
- Grids created using GRDCOPY will be in Standard International units, whereas grids created using IGG MAKE may not be. For example, a grid created using the TADV (temperature advection) function in IGG MAKE will be in degrees per day, while a similar grid created using GRDCOPY with the MATH keyword will be in degrees per second unless a factor is added to the equation to compute it in degrees per day.

The output of command IGG LIST contains the information below.

Column	Definition
1	grid number
2	year, Julian day
3	hours, minutes and seconds of the grid
4	name of the gridded parameter
5	description of the atmospheric level and units
6	name of the program generating the grid
7	hours the forcast grid is valid after generation
8	number of rows in the grid
9	number of columns in the grid
10	latitude and longitude of the upper-left corner
11	row grid spacing, in degrees
12	column grid spacing, in degrees

You can use the ADDE commands GRDCOPY and GRDLIST in place of IGG GET and IGG LIST respectively.

See Also

See commands GRDDISP and MDX to plot and contour grid point values.

Examples

IGG LIST

This entry lists all the grids in your workstation's default grid file.

IGG LIST 1 20

This entry lists all grids from 1 to 20 from the default grid file.

IGG DEL

This entry deletes grid one in the default grid file.

IGG GET 10 5 7 20 1

This entry moves grids 5, 6 and 7 from grid file 10 into grid file 20, starting at grid 1, or the next available grid.

IGG MAKE 2 SUB 1

This entry creates a new grid by subtracting grid 1 from grid 2.

IGG MAKE 2 MPY -4 ADD 5 DIV -60

This entry creates a new grid by multiplying each grid point in grid 2 by the constant 4, adding them to grid 5 and dividing by the constant 60.

IGG MAKE 100 CHLT 273

This entry creates a new grid by checking each grid point value in grid 100 and changing all grids point values less than 273 to 273. The grid points with values greater than 273 are not changed.

IGU

Grid file utility.

Formats

IGU DEL *bgridf egridf*
IGU SET *gridf*
IGU MAKE *gridf maxgrd "text*
IGU DIR *gridf date project "text*

Parameters

DEL	deletes a grid file
SET	sets the current grid file number
MAKE	generates a grid file
DIR	modifies a directory for a grid file
bgridf	beginning grid file number or local ADDE grid file specified as *group*/*descriptor*.*position* or *alias*.*position*
egridf	ending grid file number or local ADDE grid file specified as *group*/*descriptor*.*position* or *alias*.*position*
gridf	grid file number or local ADDE grid file specified as *group*/*descriptor*.*position* or *alias*.*position*
maxgrd	maximum number of grids in the grid file (default=159)
date	date the grid file was generated (default=current)
project	your project number (default=the project number specified at logon)
"*text*	identification text for your grid file; the double quote is mandatory

---

Remarks

There is no definitive limit to the number of grids that can be stored in a grid file. It is determined by the amount of disk space. The recommended limit is 1000.

Examples

IGU SET 2000

This entry points at grid file 2000. All subsequent grid operations are performed on file 2000 until you run another IGU SET command.

IGU DEL LOCAL/GRIDS.7 LOCAL/GRIDS.9

This entry deletes the grid files in positions 7 through 9 in the dataset LOCAL/GRIDS.

IMGCHA

Changes an image directory.

Format

IMGCHA *dataset [keywords]*

Parameter

dataset ADDE dataset name and position; specify one of the following formats:

group/descriptor.position

alias.position

only positive integers are valid for *position*

Keywords

BANd= changes the image band number; the range is 1 to 64

NA changes the band sequence to unavailable; reserved for GOES visible images prior to GOES-8

CTYpe= changes the calibration type

DAY= changes the Julian day (YYDDD) of the image

ELE= changes the beginning image element number

ID= changes the NEXRAD station ID to the specified value, e.g., MKX

LINe= changes the beginning image line number

MEMo= '*text*' changes the text in the memo field; 32 characters maximum

OTYpe= changes the original source type

RES= *line ele* changes the line and element resolution

SS= changes the sensor source number

STYpe= changes the source type

TIMe= changes the time (HH:MM:SS) of the image

VALcode= changes the validity code

Remarks

CAUTION:
Do NOT change any directory values unless you are familiar with the structure of the image data and its source.

Use command IMGLIST with FORM=ALL to list the image directory values that can be changed with the IMGCHA command. An example is shown below.

```
Pos Satellite/ Date Time Center Res (km) Image_Size Sensor Lat Lon Lat Lon --- ----
---- ------------ ------- --- --- --- --- ---------- 11 G-10 IMG 16 APR 02106
19:00:00 34 118 Band: 4 10.7um Surface temp; longwave window 5.54 2.55 480 X 640
proj: 0 created 2002106 194538 memo: RT GVAR type:GVAR cal type:RAW offsets:
data=3328 navigation= 256 calibration= 2816 auxiliary= 0 doc length: 0 cal length:
0 lev length: 0 PREFIX= 0 valcod: 0 zcor: 0 avg-smp: N start yyddd: 2002106 start
time:190013 start scan: 316 lcor: 3601 ecor: 16617 bytes per pixel: 2 ss: 74
Resolution Factors (base=1): Line= 4.0 Element= 4.0
```

The table below shows which IMGCHA keywords correspond to the output fields.

IMGCHA keyword	IMGLIST output field
BAND	Band
CTYPE	cal type
DAY	Date
ELE	ecor
LINE	lcor
MEMO	memo
RES	Resolution Factors
SS	ss
STYPE	type
TIME	Time
VALCODE	valcod

Examples

IMGCHA RT/GOES-IR.1 TIME=16:14:59

This entry changes the directory of the image in position 1 of datasetRT/GOES-IR.1. The image time is changed to 16:14:59.

IMGCHA N14.99 LINE=2075 MEMO='Band 2 NOAA-14 HRPT over Florida'

This entry changes the directory of the image in position 99 of the dataset with the alias name N14. The beginning image line number is changed to 2075 and the memo field is changed to "Band 2 NOAA-14 HRPT over Florida."

IMGCOPY

Copies image data from one dataset to another.

Format

IMGCOPY *sdataset ddataset [keywords]*

Parameters

sdataset source ADDE dataset name and position; specify one of the following formats:

group/descriptor.position

alias.position

a position greater than zero represents an absolute position in the dataset; a position less than or equal to zero represents a relative position in the dataset based on image time, for example, 0 is the most recent and -1 is the next most recent; to use the default position, do not specify *.position* (no default for *group/descriptor* or *alias*; default=0 for position)

ddataset destination ADDE dataset name and position; specify one of the following formats:

group/descriptor.position

alias.position

use only positive integers or the default for *position*; to use the default position, do not specify *.position* (no default for *group/descriptor* or *alias*; *position* default=the position following the most recent image)

Image Positioning Keywords

LATlon= *lat lon* latitude and longitude of the *sdataset* image to place at the *ddataset* image location specified in the PLACE keyword

LINele= *line ele sys*

line line number of the *sdataset* image to place at the *ddataset* image location specified in the PLACE keyword (default=0)

ele element number of the *sdataset* image to place at the *ddataset* image location specified in the PLACE keyword (default=*line*)

sys coordinate system of the *line* and *ele* values; specify either **F** for file (area) coordinates or **I** for image coordinates (default=F)

PLAce= location in the *ddataset* image to place the point specified in the LATLON, STATION or LINELE keywords; use one of the following:

ULEFT upper-left corner of the image (default when the LINELE keyword is used)

CENTER center of the image (default when the LATLON or STATION keyword is used)

STAtion= station ID to place at the *ddataset* image location specified in the PLACE keyword; specify as the ID, e.g., KMSN or YSSY, optionally followed by the station type in brackets, e.g., ARX[N] or ARX[V]; see the TYPE keyword in the STNLIST command for valid station types

---

Image Selection Keywords

DAY= selects from *sdataset* images with the specified day

ID= selects from *sdataset* images with the specified NEXRAD station ID, e.g., MKX

RTIme= *bmin emin* selects from *sdataset* images within the specified range of minutes in the current hour; overrides the TIME keyword

TIMe= *btime etime* selects from *sdataset* images within the specified time range

btime etime **C** copies all POES data for the specified time range; valid only with real-time POES data on an SDI server; cannot be used with LATLON, LINELE, PLACE, STATION, or SIZE keyword

---

Output Keywords

BANd= image band number to copy (default=existing band for single-band images; no default for multiband images); if the source image is AIRS Level 1b data in HDF format, you can copy a range of bands by specifying two values that indicate the beginning and ending bands; if the source image is GVAR raw data on an SDI server, MTSAT raw data on an SDI server, MODIS Level 1b data in HDF format, or real-time segmented Meteosat Second Generation Level 1.5 files transmitted by EUMETSAT, you can copy selected bands by specifying them in the same format as frames are specified in the LS command, e.g., BAND=2 5 for bands 2 and 5, and BAND=1 3-6 10 for bands 1, 3, 4, 5, 6 and 10; if there are multiple resolutions among the bands they are all copied at the lowest resolution

ALL copies all bands; if there are multiple resolutions among the bands they are all copied at the lowest resolution; BAND=ALL is not valid with STYPE=VISR

CF= **YES** creates a CF-compliant netCDF file (default)

NO creates a netCDF file in the previous (pre-2009.2) format, which is not CF-compliant; the CF=YES/NO keyword is valid only when writing to netCDF format destination datasets; it is ignored when writing to datasets of other formats

DOC= **NO** does not include image documentation when copying the *sdataset* image data (default)

YES includes image documentation; specify DOC=YES to include necessary calibration values when copying POES images

GAUge= *percent* displays the system's progress during the image copy; specify *percent* as a number between 0 and 50 to control how often status messages are displayed; for example, if you specify 25, a message is shown when the copy is 25%, 50%, and 75% complete

MAG= *lmag emag* line and element magnification of the data; use a positive integer for blowup and a negative integer for blowdown (default=1 *lmag*)

NAVtype= **LALO** *linc einc* use lat/lon navigation instead of the regular (native) navigation; this keyword is valid only if the source image is area or AVHRR Level 1b format; *linc* and *einc* specify the image line and element increments, e.g., NAVTYPE=LALO 4 3 requests lat/lon nav using the data from every fourth line and every third element (default=use native nav; if you specify LALO, lat/lon nav is used and the *linc* and *einc* defaults are the computed values that result in using the largest number of data points allowable without exceeding the max size of the AUX block)

RECT remaps MODIS imagery into a rectilinear projection and applies a bowtie correction using MRTSwath (MODIS Reprojection Tool - Swath, package courtesy of the U.S. Geological Survey); use this keyword only when the source image is MODIS Level 1B or Level 2 data on a remote server with the MRTSwath package installed; this keyword is ignored if used with any other type of data

OVERride= **NO** does not copy the image if the most recent image in *ddataset* contains the same date, time and band (default)

YES copies the image to the *ddataset* position following the most recent image even if the date, time and band are same; this keyword is ignored if a position number is specified in *ddataset*

SIZe= *line ele* number of lines and elements to copy (default=480 640 unless copying POES data and specifying the C option with the TIME keyword)

ALL copies the entire *sdataset* image including all bands and documentation; if the image has multiple bands with different sizes and resolutions (Stretched Data Format files on SDI servers, for example), all bands are copied at the lowest resolution and smallest size; the only other keywords valid with SIZE=ALL are DAY, RTIME, TIME, ALL, REPEAT, and CYCLE

	SAME	retains the original image size; includes the image documentation; SIZE=SAME can be used to copy a single band from a multibanded image
STYpe=	**VISR**	reduces 2-byte data to 1-byte brightness data; does not include the image documentation unless DOC=YES or SIZE=SAME is also specified; do not use STYPE=VISR with BAND=ALL or SIZE=ALL; see the Remarks
TRACKing=	**YES**	tracks image if allowed by the server; tracking allows commands to receive data as it's being ingested by the server; commands complete when their data request has been fulfilled; tracking is only supported on some servers (e.g., Area, GVAR and MTSAT) and it can be disabled by the server administrator
	NO	does not track image so server only sends data that is currently available (default=value of MCTRACK environment variable; if MCTRACK is not set, default=value set by server or server administrator)

UNIT= calibration unit for the output image file; use this keyword with GEOT- and NCDF-format destination datasets; to determine available units, display the source image and run the D command (default=primary calibration unit in the source image)

WL=	*bwl ewl*	copies all bands with image wavelengths in the range *bwl* to *ewl*; valid only with high spectral resolution data, e.g., Aqua AIRS
WN=	*bwn ewn*	copies all bands with image wave numbers in the range *bwn* to *ewn*; valid only with high spectral resolution data, e.g., Aqua AIRS

Loop Maintenance Keyword

CYCle= *inc tol* updates the destination dataset by copying the image specified in *sdataset* to the *ddataset* position determined by the following formula:

position=modulus(*(time-tol)*/*inc*,*numpos*)+1

see the Remarks (default=1:00 0, meaning a 1-hour increment and no time tolerance)

Multiple Copy Keywords

ALL=	**YES**	copies multiple *sdataset* images into *ddataset* positions; if the number of positions is greater than or equal to the number of images, all *sdataset* images are copied in time order with the oldest image in *ddataset* position 1; if the number of positions is less than the number of images, images are copied to all positions with the most recent image in the last *ddataset* position, the second most recent image in the previous position, etc; this keyword overrides the positions specified in *sdataset* and *ddataset*
	NO	does not copy multiple images (default)
CONTinue=	**YES**	continues copying of images when using ALL= or REPEAT= even if one or more of the source images is skipped because it does not contain the specified geographical region (default=**NO**, meaning that IMGCOPY stops if it encounters a source image that does not contain the specified geographical region); this keyword's YES option is valid only with AREA or raw GVAR format images

REPeat= number of *sdataset* images to copy; the image specified in *sdataset* is copied to the position specified in *ddataset*, the next image is copied to the next position, etc.; this keyword is valid only if the position numbers for *sdataset* and *ddataset* are positive integers (default=1)

Remarks

By default, the upper-left 480 lines and 640 elements of the *sdataset* image are copied to the specified position in *ddataset*. Use the keywords LATLON, LINELE, STATION and PLACE to specify different locations in the *sdataset* or *ddataset* images. Use the keyword SIZE to specify a different line and element size and the keyword MAG to blowup or blowdown the image.

The keyword STYPE=VISR converts the original 2-byte data to 1-byte brightness data. It does not include the image documentation unless DOC=YES or SIZE=SAME is also specified. 2-byte image data is usually a raw value which can be operated on with calibration coefficients to convert the data to other values such as radiance, temperature or albedo. If you do not need documentation or values other than albedo, use this keyword.

The keywords TIME and RTIME select an image to copy. They allow you to limit image selection to a specific time range. This is useful for choosing an image when new images are received frequently, such as GOES images during a Rapid Scan schedule. These keywords also allow you to spread out the scheduled image copies so scheduler activity is more evenly distributed throughout the hour.

The keyword CYCLE is most useful when used with the time scheduler to copy the most recent image to another dataset. The image is copied to the position in the destination dataset determined by the following formula.

position=modulus(*(time-tol)/inc, numpos*)+1

Time is the image time of the selected image, *tol* and *inc* are the time tolerance and increment values specified in the CYCLE keyword, and *numpos* is the number of positions defined in the destination dataset. The position determined with this keyword overrides the *position* specified in *ddataset*.

For example, if the command:

IMGCOPY SAT/GOES-IR4K CLIENT/GOESIR STATION=DCA CYCLE=1:00

is scheduled to run once an hour and five positions are defined in the dataset CLIENT/GOESIR, the 13:00 UTC image in SAT/GOES-IR4K is copied to position 4 in CLIENT/GOESIR, since *time*=13.0, *inc*=1.0 and *numpos*=5 (13.0/1.0=13; 13/5=2 with a remainder of 3; 3+1=4). The 14:00 UTC image is copied to position 5; the 15:00 UTC image is copied to position 1, etc.

If a similar IMGCOPY command is scheduled to run every half-hour, but CYCLE=00:30 and the destination dataset has six positions, the 20:00 UTC image is copied to position 5, since *time*=20.0, *inc*=0.5 and *numpos*=6 (20.0/.5=40; 40/6=6 with a remainder of 4; 4+1=5). The 20:30 UTC image is copied to position 6; the 21:00 UTC image is copied to position 1, etc.

Specify a *tol* value in the CYCLE keyword only when the exact image times vary in a way that may result in one or more of the images being copied into the position following the desired one. This can happen when the selected image has an image time greater than the ending of the time range defined for the desired position by the *inc* value and the number of positions in the destination dataset. The *tol* value adjusts the image time backward in the position calculation equation so the image is copied to the desired position. For example, if you want to copy each hour's "half-past" image (e.g., 00:30, 01:30, ..., 23:30) but the exact image times vary from 25 to 33 minutes past the hour, specify 1:00 for *inc* and a value of at least 00:04 for *tol* to ensure that the :30 to :33 images are copied to the correct positions so that none of the :25 to :29 images overwrite :30 to :33 images from the the previous hour. The value specified in *tol* must be less than one-half the value specified for *inc*.

You can use IMGCOPY to create a netCDF image file. To do so, specify a destination dataset that was created (with the DSSERVE command) with NCDF for its *format* and IMAGE for its TYPE= value. Use the UNIT keyword in IMGCOPY to select the calibration unit for the netCDF file. Only single-band copies are allowed to netCDF files, so if the source image is multi-band you must use the BAND keyword to select the band for the output netCDF file.

Examples

IMGCOPY G8/VIS-CONUS MY/DATA

This entry copies the most recent image in dataset G8/VIS-CONUS to the position following the most recent image in dataset MY/DATA. The destination image is 480 lines by 640 elements with the upper-left corner of the source image placed at its upper-left corner.

IMGCOPY G8/VIS-CONUS.-1 MY/DATA.12 STATION=STL SIZE=600 1000

This entry copies the second most recent image in dataset G8/VIS-CONUS to position 12 in dataset MY/DATA. The destination image is 600 lines by 1000 elements with St. Louis placed at its center.

IMGCOPY G8/VIS-CONUS.5 MY/DATA.1 LATLON=35.5 100.2 PLACE=ULEFT MAG=2

This entry copies the image in position 5 in dataset G8/VIS-CONUS to position 1 in dataset MY/DATA. The destination image is 480 lines by 640 elements, is blown up by a factor of 2 from the source image's resolution, and has earth coordinate 35.5° N, 100.2° W placed at its upper-left corner.

IMGCOPY G8/VIS-GLOBE.5 MY/DATA.1 LINELE=1400 2400 F PLACE=CENTER MAG=-7

This entry copies the image in position 5 in dataset G8/VIS-GLOBE to position 1 in dataset MY/DATA. The destination image is 480 lines by 640 elements, is blown down by a factor of 7 from the source image's resolution, and has file (area) coordinate 1400,2400 placed at its center.

IMGCOPY G8/VIS-GLOBE.5 MY/DATA.1 SIZE=ALL

This entry copies the image in position 5 in dataset G8/VIS-GLOBE to position 1 in dataset MY/DATA. The entire source image, including all bands and the documentation section, is copied to the destination image.

IMGCOPY N14/GAC SCRATCH/POES.9 DAY=#Y TIME=11:45 12:15 BAND=2 SIZE=SAME

This entry copies band 2 of the image in dataset N14/GAC with today's date and an image time between 11:45 and 12:15 UTC to position 9 in dataset SCRATCH/POES. The destination image is the same size and coverage as the source image. If more than one image in the source dataset

meets the specified time and date criteria, the most recent of those images is copied. If none of the source dataset's images meets those criteria, no image is copied.

IMGCOPY N14/GAC SCRATCH/POES.9 DAY=#Y TIME=11:45 12:15 BAND=ALL SIZE=900 408

This entry copies all bands of the image in dataset N14/GAC with today's date and an image time between 11:45 and 12:15 UTC to position 9 in dataset SCRATCH/POES. The destination image is 900 lines by 408 elements with the upper-left corner of the source image placed at its upper-left corner. If more than one image in the source dataset meets the specified time and date criteria, the most recent of those images is copied. If none of the source dataset's images meets those criteria, no image is copied.

IMGCOPY N14/GAC SCRATCH/POES.9 DAY=#Y TIME=12:34 12:36 C BAND=ALL

This entry copies all POES satellite data in dataset N14/GAC recorded today between 12:34 and 12:36 UTC. This data is copied to position 9 in dataset SCRATCH/POES. Because the TIME keyword's C option is specified, the command ignores the start times of the images in dataset N14/GAC and instead copies all data recorded during the specified time range. The C option is valid only with POES data on an SDI server. Use the NAVDISP command to preview the data coverage for a time range before copying it with IMGCOPY.

IMGCOPY RADAR/MDR_IMAGES.1 LOCAL/MDR.10 STATION=PIT REPEAT=4

This entry copies four consecutive images in dataset RADAR/MDR_IMAGES to four consecutive positions in dataset LOCAL/MDR. The images in positions 1, 2, 3 and 4 in dataset RADAR/MDR_IMAGES are copied to positions 10, 11, 12 and 13 in dataset LOCAL/MDR, respectively. The destination images are each 480 lines by 640 elements with Pittsburgh placed at the center.

IMGCOPY RADAR/MDR _IMAGES MY/MDR STATION=PIT PLACE=ULEFT ALL=YES

This entry copies multiple images from dataset RADAR/MDR_IMAGES to dataset MY/MDR. The number of images copied depends on the number of images in RADAR/MDR_IMAGES and the available positions in MY/MDR. For example, if RADAR/MDR_IMAGES contains 10 images and MY/MDR has six positions, the 6 most recent images are copied in time order to positions 1 through 6. If RADAR/MDR_IMAGES contains 10 images and MY/MDR has 10 or more positions, all 10 images are copied in time order beginning with the oldest image in position 1. The destination images are each 480 lines by 640 elements with Pittsburgh placed in the upper-left corner.

NOTE: You can run either of the two commands below once an hour from the time scheduler (see commands SKE, SKL and SKU) to copy the current hour's image to another dataset. The only difference between them is that they use different methods of choosing the position in the destination dataset. In the first example, the image is always copied to the position following the most recent image. In the second example, the image is copied to the position determined by this formula:

position=modulus(*time*/*inc*,*numpos*)+1

IMGCOPY RT/GOES-IR MY/GOES LATLON=40 100 RTIME=0 5

This entry copies the image in dataset RT/GOES-IR with an image time between 0 and 5 minutes after the current hour to the position following the most recent image in dataset MY/GOES. The destination image is 480 lines by 640 elements with earth coordinate 40° N, 100° W placed at its center.

IMGCOPY RT/GOES-IR MY/GOES LATLON=40 100 RTIME=0 5 CYCLE=1:00

This entry copies the image in dataset RT/GOES-IR with an image time between 0 and 5 minutes after the current hour to dataset MY/GOES. The destination image is 480 lines by 640 elements with earth coordinate 40° N, 100° W placed at its center. The position number in MY/GOES of the destination image is determined by the formula:position=modulus(*time*/1,*numpos*)+1 where *numpos* is the number of positions in MY/GOES. For example, if MY/GOES has six positions, the 17:00 UTC image is copied to position 6, the 18:00 UTC image is copied to position 1, etc. See the Remarks for additional details and examples.

IMGDEL

Deletes images in a dataset.

Format

IMGDEL *dataset epos [keywords]*

Parameters

dataset ADDE dataset name and position; specify one of the following formats:

group/descriptor.position

alias.position

a *position* greater than zero represents an absolute position in the dataset; a *position* less than or equal to zero represents a relative position in the dataset based on image time, for example, 0 is the most recent and -1 is the next most recent; the *position* ALL represents all positions in the dataset

epos ending position number of the data on the server; must be greater than 0; valid only if the *position* specified in *dataset* is greater than 0

Keywords

DAY=	*bday eday*	deletes images in the range of dates specified (no default for *bday; eday* default=*bday*)
ID=		deletes images with the specified NEXRAD station ID, e.g., MKX
PREview=	**YES**	lists the results of the delete operation without deleting the images
	NO	deletes the specified images (default)
SS=		deletes images with the specified sensor source number
TIMe=	*btime etime*	deletes images in the time range specified (no default for *btime; etime* default=*btime*)
VERify=	**YES**	asks for verification before deleting each image
	NO	deletes the images without verification (default)

Remarks

IMGDEL works only with datasets consisting of McIDAS area files. (It does not work with datasets consisting of other formats, e.g., GVAR, HDF or Level 1b.)

If you specify VERIFY=YES, you are prompted with the following instructions for each image selected.

```
Press Alt X to DELETE the image file Press Alt G to SKIP the image file Press Alt Q
to EXIT the program
```

The images marked for deletion with **Alt X** are not deleted until you have responded with an **Alt X** or **Alt G** for every image. If you exit the program using **Alt Q** before marking all the images, none are deleted.

Examples

IMGDEL STORM/GOES-IR.0

This entry deletes the most recent image in the dataset STORM/GOES-IR.

IMGDEL STORM/GOES-IR.1

This entry deletes the image in position 1 in the dataset STORM/GOES-IR.

IMGDEL STORM/GOES-IR.5 10

This entry deletes the images in positions 5 through 10 in the dataset STORM/GOES-IR.

IMGDEL STORM/GOES-IR.ALL

This entry deletes all the images in the dataset STORM/GOES-IR.

IMGDEL STORM/GOES-IR.ALL VERIFY=YES

This entry lists all the images in the dataset STORM/GOES-IR one at a time. For each image you are prompted to press **Alt X** to delete it, **Alt G** to keep it, or **Alt Q** to exit the program.

IMGDEL STORM/GOES-IR.ALL DAY=95100 95110

This entry deletes all the images with dates 95100 through 95110 in the dataset STORM/GOES-IR.

IMGDEL STORM/GOES-IR.ALL DAY=95100 95110 TIME=0 12

This entry deletes all the images in dataset STORM/GOES-IR with times 0 through 12 UTC on each day 95100 through 95110.

IMGDEL IR.1 4

This entry deletes the images in positions 1 through 4 in the dataset with the alias name IR.

IMGDISP

Displays image data.

Formats

IMGDISP *dataset frame [keywords]*
IMGDISP *predprod [keywords]*

Parameters

dataset ADDE dataset name and position; specify one of the following formats:

group/descriptor.position

alias.position

a *position* greater than zero represents an absolute position in the dataset; a *position* less than or equal to zero represents a relative position in the dataset based on image time, for example, 0 is the most recent and -1 is the next most recent; to use the default position, do not specify *.position* (no default for *group/descriptor* or *alias*; default=0 for *position*)

frame frame number to display the image; specify **OPP** to display the image on the frame opposite the current frame; see the CYCLE and SEQ keywords to maintain an image loop on a range of frames (default=current)

predprod predefined product in the **IMGDISP.CORE**, **IMGDISP.SITE** or **IMGDISP.USER** file; see the Remarks

---

Frame Positioning Keywords

LATlon= *lat lon* latitude and longitude of the image to place at the location specified in the PLACE keyword

LINele= *line ele sys*

line line number of the image to place at the location specified in the PLACE keyword (default=0)

ele element number of the image to place at the location specified in the PLACE keyword (default=*line*)

sys coordinate system of the *line* and *ele* values; specify either **F** for file (area) coordinates or **I** for image coordinates (default=F)

PLAce= location on the frame to place the point specified in the LATLON, STATION or LINELE keywords; use one of the following:

ULEFT upper-left corner of the frame (default when the LINELE keyword is used)

CENTER center of the frame (default when the LATLON or STATION keyword is used)

STAtion= station ID to place at the location specified in the PLACE keyword; specify as the ID, e.g., KMSN or YSSY, optionally followed by the station type in brackets, e.g., ARX[N] or ARX[V]; see the TYPE keyword in the STNLIST command for valid station types

---

Image Selection Keywords

DAY= selects from *dataset* images with the specified day

ID= selects from *dataset* images with the specified NEXRAD station ID, e.g., MKX

RTIme= *bmin emin* selects from *dataset* images within the specified range of minutes in the current hour; overrides the TIME keyword

TIMe= *btime etime* selects from *dataset* images within the specified time range

Output Keywords

BANd= image band number to display (default=existing band for single-band images; no default for multiband images)

EU= name of the enhancement table to apply to the image

GEOres=

YES displays the geographic resolution at the center of the source image for the resolution value in the annotation text at the bottom of of the frame

NO displays the image resolution factor (where "1" equals the base resolution of the instrument) for the resolution value in the annotation text at the bottom of the frame (default); if a positive MAG= is also specified then the displayed value is the image resolution factor divided by the MAG= value; if a negative MAG= is specified then the displayed value is the image resolution factor multiplied by the absolute value of MAG=

GRAy=

NO does not display a grayscale bar (default)

YES adds a continuous grayscale bar to the bottom of the image

repeat inc start line

adds a custom grayscale bar to the image

repeat brightness repeat factor (default=5)

inc brightness level increment; if greater than 0, the bar is incremented from 0 to 255; if less than 0, the bar is incremented from 255 to 0 (default=2)

start starting brightness level (default=0)

line starting TV line; must be greater than zero (default=number of lines in the frame minus 23)

MAG= *lmag emag* line and element magnification of the data; use a positive integer to blow up and a negative integer to blow down (default=1 *lmag*)

NAVtype=

LALO *linc einc* use lat/lon navigation instead of the regular (native) navigation; this keyword is valid only if the source image is area or AVHRR Level 1b format; *linc* and *einc* specify the image line and element increments, e.g., NAVTYPE=LALO 4 3 requests lat/lon nav using the data from every fourth line and every third element (default=use native nav; if you specify LALO, lat/lon nav is used and the *linc* and *einc* defaults are the computed values that result in using the largest number of data points allowable without exceeding the max size of the AUX block)

RECT remaps MODIS imagery into a rectilinear projection and applies a bowtie correction using MRTSwath (MODIS Reprojection Tool - Swath, package courtesy of the U.S. Geological Survey); use this keyword only when the source image is MODIS Level 1B or Level 2 data on a remote server with the MRTSwath package installed; this keyword is ignored if used with any other type of data

REFresh= '*commands*' *gframe*

runs the specified McIDAS-X commands after IMGDISP displays the image; use *gframe* to specify the beginning frame to display graphics created with the commands; see the Remarks (no default for *commands*; *gframe* default=*bfra* when using the SEQ, CYCLE, or ALL keyword, otherwise default=*frame*)

SIZe= *line ele* number of lines and elements of the image to display; do not exceed the frame size (default=frame size); note that if you specify PLACE=CENTER with SIZE=*line ele*, the location specified in LATLON=, LINELE= or STATION= is displayed at the TV coordinates equal to the *line* and *ele* values divided by 2

SU= name of the stretch table to apply to the image

TEXt= '*text*' *fgd bgd*

the annotation line's text and foreground and background brightness levels; the beginning and ending single quotes are mandatory for specifying text other than the default image annotation (default=X 232 32)

TRACKing=	**YES**	tracks image if allowed by the server; tracking allows commands to receive data as it's being ingested by the server; commands complete when their data request has been fulfilled; tracking is only supported on some servers (e.g., Area, GVAR and MTSAT) and it can be disabled by the server administrator
	NO	does not track image so server only sends data that is currently available (default=value of MCTRACK environment variable; if MCTRACK is not set, default=value set by server or server administrator)

WL= displays band with image wavelength closest to the specified value; valid only with high spectral resolution data, e.g., Aqua AIRS

WN= displays band with image wave number closest to the specified value; valid only with high spectral resolution data, e.g., Aqua AIRS

Multiple Image Load Keywords

ALL=	*bfra efra*	displays multiple images; if the number of frames in the range *bfra-efra* is greater than or equal to the number of images, all images are displayed in time order with the oldest image in *bfra*; if the number of frames is less than the number of images, images are displayed in all frames with the most recent image in *efra*, the second most recent image in the previous frame, etc.; do not use this keyword if a *position* other than the default value of 0 is specified in *dataset*
REPeat=	*num pinc finc*	displays *num* number of images in the dataset; the image specified in *dataset* is displayed in the frame specified in *frame*; *pinc* and *finc* determine the position and frame increments for subsequent image loads; this keyword is valid only if the *dataset position* number, *num*, *pinc* and *finc* are positive integers (default=1 1 1)

Loop Maintenance Keywords

CYCle= *inc bfra efra tol*

updates the range of frames *bfra-efra* by displaying the image specified in *dataset* on the frame determined by the following formula:

frame=modulus((*time-tol*)/*inc*,*numfra*)+*bfra*

see the Remarks (*inc* default=1:00, meaning a 1-hour increment; *tol* default=0, meaning no time tolerance; no default for *bfra* and *efra*)

SEQ= *bfra efra* updates the range of frames *bfra-efra* by displaying the image specified in *dataset* in the frame following the frame with the most recent image; the selected frame overrides the *frame* parameter

---

Miscellaneous Keywords

CAL=	**QTIR**	uses POES quick calibration for faster image loads of POES data
SF=	**YES**	automatically shows the frame prior to displaying the image
	NO	does not show the frame (default)

Remarks

By default, the upper-left corner of the image is positioned at the upper left corner of the frame. Use the keywords LATLON, LINELE, STATION and PLACE to specify a different image or frame position. Use the keyword MAG to blow up or blow down the image.

The predefined products specified in the *predprod* parameter are contained in the text files **IMGDISP.CORE**, **IMGDISP.SITE** and **IMGDISP.USER**. The **.USER** file overrides the **.SITE** file, which overrides the **.CORE** file. The **.CORE** file is supplied with McIDAS-X and

should not be modified. You can, however, change the settings of its products and/or create your own products. To do so, copy the **.CORE** file from **~mcidas/data** to your **$HOME/mcidas/data** or another **MCPATH** directory, rename it to **IMGDISP.USER**, then edit the existing products and/or add new ones.

The keywords TIME and RTIME select an image to display. They allow you to limit image selection to a specific time range. This is useful for choosing an image when new images are received frequently, such as GOES images during a Rapid Scan schedule. These keywords also allow you to spread out the scheduled image displays so scheduler activity is more evenly distributed throughout the hour.

The keyword CYCLE is most useful when used with the time scheduler to display the most recent image in a loop of frames. The image is displayed on the frame determined by the following formula.

frame=modulus((*time-tol*)/*inc*,*numfra*)+*bfra*

Time is the image time of the selected image, *tol* and *inc* are the time tolerance and increment values specified in the CYCLE keyword, *numfra*=*efra*-*bfra*+1, and *bfra* and *efra* are the range of frames specified in the CYCLE keyword. The frame determined with this keyword overrides the *frame* parameter

For example, if the command:

IMGDISP SAT/GOES-IR4K STATION=DCA CYCLE=1:00 1 5

is scheduled to run once an hour, the 13:00 UTC image in SAT/GOES-IR4K is displayed on frame 4, since *time*=13.0, *inc*=1.0, *numfra*=5 and *bfra*=1 (13.0/1.0=13; 13/5=2 with a remainder of 3; 3+1=4). The 14:00 UTC image is displayed on frame 5; the 15:00 UTC image is displayed on frame 1, etc.

If a similar IMGDISP command is scheduled to run every half-hour, but CYCLE=00:30 11 16, the 20:00 UTC image is displayed on frame 15, since *time*=20.0, *inc*=0.5, *numfra*=6 and *bfra*=11 (20.0/.5=40; 40/6=6 with a remainder of 4; 4+11=15). The 20:30 UTC image is displayed on frame 16; the 21:00 UTC image is displayed on frame 11, etc.

Specify a *tol* value in the CYCLE keyword only when the exact image times vary in a way that may result in one or more of the images being displayed in the frame following the desired one. This can happen when the selected image has an image time greater than the ending of the time range defined for the desired frame by the *inc*, *bfra* and *efra* values. The *tol* value adjusts the image time backward in the frame calculation equation so the image is displayed in the desired frame. For example, if you want to display each hour's "half-past" image (e.g., 00:30, 01:30, ..., 23:30) but the exact image times vary from 25 to 33 minutes past the hour, specify 1:00 for *inc* and a value of at least 00:04 for *tol* to ensure that the :30 to :33 images are displayed in the correct frames so that none of the :25 to :29 images overwrite :30 to :33 images from the the previous hour. The value specified in *tol* must be less than one-half the value specified for *inc*.

The REFRESH keyword runs the specified McIDAS commands after IMGDISP displays the image. You can use the notations (IMA) and (GRA) as parameters or keyword options in the commands. (IMA) is the number of the frame where the image is displayed. (GRA) is the number of the frame to display graphics produced by the commands. (GRA) is the same as (IMA) unless you specify a different beginning graphics frame number with the *gframe* parameter.

For example, to display new images in frames 1 through 5 but draw the corresponding maps and current surface temperature plots on frames 6 through 10, specify the following keywords:

SEQ=1 5 REFRESH='SF (IMA);EG (GRA);MAP GRA=(GRA) IMA=(IMA);SFCPLOT T NAV=C GRA=(GRA)' 6

If IMGDISP displays a new image in frame 1, the surface temperature plot and map are drawn on frame 6 because (IMA) is set to 1 and (GRA) is set to 6. If the new image is displayed on frame 2, the temperature plot and map are drawn on frame 7.

Examples

IMGDISP G8/VIS-CONUS

This entry displays the most recent image in dataset G8/VIS-CONUS on the current frame. The upper-left corner of the image is placed in the upper-left corner of the frame.

IMGDISP G8/VIS-CONUS.-1 STATION=STL TEXT='GOES-8 Visible Centered on St. Louis' 200 10

This entry displays the second most recent image in dataset G8/VIS-CONUS on the current frame, with St. Louis placed at the center of the frame. The default annotation at the bottom of the frame is replaced with the text *GOES-8 Visible Centered on St. Louis* in foreground brightness level 200 and background brightness level 10.

IMGDISP WV.1 STATION=STL

This entry displays the image in position 1 of the dataset specified in the predefined product WV or specified in the alias WV. The image is displayed with St. Louis placed at the center of the frame. If WV is a predefined product, it's defined in the **IMGDISP.SITE** or **IMGDISP.USER** file. If WV is an alias, it was created with the AKA command. This IMGDISP entry will error if WV is both a predefined product and an alias.

IMGDISP MKX_BREF1 ID=EVX MAG=2

This entry displays the product MKX_BREF1 that's defined in the **IMGDISP.CORE** file or modified in the **IMGDISP.SITE** or **IMGDISP.USER** file. The ID=EVX and MAG=2 keywords, which specify the NEXRAD station to select and an image blowup factor of 2, override the settings (if they exist) for those keywords in the predefined product. If an alias named MKX_BREF1 was previously created with the AKA command, this IMGDISP entry will error because you cannot have both an alias and a predefined product with the same name.

IMGDISP G8/VIS-CONUS.5 10 LATLON=35.5 100.2 PLACE=ULEFT MAG=2 GRAY=YES

This entry displays the image in position 5 in dataset G8/VIS-CONUS on frame 10, with earth coordinate 35.5° N, 100.2° W placed at the upper-left corner of the frame. The image is blown up by a factor of 2 from its original resolution and has the default grayscale bar at the bottom.

IMGDISP G8/VIS-GLOBE MAG=-7 LINELE=1400 2400 F PLACE=CENTER

This entry displays the most recent image in dataset G8/VIS-GLOBE on the current frame, with file (area) coordinate 1400, 2400 placed at the center of the frame. The displayed image is blown down by a factor of 7.

IMGDISP N14/GAC 1 DAY=#Y TIME=11:45 12:15 BAND=2 EU=POESVIS

This entry displays band 2 of the image in dataset N14/GAC with today's date and an image time between 11:45 and 12:15 UTC on frame 1. The enhancement table POESVIS is applied as the image is loaded. If more than one image in the dataset meets the specified time and date criteria, the most recent of those images is loaded. If none of the dataset's images meets the specified time and date criteria, no image is loaded.

IMGDISP STORM/IR.1 10 STATION=PIT REPEAT=4

This entry displays four consecutive images in dataset STORM/IR, with Pittsburgh placed at the center of each frame. The images in positions 1, 2, 3 and 4 are displayed on frames 10, 11, 12 and 13 respectively.

IMGDISP STORM/IR.1 20 STATION=PIT REPEAT=4 2 1

This entry displays four images in dataset STORM/IR, with Pittsburgh placed at the center of each frame. Because the position increment is 2 in the REPEAT keyword, the images in positions 1, 3, 5 and 7 are displayed on frames 20, 21, 22 and 23 respectively.

IMGDISP STORM/IR STATION=PIT PLACE=ULEFT ALL=1 10

This entry displays up to 10 images in dataset STORM/IR, with Pittsburgh placed at the upper-left corner of each frame. If STORM/IR contains more than 10 images, the ten most recent images are displayed in time order on frames 1 through 10. If STORM/IR contains 10 or less images, all images in STORM/IR are displayed in time order with the oldest image on frame 1.

NOTE: You can run either of the two commands below once an hour from the time scheduler (see commands SKE, SKL and SKU) to display the current hour's image in a loop of 6 frames. The only difference between them is that they use different methods of choosing the frame to display the new image. In the first example, the image is always displayed on the frame following the frame with the most recent image. In the second example, the image is displayed on the frame determined by the following formula:

frame=modulus(*time*/*inc*,*numfra*)+*bfra*

IMGDISP RT/GOES-IR LATLON=40 100 RTIME=0 5 SEQ=1 6

This entry displays the image in dataset RT/GOES-IR with an image time between 0 and 5 minutes after the current hour. The image is displayed with earth coordinate 40° N, 100° W placed at the center on the frame following the frame within the range 1 through 6 displaying the most recent image.

IMGDISP RT/GOES-IR LATLON=40 100 RTIME=0 5 CYCLE=1:00 1 6

This entry displays the image in dataset RT/GOES-IR with an image time between 0 and 5 minutes after the current hour. The image is displayed with earth coordinate 40° N, 100° W placed at the center on the frame determined by the following formula:

frame=modulus(*time*/1,6)+1

For example, the 17:00 UTC image is displayed in frame 6, the 18:00 UTC image is displayed in frame 1, etc. See the Remarks for additional details and examples.

IMGDISP RT/GOES-IR LATLON=40 100 RTIME=0 5 CYCLE=1:00 1 6 REFRESH='SF (IMA);EG (GRA); MAP GRA=(GRA) IMA=(IMA);SFCPLOT T SAT GRA=(GRA)' 11

This entry is the same as the one above, except that a surface temperature plot and map are drawn after the image is displayed. The image is displayed on a frame between 1 and 6. The temperature plot and map are drawn on the corresponding frame between 11 and 16 because the *gframe* parameter in the REFRESH keyword is set to 11. For example, if the image is displayed on frame 1, the temperature plot and map are drawn on frame 11; if the image is displayed on frame 2, the temperature plot and map are drawn on frame 12.

IMGFILT

Applies filters to image data.

Formats

IMGFILT *sdataset ddataset [keywords] "comment*
IMGFILT *sdataset1 sdataset2 ddataset [keywords] "comment*

Parameters

sdataset source ADDE dataset name and position; specify one of the following formats:

group/descriptor.position

alias.position

a position greater than zero represents an absolute position in the dataset; a position less than or equal to zero represents a relative position in the dataset based on image time, for example, 0 is the most recent and -1 is the next most recent; to use the default position, do not specify *.position*; specify two source datasets only if using FILTER=CLOUD, CORE, DIS or MERGE (no default for *group/descriptor* or *alias*; default=0 for *position*)

ddataset destination ADDE dataset name and absolute position; specify one of the following formats:

group/descriptor.position
alias.position

use only positive integers for *position*

"*comment* description placed in the image file directory's memo field; 32 characters maximum (default=no comment)

Image Positioning Keywords

LATlon= *lat lon* latitude and longitude of the *sdataset* image to place at the *ddataset* image location specified in the PLACE keyword

LINele= *line ele sys*

line line number of the *sdataset* image to place at the *ddataset* image location specified in the PLACE keyword (default=0)

ele element number of the *sdataset* image to place at the *ddataset* image location specified in the PLACE keyword (default=*line*)

sys coordinate system of the *line* and *ele* values; specify either **F** for file (area) coordinates or **I** for image coordinates (default=F)

PLAce= location in the *ddataset* image to place the point specified in the LATLON, STATION or LINELE keyword

ULEFT upper-left corner of the image (default when the LINELE keyword is used)

CENTER center of the image (default when the LATLON or STATION keyword is used)

STAtion= station ID to place at the *ddataset* image location specified in the PLACE keyword; specify as the ID, e.g., KMSN or YSSY, optionally followed by the station type in brackets, e.g., ARX[N] or ARX[V]; see the TYPE keyword in the STNLIST command for valid station types

---

Output Keywords

BANd= *b1 b2* bands in the source image to filter; specify *b2* only if using FILTER=CLOUD, CORE, DIS or MERGE (*b1* default=existing band for single-band images; no *b1* default for multiband images; *b2* default=*b1*)

CF= **YES** creates a CF-compliant netCDF file (default)

NO creates a netCDF file in the previous (pre-2009.2) format, which is not CF-compliant; the CF=YES/NO keyword is valid only when writing to netCDF format destination datasets; it is ignored when writing to datasets of other formats

ID= selects from *sdataset* images with the specified NEXRAD station ID, e.g., MKX

MAG= *lmag emag* line and element magnification of the data; use a positive integer for blowup and a negative integer for blowdown (default=1 *lmag*)

NAVtype= **RECT** remaps MODIS imagery into a rectilinear projection and applies a bowtie correction using MRTSwath (MODIS Reprojection Tool - Swath, package courtesy of the U.S. Geological Survey); use this keyword only when the source image is MODIS Level 1B or Level 2 data on a remote server with the MRTSwath package installed; this keyword is ignored if used with any other type of data

PROd= product name for destination image, up to four characters (default=PRD)

SCAle= *prodlo prodhi britlo brithi*

minimum and maximum product and brightness values for the calibration (*prodlo prodhi* defaults=dataset_min dataset_max; *britlo brithi* defaults=0 255)

SIZe= *line ele* number of lines and elements of the *sdataset* image to copy (default=480 640)

ALL copies the entire *sdataset* image including all bands and the documentation section

TAIl= *lo hi* tail percentage after filtering (default=0 0)

TRACKing= **YES** tracks image if allowed by the server; tracking allows commands to receive data as it's being ingested by the server; commands complete when their data request has been fulfilled; tracking is only supported on some servers (e.g., Area, GVAR and MTSAT) and it can be disabled by the server administrator

NO does not track image so server only sends data that is currently available (default=value of MCTRACK environment variable; if MCTRACK is not set, default=value set by server or server administrator)

UNIt= calibration units for output (default=BRIT)

WL= *l1 l2* filters source image bands with image wavelengths closest to those specified in *l1* and *l2*; valid only with high spectral resolution data, e.g., Aqua AIRS; specify *l2* only if using FILTER=CLOUD, CORE, DIS or MERGE (no default for *l1*; *l2* default=*l1*)

WN= *n1 n2* filters source image bands with image wave numbers closest to those specified in *n1* and *n2*; valid only with high spectral resolution data, e.g., Aqua AIRS; specify *n2* only if using FILTER=CLOUD, CORE, DIS or MERGE (no default for *n1*; *n2* default=*n1*)

Filter and Stretch Keywords

FILter= filter used to generate the destination image; see the Remarks for more details

CLEAN *replace bline eline pdiff ldiff*

CLOUD *replace constant*

CORE *brkpnt1 brkpnt2 replace1 replace2*

DIS *brkpnt1 brkpnt2 brkpnt3 brkpnt4 replace*

GRAD

HIGH1 *distance efficiency*

HIGH2

HOLE *brkpnt1 brkpnt2*

LOW1 *distance efficiency*

LOW2 *lcoef ecoef*

MERGE *brkpnt1 brkpnt2 constant*

REPLACE *outval bline eline belem eelem inval1..invaln*

SHOT *bline eline pdiff*

SPOT *omcon oacon imcon iacon cmin cmax* DISt= TARget=

STRetch= applies a stretch to *sdataset*'s brightness values

CONT contrast stretch (default)

HIST *bin* histogram stretch; increment, or bin width (default=16)

Remarks

IMGFILT generates a new image by applying a filter to data from a source image or images. The CLOUD, CORE, DIS and MERGE filters use two source datasets; all other filters use only one.

All filters are applied to each element of an image line. The CLEAN filter is also applied to each line in the image. The HIGH2 and LOW2 filters are applied to each element, but use surrounding lines and elements in their calculations.

The IMGFILT command automatically applies a contrast stretch to images after the filter process. The range of the stretch is specified with the SCALE keyword. Product values outside the range are set to zero. If no values are supplied, the minimum to maximum product values are stretched from 0 to 255. The tail percentages specified with the TAIL keyword clip the *ddataset* minimum and maximum data values before the stretch. Use this option to sharpen an image.

The *ddataset* has a generated product source type, PRD, and a 1-byte data calibration type, BRIT. Elements of the *ddataset* are in brightness units. The image file's calibration allows you display native product values by specifying the native units in the IMGPROBE command.

Use IMGOPER to perform arithmetic operations on individual image elements.

You can use IMGFILT to create a netCDF image file. To do so, the destination dataset must be NCDF-format and IMAGE-type. See the Remarks section in the DSSERVE command for more information.

Each option for the FILTER keyword is described below. Positional parameters and the equation used by the filter to generate the destination image are listed. Defaults for the filters are based on brightness (UNIT=BRIT).

CLEAN

This option cleans an image by first applying a shot noise filter to each element in the source image, and then locating and replacing bad lines. Each element is compared to the elements on either side and replaced if the values are significantly different. The average value of each line is then compared to the lines above and below and replaced if it is significantly different.

Positional parameters:

replace **AVE** average of surrounding values (default)

MIN *sdataset* minimum value

MAX *sdataset* maximum value

bline beginning line in the source image to clean (default=first line)

eline ending line in the source image to clean (default=last line)

pdiff absolute difference between an element's value and the value of the element on either side (default=15)

ldiff percentage difference between a line's average value and the average value of the line above and below (default=15)

Equation for each *ddataset* element:

if (the difference between the *sdataset* and the surrounding elements is < *pdiff*)
then *ddataset* = *sdataset*
else *ddatset* = *replace*

Equation for each *ddataset* line:

if (the difference between *sdataset* line average and the surrounding lines is < *ldiff*) and (deviation of element values on the *sdataset* line is > 0)
then *ddataset* = *sdataset*
else *ddataset* = *replace*

CLOUD

This option applies a cloud filter to an image by comparing each element to the elements of another image. Use this filter to mask off a portion of the first source image.

Positional parameters:

replace replacement value (default=minimum value in either *sdataset*)

constant additive constant (default=0)

Equation for each *ddataset* element:

if (*sdataset1* > *sdataset2* + *constant*)
then *ddataset* = *sdataset1*
else *ddataset* = *replace*

CORE

This option applies a core filter to two images by comparing elements to breakpoints. The resulting image has only two values.

Positional parameters:

brkpnt1 *sdataset1* breakpoint value (default=minimum value in either *sdataset*)

brkpnt2 *sdataset2* breakpoint value (default=maximum value in either *sdataset*)

replace1 success condition replacement value (default=maximum value in either *sdataset*)

replace2 failure condition replacement value (default=minimum value in either *sdataset*)

Equation for each *ddataset* element:

if (*sdataset1* > *brkpnt1*) and (*sdataset2* > *brkpnt2*)
then *ddataset* = *replace1*
else *ddataset* = *replace2*

DIS

This option applies a discriminate filter to two images by comparing elements in each image to different high and low breakpoints. Use this filter to mask off a portion of the first source image.

Positional parameters:

brkpnt1 low end breakpoint value for *sdataset1* (default=minimum value in either *sdataset*)

brkpnt2 high end breakpoint value for *sdataset1* (default=maximum value in either *sdataset*)

brkpnt3 low end breakpoint value for *sdataset2* (default=minimum value in either *sdataset*)

brkpnt4 high end breakpoint value for *sdataset2* (default=maximum value in either *sdataset*)

replace failure condition replacement value (default=minimum value in either *sdataset*)

Equation for each *ddataset* element:

if (*brkpnt1* < *sdataset1* < *brkpnt2*) and (*brkpnt3* < *sdataset2* < *brkpnt4*)
then *ddataset* = *sdataset1*
else *ddataset* = *replace*

GRAD

This option applies a gradient filter to the source image.

Equation for each *ddataset* element:

ddataset = ABSOLUTE($sdataset_n$ - ($sdataset_{n+1}$))

HIGH1

This option applies a one-dimensional high-pass filter to each element in the source image. It eliminates low-frequency deviations by clustering them around the distribution's midpoint. A sample average is determined using values on either side of the source element and the percentage of filter efficiency.

Positional parameters:

distance sample length surrounding the source element; used for sample average (default=50)

efficiency filter efficiency (default=100)

Equation for each *ddataset* element:

ddataset = (*sdataset* - (sample average) + (sample midpoint))

HIGH2

This option applies a two-dimensional high-pass filter to each element in the source image. It eliminates low-frequency deviations by clustering them around the distribution's midpoint. A sample average is calculated by the program using values in a two-dimensional region around the source element.

Equation for each *dataset* element:

ddataset = (*sdataset* - (sample average) + (sample midpoint))

HOLE

This option searches for missing data and fills the holes in the image using the average of the surrounding element values.

Positional parameters:

brkpnt1 low end breakpoint value (default=minimum *sdataset* value)

brkpnt2 high end breakpoint value (default=maximum *sdataset* value)

Equation for each *ddataset* element:

if (*brkpnt1* < *sdataset* < *brkpnt2*)
then *ddataset* = (average of the surrounding *sdataset* element values)
else *ddataset* = *sdataset*

LOW1

This option applies a one-dimensional low-pass filter to each element of the source image. It eliminates high-frequency deviations by replacing the source image data values with the average of the values on either side of the source element. The sample average is determined using the percentage of filter efficiency.

Positional parameters:

distance sample length in pixels surrounding the source element; used for sample average (default=50)

efficiency filter efficiency (default=100)

Equation for each *ddataset* element:

ddataset = (sample average)

LOW2

This option applies a two-dimensional low-pass filter to each element in the source image. It eliminates high-frequency deviations by replacing the source image data values with the average of the values in the sample defined with the line and element coefficients.

Positional parameters:

lcoef line coefficient; 0.0 < *lcoef* < 1.0 (default=0.5)

ecoef element coefficient; 0.0 < *ecoef* < 1.0 (default=*lcoef*)

Equation for each *ddataset* element:

ddataset = (sample average)

MERGE

This option merges two images by selecting the *sdataset1* value for each element if its value is between the specified breakpoints. Otherwise it selects the *sdataset2* value minus the specified constant.

Positional parameters:

brkpnt1 low end breakpoint value (default=minimum value in either *sdataset*)

brkpnt2 high end breakpoint value (default=maximum value in either *sdataset*)

constant subtractive constant (default=0)

Equation for each *ddataset* element:

if (*brkpnt1* < *sdataset1* < *brkpnt2*)
then *ddataset* = *sdataset1*
else *ddataset* = (*sdataset2* - *constant*)

REPLACE

This option replaces selected elements in the defined region of the source image with the specified replacement value.

Positional parameters:

outval replacement value (default=0)

bline beginning line in the source image region (default=first line)

eline ending line in the source image region (default=last line)

belem beginning element in the source image region (default=first element)

eelem ending element in the source image region (default=last element)

inval1..invaln source image values in the region to replace with *outval*; specify values in the list format, e.g., *val1 val2 val3* etc., and/or a range format, e.g., *bval-eval* (default=0-255)

Equation for each *ddataset* element:

if *sdataset* lies outside the region defined by the line and element ranges, or its value is not in *inval1..invalN*
then *ddataset* = *sdataset*
else *ddataset* = *outval*

SHOT

This option cleans an image by applying a shot noise filter to each element in the source image. Each element is compared to the elements on either side and replaced if the values are significantly different.

Positional parameters:

bline beginning line in the source image to clean (default=first line)

eline ending line in the source image to clean (default=last line)

pdiff maximum percentage of the product range specified in the SCALE keyword to allow before a new value for the pixel is derived using the average of two adjacent pixels (default=15)

Equation for each *ddataset* element:

if the difference between *sdataset* and the surrounding elements is < *pdiff*
then *ddataset* = *sdataset*
else *dataset* = (average of the surrounding *sdataset* element values)

SPOT

This option illuminates or darkens one or more spots on an image. The locations and sizes of the spots are determined by the TARGET and DIST keywords.

Positional parameters:

omcon value by which to multiply elements located outside the spots defined by the TARGET and DIST keywords (default=0.0)

oacon value to add to elements located outside the spots defined by the TARGET and DIST keywords (default=0.0)

imcon value by which to multiply elements located inside the spots defined by the TARGET and DIST keywords (default=1.0)

iacon value to add to elements located inside the spots defined by the TARGET and DIST keywords (default=0.0)

cmin conditional minimum for values located inside the spots defined by the TARGET and DIST keywords; values less than *cmin* are treated as if they are outside the spots (default=no minimum)

cmax conditional maximum for values located inside the spots defined by the TARGET and DIST keywords; values greater than *cmax* are treated as if they are outside the spots (default=no maximum)

Equation for each *ddataset* element:

if sdataset is located outside the spots defined by the TARGET and DIST keywords
then *ddataset* = (*sdataset1* * *omcon*) + *oacon*
else *ddataset* = (*sdataset1* * *imcon*) + *iacon*

Keywords valid only with the **SPOT** filter:

DISt= *dist1[unit1] . . distN[unitN]* radial distance(s) from the target(s); the distance value(s) determine the size of the spot(s); run CU LIST to list valid units (default=0[KM] for first value; last specified value for all subsequent values)

TARget= *tar1 . . tarN*
target location(s); these determine the location(s) of the spot(s); valid input formats are:

lat lon latitude and longitude, e.g., 43:08:26 89:20:43

station station ID or ID number, e.g., MSN or 72645

'city,st,co'	city, where *st* and *co* are the two-letter state and country codes for the city, e.g., 'MADISON,WI,US' or 'TORONTO,,CA'

Examples

IMGFILT GOES7/VIS.1 LOCAL/AREA.1

This entry applies a contrast stretch to the brightness data in the image in position 1 of dataset GOES7/VIS. The resulting image is placed in position 1 of dataset LOCAL/AREA.

IMGFILT BLIZZARD/G7-IR-4K.2 BLIZZARD/CLEAN.2 FILTER=CLEAN STATION=DCA

This entry applies clean and shot noise filters and contrast stretch to the image in position 2 of dataset BLIZZARD/G7-IR-4K. The resulting image is placed in position 2 of dataset BLIZZARD/CLEAN. The destination image is 480 lines by 640 elements and centered over Washington, DC.

IMGFILT COMPOSITE/MOSAIC LOCAL/MOSAIC.1 FILTER=HOLE 0 1 LATLON=-90 0

This entry applies a hole filter to the most recent image in dataset COMPOSITE/MOSAIC. The resulting image is placed in position 1 of dataset LOCAL/MOSAIC. The filter averages elements that have brightness values between 0 and 1 (inclusive) to fill holes and gaps in the data. The destination image is 480 lines by 640 elements and centered over the South Pole.

IMGFILT A/A.8101 A/A.4000 FILTER=REPLACE 119 2001 2209 3362 3726

This entry replaces the brightness values in a region of the image in position 8101 of dataset A/A. The brightness values between image lines 2001 and 2209 and elements 3362 and 3726 are replaced with a new brightness value of 119. The resulting image is 480 lines by 640 elements. It is placed in position 4000 of dataset A/A.

IMGFILT GOES9/IR A/A.4001 BAND=3 FILTER=LOW2 STRETCH=HIST STATION=IRK SIZE=600 600 PROD=LO-2 UNIT=TEMP

This entry applies a two-dimensional low pass filter and histogram stretch to the band 3 temperature data in the most recent image in dataset GOES9/IR. The resulting 600 line by 600 element image is placed in position 4001 of dataset A/A, has the product name LO-2, and is centered over Kirksville, MO.

IMGFILT EASTS/CONUS LOCAL/AREA.1 BAND=1 STA=KCHI MAG=-3 -6 FILT=SPOT .5 X 1.25 TARGET=BIS OAX TLX MKX LSX MRX PBZ RAX DIST=124[NMI]

This entry applies a spot filter to band 1 of the most recent image in dataset EASTS/CONUS. The resulting image is centered on Chicago, blown down by a factor of 3 in the line direction and 6 in the element direction, and is placed in position 1 of dataset LOCAL/AREA. The filter sets the brightness values of pixels outside the spots defined by the TARGET and DIST keywords to 50% of their original values, and pixels inside the spots to 125% of their original values. In this case, the spots show the 124 nautical mile coverage of the NEXRAD "short range" (S Rng) reflectivity products for NEXRAD stations BIS, OAX, TLX, MKX, LSX, MRX, PBZ and RAX.

IMGLIST

Lists images in a dataset.

Format

IMGLIST *dataset epos [keywords]*

Parameters

dataset ADDE dataset name and position; specify one of the following formats:

group/descriptor.position

alias.position

a *position* greater than zero represents an absolute position in the dataset; a *position* less than or equal to zero represents a relative position in the dataset based on image time, for example, 0 is the most recent and -1 is the next most recent; the *position* ALL represents all positions in the dataset; to use the default position, do not specify *.position* (no default for *group/descriptor* or *alias*; default=0 for *position*)

epos ending position number of the data on the server; must be greater than 0; valid only if the *position* specified in *dataset* is greater than 0

Keywords

BANd=	*bband eband*	lists information only for the specified range of bands; uses BAND as the FORM keyword default
	ALL	lists all bands (default)
DAY=	*bday eday*	lists only images within the specified range of dates (no default for *bday*; *eday* default=*bday*)
FORm=	image information to list (default=standard one-line image summary)	
	AUDIT	lists the one-line image summary plus the audit trail
	BAND	lists the one-line image summary plus the band information
	ALL	lists the expanded image directory plus the band information
	MEMO	list OPS output plus memo field and creation information
	OPS	list image directory information in LA command format
ID=	*station*	lists only images with the specified NEXRAD station ID, e.g., MKX
	LIST	lists the most recent image for all NEXRAD stations
SS=	lists only images with the specified sensor source number	
TIMe=	*btime etime*	lists only images within the specified range of times (no default for *btime*; *etime* default=*btime*)
WL=	*bwl ewl*	lists all bands with image wavelengths in the range *bwl* to *ewl*; uses FORM=BAND as the default output format; valid only with high spectral resolution data, e.g., Aqua AIRS
WN=	*bwn ewn*	lists all bands with image wave numbers in the range *bwn* to *ewn*; uses FORM=BAND as the default output format; valid only with high spectral resolution data, e.g., Aqua AIRS

Remarks

If a negative *position* is specified in *dataset*, all images from that position to the most recent are listed. For example, if you specify position -5, the six most recent images (positions 0, -1, -2, -3, -4, -5) are listed.

When using FORM=BAND or FORM=ALL, the latitude and longitude resolution values listed with each band are for the image's center point. The line and element resolution factors (where "1" equals the base resolution factor of the instrument) are listed in the last line of the FORM=ALL output. If there are multiple bands with different resolutions, the resolution factors in the last line are for the band(s) with the lowest resolution.

Examples

IMGLIST STORM/GOES-IR

This entry lists the most recent image in the dataset STORM/GOES-IR.

IMGLIST STORM/GOES-IR.1

This entry lists the image in position 1 in the dataset STORM/GOES-IR.

IMGLIST STORM/GOES-IR.-3

This entry lists the four most recent images (positions 0, -1, -2, -3) in the dataset STORM/GOES-IR.

IMGLIST STORM/GOES-IR.-4 BAND=3

This entry lists the five most recent images in the dataset STORM/GOES-IR that contain band 3.

IMGLIST STORM/GOES-IR.1 20 BAND=3

This entry lists the images in positions 1 through 20 in the dataset STORM/GOES-IR that contain band 3.

IMGLIST STORM/GOES-IR.5 10 FORM=BAND

This entry lists the images in positions 5 through 10 in the dataset STORM/GOES-IR. It also lists information about the bands in each image.

IMGLIST STORM/GOES-IR.ALL DAY=95100 95110

This entry lists all the images with dates 95100 through 95110 in the dataset STORM/GOES-IR.

IMGLIST STORM/GOES-IR.ALL DAY=95100 95110 TIME=0 12

This entry lists all the images in the dataset STORM/GOES-IR with times 0 through 12 UTC on each day 95100 through 95110.

IMGMAG

Changes the magnification of a displayed image.

Format

IMGMAG *frame [keywords]*

Parameter

frame frame to display the magnified image (default=current)

Keywords

MAG=	*lmag emag*	line and element magnification factors; see the Remarks (default=1 *lmag*)
REPeat=	number of consecutive frames to load (default=1)	
SF=	**YES**	shows the destination frame after the image is loaded
	NO	does not show the destination frame (default)
SFRame=	frame displaying the image to magnify (default=current)	
TRACKing=	**YES**	tracks image if allowed by the server; tracking allows commands to receive data as it's being ingested by the server; commands complete when their data request has been fulfilled; tracking is only supported on some servers (e.g., Area, GVAR and MTSAT) and it can be disabled by the server administrator
	NO	does not track image so server only sends data that is currently available (default=value of MCTRACK environment variable; if MCTRACK is not set, default=value set by server or server administrator)

Remarks

IMGMAG redisplays an image using factors specified with the MAG keyword by running an IMGDISP command. Because it runs an IMGDISP command, the following conditions must be met for IMGMAG to work.

- The original image must have been displayed using the IMGDISP command.
- The image file must be present on disk (as opposed to workstation memory).

If you use the REPEAT keyword, it is appended to the IMGDISP command run by IMGMAG. Subsequent images from the dataset are displayed on additional frames using the same geographic coverage, center point, and blowup factors.

The magnification factors specified with the MAG keyword are relative to the displayed image's resolution, not the image file's resolution. Values greater than 1 produce a blowup of the displayed image, and values less than 0 produce a blowdown.

If you specify MAG values greater than 1, the cursor changes to a box showing the geographic coverage of the magnified image to be displayed. Move the box to the area of the image you want to blow up and click the right mouse button or press **Alt G**. The cursor returns to its original size and shape after the command runs.

If you specify MAG keyword values less than 0, the cursor changes to a crosshair. Select the center point of the destination image by moving the cursor and clicking the right mouse button or pressing **Alt G**. The cursor returns to its original size and shape after the command runs.

To display the image with the same resolution but a different center point, specify MAG=1 and select the new center point.

Examples

IMGMAG

This entry redisplays the image in the current frame with a new center point. This point is chosen by moving the cursor and clicking the right mouse button or pressing **Alt G**. The magnification factor is not changed.

IMGMAG 2 MAG=2 3

This entry redisplays the image in the current frame on frame 2 using a line blowup factor of two and an element blowup factor of three. The geographic coverage of the output image is selected by moving the cursor and clicking the right mouse button or pressing **Alt G**.

IMGMAG 2 MAG=2 3 REPEAT=4

This entry is the same as the one above, but it also displays the next three images from the dataset on frames 3, 4, and 5. All four images have the same geographic coverage because IMGMAG runs a single IMGDISP command with the same center point and blowup factors for all the images.

IMGMAG 11 SFRAME=10 MAG=-3

This entry displays the image in frame 10 on frame 11 using a line and element blowdown factor of three. The output image is centered at the point chosen with the crosshair cursor.

IMGOPER

Applies mathematical functions to image data.

Format

IMGOPER *sdataset1 . . sdatasetn ddataset [keywords] "comment*

Parameters

sdataset1 . . n source ADDE dataset names and positions; specify one of the following formats:

group/descriptor.position

alias.position

a *position* greater than zero represents an absolute position in the dataset; a *position* less than or equal to zero represents a relative position in the dataset based on image time; for example, 0 is the most recent and -1 is the next most recent; to use default position, do not specify *.position* (no default for *group/descriptor* or *alias*; default=0 for *position*)

ddataset destination ADDE dataset name and absolute position; specify one of the following formats:

group/descriptor.position

alias.position

use only positive integers *position*

"*comment* description placed in the image file directory's memo field; 32 characters maximum (default=no comment)

Image Positioning Keywords

LATlon= *lat lon* latitude and longitude of *sdataset* images to place at *ddataset* image location specified in the PLACE keyword

LINele= *line ele sys*

line line number of the *sdataset* image to place at the *ddataset* image location specified in the PLACE keyword (default=0)

ele element number of the *sdataset* image to place at the *ddataset* image location specified in the PLACE keyword (default=*line*)

sys coordinate system of the *line* and *ele* values; specify either **F** for file (area) coordinates or **I** for image coordinates (default=**F**)

PLAce= location in the *ddataset* image to place the point specified in the LATLON, STATION or LINELE keyword; use one of the following:

ULEFT upper-left corner of the image (default when the LINELE keyword is used)

CENTER center of the image (default when the LATLON or STATION keyword is used)

STAtion= station ID to place at the *ddataset* image location specified in the PLACE keyword; specify as the ID, e.g., KMSN or YSSY, optionally followed by the station type in brackets, e.g., ARX[N] or ARX[V]; see the TYPE keyword in the STNLIST command for valid station types

---

Image Selection Keywords

DAY= selects from *sdataset* images with the specified day

ID= selects from *sdataset* images with the specified NEXRAD station ID, e.g., MKX

RTIme= *bmin emin* selects from *sdataset* images within the specified range of minutes in the current hour; overrides the TIME keyword

TIMe= *btime etime* selects from *sdataset* images within the specified time range

---

Data Selection Keywords

BANd= *b1 . . bn* applies the function to the specified image bands (default=existing band for single-band images; no default for multiband images)

LLMt= *l1 . . ln* lower limit thresholds for each input data value in the equation; if input value is less than the specified limit, the output value is zero

MAG= *lmag emag* line and element magnification of the data; use a positive integer for blowup and a negative integer for blowdown (default=1 *lmag*)

MISs= m1 . . mn missing data values for source datasets; if input value equals missing data value, the output value is zero (default=0)

NONE all pixels from source datasets are included in the analysis

SIZe= *line ele* applies the function to the specified number of *sdataset* lines and elements (default=480 640)

ALL applies the function to the entire *sdataset* image; all image positioning keywords are invalid with SIZE=ALL

TRACKing= **YES** tracks image if allowed by the server; tracking allows commands to receive data as it's being ingested by the server; commands complete when their data request has been fulfilled; tracking is only supported on some servers (e.g., Area, GVAR and MTSAT) and it can be disabled by the server administrator

NO does not track image so server only sends data that is currently available (default=value of MCTRACK environment variable; if MCTRACK is not set, default=value set by server or server administrator)

ULMt= *u1 . . un* upper limit thresholds for each input data value in the equation; if input value is greater than the specified limit, the output value is zero

UNIt= calibration type for the data, for example, BRIT, RAD, TEMP (default=BRIT)

Equation Format Keywords

ACOn= additive constant for the equation (default=0.0)

COEf= *c1 . . cn* coefficients for each term in the equation (default=1.0)

FORm= **ADD** addition; adds each term in the equation; to subtract terms, specify COEF=-1 or SIGN=M (default)

CMAX *max new* conditional maximum; uses only the term in the equation with the highest value less than the specified maximum; if all terms are greater than *max*, the *new* value is used (default=255 255)

CMIN *min new* conditional minimum; uses only the term in the equation with the lowest value greater than the specified minimum; if all terms are less than *min*, the *new* value is used (default=0 0)

MAX maximum; uses only the term in the equation with the maximum value

MIN minimum; uses only the term in the equation with the minimum value

MULT multiplication; multiplies each term in the equation; to divide terms, specify POW=-1

FUNc= transcendental function used in the equation; use one of the following:

	ALOG	antilogarithm
	EXP	exponential function
	LN	natural log
	LOG	logarithm
	SQRT	square root

MCOn= multiplicative constant for the equation (default=1.0)

OFF= *o1 . . on* offsets for each term in the equation (default= 0.0)

POWer= *p1 . . pn* powers for each term in the equation (default=1.0)

SIGn= *s1 . . sn* multiplies the data values by 1 or -1

P	plus; multiplies by 1 (default)
M	minus; multiplies by -1

Output Keywords

CF=	**YES**	creates a CF-compliant netCDF file (default)
	NO	creates a netCDF file in the previous (pre-2009.2) format, which is not CF-compliant; the CF=YES/NO keyword is valid only when writing to netCDF format destination datasets; it is ignored when writing to datasets of other formats
DOC=	**YES**	copies the *sdataset* image line documentation to the *ddataset* image (default=**NO**)
NAVtype=	**RECT**	remaps MODIS imagery into a rectilinear projection and applies a bowtie correction using MRTSwath (MODIS Reprojection Tool - Swath, package courtesy of the U.S. Geological Survey); use this keyword only when the source image is MODIS Level 1B or Level 2 data on a remote server with the MRTSwath package installed; this keyword is ignored if used with any other type of data

PROd= user-defined product name for the *ddataset* image, up to four characters (default=**PROD**)

SCAle= *prodlo prodhi britlo brithi*

minimum and maximum product and brightness values for the calibration (default=0 255 0 255)

Remarks

IMGOPER generates a new image by applying mathematical functions to data from one or more source images. The following equation computes the data value for each line/element pair in the destination image. This operation is performed repeatedly using data from source image line/element pairs as input values (*input*) until the entire image is completed.

output data value =

FUNC[ACON+(MCON*(FORM((COEF1*(OFF1+((SIGN1)**input1*)**POW1))

(COEF2*(OFF2+((SIGN2)**input2*)**POW2)) . .

(COEF*n* *(OFF*n* +((SIGN*n*)**inputn*)**POW*n*)))))]

The part of the equation operating on a single input data value, (COEF*n**(OFF*n*+((SIGN*n*)**inputn*)**POW*n*)), is referred to as a *term*. *n* represents the number of *sdataset* images. It may not be larger than 100. *input1 . . n* represents the individual data values from each of the source images. The FORM determines how the terms are combined. For example, if you specify FORM=MULT, the terms are multiplied.

Before the input data values are used in the equation, they are checked against the lower and upper limit thresholds set with the keywords LLMT and ULMT. If you do not specify limit thresholds, all data is used.

To use data from more than one band in a single multibanded image, you must specify that image with the *sdataset* parameter for each band used. For example, the following entry subtracts data values for band 10 from band 8, both in image A/A.1, and places the resulting image in A/A.5.

IMGOPER A/A.1 A/A.1 A/A.5 FORM=ADD BAND=8 10 COEF=1 -1

All source images must have the same projection, resolution and navigation. If they do not, use the command IMGCOPY or IMGREMAP before using IMGOPER. The source image can have unlimited number of lines and no more than 22,000 elements.

The destination image receives the directory information from the last source image. To change the image directory information, use the IMGCHA command.

To get better precision for operations that result in small values, use the keywords MCON and SCALE.

When using multiple source images, you must specify keyword values in the same order as the source images: *value1* applies to the data in *sdataset1*. If a keyword allows for multiple values, the last entered value is the default for all subsequent values. The default listed for the keyword only applies if the keyword is not used. In the following command, for example, because an offset was specified only for images A/A.1 and A/A.2, image A/A.3 receives an offset of 10, the last entered value.

IMGOPER A/A.1 A/A.2 A/A.3 D/D.5 FORM=ADD OFF=1 10

Currently each data value in the output image, *ddataset*, can contain only 1-byte data. If the range of output data values is greater than 256, the output data loses accuracy.

You can use IMGOPER to create a netCDF image file. To do so, the destination dataset must be NCDF-format and IMAGE-type. See the Remarks section in the DSSERVE command for more information.

Examples

IMGOPER A/A.1 A/A.1 A/A.5 FORM=ADD BAND=8 10 SIGN=P M

This entry subtracts data values of band 10 from band 8, both in the image in position 1 in dataset A/A, and places the resulting image in position 5 in dataset A/A.

IMGOPER A/A.1 A/A.4 A/A.5 FORM=ADD BAND=2 3 COEF=1 -2

This entry multiplies the data values of band 3 of the image in position 4 in dataset A/A by 2 and subtracts them from band 2 of the image in position 1 in dataset A/A and places the resulting image in position 5 in dataset A/A.

IMGOPER OLD/DATA.1 OLD/DATA.1 OLD/PROD BAND=8 10 COEF=1 -1 SCALE=-10 10 PROD=DIFF UNIT=TEMP

This entry subtracts band 10 temperature data from band 8 temperature data of the image in position 1 in dataset OLD/DATA and places the resulting image in dataset OLD/PROD in the position following the most recent image. The output values are given the product name DIFF. The difference values between -10 and 10 are scaled from 0 to 255.

IMGOPER MSI.4 MSI.4 RATIO.1 FORM=MULT BAND=7 8 POW=1 -1 PROD=RATI SCALE=50 150 50 200 UNIT=BRIT MCON=100

This entry divides the brightness values from band 7 by the brightness values from band 8 of the image in position 4 in the dataset with the alias name MSI. The resulting image is placed in position 1 in the dataset with the alias name RATIO. The output values are given the product name RATI and multiplied by 100. The product values from 50 to 150 are scaled from brightness 50 to 200.

IMGOPER GOES/IR.1 GOESIR.2 PROD.4 BAND=8 8 COEF=-1 1 SCALE=-10000 10000 UNIT=RAD PROD=TEST MCON=1000

This entry subtracts band 8 radiance data in the image in position 1 in dataset GOES/IR from band 8 radiance data in the image in position 2 in dataset GOES/IR and places the resulting image in position 4 in dataset PROD. The output values are given the product name TEST and multiplied by 1000. The product values from -10000 to 10000 are scaled from brightness 0 to 255.

IMGOPER I/TEST.8 I/PROD.4 FUNC=LOG SCALE=0 20 0 255 UNIT=RAD MCON=1000

This entry multiplies the data values in the image in position 8 in dataset I/TEST by 1000 and then takes the logarithm. The results from 0 to 20 are scaled from brightness 0 to 255 and placed in position 4 in dataset I/PROD.

IMGOPER A/ALL.8344 A/ALL.4004 MISS=0 FUNC=LN PROD=NTLN SCALE=0 6 UNIT=BRIT

This entry takes the natural logarithm of the brightness image data in the image in position 8344 in dataset A/ALL and places the resulting image in position 4004 in the same dataset. The output values are given the product name NTLN. The product values from 0 to 6 are scaled from brightness

0 to 255.

IMGOPER G-7.1 MET3.1 A.1 SCALE=0 130050 0 255 PROD=PRMS POW=2 FORM=ADD "Intermediate RMS
IMGOPER A.1 A.2 PROD=RMS FUNC=SQRT MCON=0.5 UNIT=PRMS "Actual RMS

These entries compute the Root Mean Square (RMS) difference between the data in the images in position 1 in the datasets with the alias names G-7 and MET3. The first entry computes the sum of the squares of the data. The second entry computes the average square root of the results of the first command. Before running IMGOPER, the two source images must be remapped so that they have identical navigation.

IMGOPER GOES7/IR.30 METEOSAT/IR.3 COMPOSITE.1 FORM=CMIN SIZE=ALL

This entry takes the entire image in position 30 in dataset GOES7/IR and the entire image in position 3 in dataset METEOSAT/IR and combines the two images using the CMIN option. Before running IMGOPER, the two source images must be remapped so that they have identical navigation.

IMGOPER TEST/AREA.1 TEST/AREA.2 LLMT=108 ULMT=144 SCALE=108 144 2 38
IMGOPER TEST/AREA.1 TEST/AREA.3 LLMT=148 ULMT=196 SCALE=148 196 43 136
IMGOPER TEST/AREA.2 TEST/AREA.3 TEST/AREA.4 FORM=ADD MISS=NONE

These entries apply a series of scaling to the brightness values in the image in position 1 in dataset TEST/AREA. All brightness values between 0 and 107 are set to zero, all brightness values between 108 and 144 are scaled from 2 to 38, and all brightness values between 148 and 196 are scaled from 43 to 136. All values outside the specified LLMT and ULMT values are assigned a value of zero. The third command adds the output from the first two together so that TEST/AREA.4 has the desired results.

IMGCOPY NEXRAD/BREF1.1 DTX/RADAR.51 ID=DTX SIZE=ALL
IMGCOPY NEXRAD/BREF1.3 DTX/RADAR.52 ID=DTX SIZE=ALL
IMGCOPY NEXRAD/BREF1.5 DTX/RADAR.53 ID=DTX SIZE=ALL
IMGCOPY NEXRAD/BREF1.7 DTX/RADAR.54 ID=DTX SIZE=ALL
IMGOPER DTX/RADAR.51 DTX/RADAR.52 DTX/RADAR.53 DTX/RADAR.54 DTX/RADAR.55 FORM=ADD COEF=0.25 0.25 0.25 0.25

These four IMGCOPY entries copy the four images in positions 1, 3, 5 and 7 of dataset NEXRAD/BREF1 to positions 51-54 of dataset DTX/RADAR. The IMGOPER entry then computes the mean of the four images and writes the output to the image in position 55 of dataset DTX/RADAR.

IMGPLOT

Displays a scatter plot of data values from two images.

Format

IMGPLOT *ximage yimage [keywords]*

Parameters

ximage image to use for x-axis of plot; specify a frame number or an ADDE dataset name and position (default=current frame)

yimage image to use for y-axis of plot; specify a frame number or an ADDE dataset name and position (default=frame opposite *ximage*)

Keywords

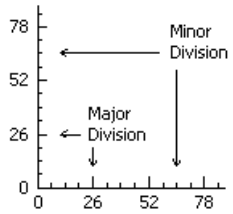
BANd=	*xband yband*	image band numbers to plot (default=displayed band or default band from dataset image)
COLor=		graphics color level of the plot (default=3)
FIT=	*order color*	draws a curve fit of the data, where *order* is the polynomial order (0-7) and *color* is the graphics color level (*order* default=0, meaning do not draw a curve fit; *color* default=5)
GRA=		graphics frame to display the scatter plot (default=current)
ID=	*xid yid*	NEXRAD station IDs (e.g., MKX and ARX) of the desired images in the datasets specified in *ximage* and *yimage*; this keyword is needed only with NEXRAD datasets
LATlon=	*lat lon*	center latitude and longitude of the image region to plot; not valid with the STATION keyword (default=location of cursor on frame)
LSTAT=	**YES**	includes statistics for the image region (default)
	NO	does not include statistics
SIZE=	*line ele*	number of lines and elements defining the image region to plot (default=cursor size)
STAtion=		station ID at the center of the image region to plot; specify as the ID, e.g., KMSN or YSSY, optionally followed by the station type in brackets, e.g., ARX[N] or ARX[V]; see the TYPE keyword in the STNLIST command for valid station types; this keyword is not valid with the LATLON keyword
TRACKing=	**YES**	tracks image if allowed by the server; tracking allows commands to receive data as it's being ingested by the server; commands complete when their data request has been fulfilled; tracking is only supported on some servers (e.g., Area, GVAR and MTSAT) and it can be disabled by the server administrator
	NO	does not track image so server only sends data that is currently available (default=value of MCTRACK environment variable; if MCTRACK is not set, default=value set by server or server administrator)
UNIt=	*xunit yunit*	calibration units to plot, for example, RAW, RAD, TEMP, BRIT, ALB; list the valid units for an image using the IMGPROBE command (default=BRIT *xunit*)
XDIv=	*major minor*	number of major and minor divisions for the x-axis (default=10 4)
XRAnge=	*min max*	range of data values for the x-axis of the plot (default=calculated)
YDIv=	*major minor*	number of major and minor divisions for the y-axis (default=10 4)
YRAnge=	*min max*	range of data values for the y-axis of the plot (default=calculated)

Remarks

IMGPLOT displays a scatter plot of data values from two images. Specify the images in *ximage* and *yimage* with frame numbers or with ADDE dataset names and positions. If you specify frame numbers, you must display the images using the IMGDISP command, and the image files must be present on disk (as opposed to workstation memory).

The location and size of the image region to be plotted are determined by the STATION or LATLON, and SIZE keywords. If you do not specify STATION or LATLON, you must select the region to be plotted. Move the cursor to the desired location on one of the frames and click the right mouse button or press **Alt G**.

The XDIV and YDIV keywords define the number of major and minor divisions in the plot. *Minor* defines the number of subdivisions for each major segment. Labels are placed at each *major* division. See the diagram below.



Examples

IMGPLOT

This entry draws a scatter plot of brightness data in the images displayed in the current and opposite frames. The scatter plot is drawn on the current frame in graphics color level 3 and uses the brightness data inside the cursor.

IMGPLOT 1 2 STATION=KSLC SIZE=480 640 UNIT=TEMP GRA=3

This entry draws a scatter plot of temperature data in the images displayed in frames 1 and 2. The scatter plot is drawn on frame 3 and uses the temperature data inside a 480 by 640 pixel region centered at Salt Lake City.

IMGPLOT 10 10 LATLON=35 90 BAND=4 4 UNIT=BRIT TEMP XRANGE=120 220 GRA=11

This entry draws a scatter plot of band 4 brightness and temperature data in the image displayed in frame 10. The scatter plot is drawn on frame 11 and uses the brightness and temperature data inside the cursor. The plot's x-axis ranges from 120 to 220 brightness levels.

IMGPLOT EAST/FDIR.1 EAST/FDV.1 BAND=3 1 STATION=KMIA SIZE=100 100

This entry draws a scatter plot of band 3 brightness data in the image EAST/FDIR.1, and band 1 brightness data in the image EAST/FDV.1. The scatter plot is drawn on the current frame and uses the brightness data inside a 100 by 100 pixel region centered at Miami.

IMGPROBE

Lists or plots image data for a region of the displayed image.

Format

IMGPROBE *option region unit [keywords]*

Parameters

option	**LIST**	lists the values for the data points in the specified region (default)
	STAT	lists the statistical parameters for the data points in the specified region
	CONT	contours the values of the data points in the specified region
	HIST	displays a histogram of the data points in the specified region
	PLOT	plots the values of the data points in the specified region; if the *region* is BOX or LINE, a line plot is drawn; if the *region* is POINT, the value of the data point is plotted see the Remarks for valid combinations of *option* and *region*
region	**BOX**	data points in a rectangle defined by the cursor limits or with the SIZE keyword
	LINE	data points along a line defined with the mouse
	POINT	data point at the center of the cursor's current position (default) see the Remarks for valid combinations of *option* and *region*
unit	output calibration units (default=BRIT, meaning grayscale brightness values between 0 and 255; all available units are listed when LIST POINT is requested)	

Keywords

BANd=	image band from which to obtain data (default=band displayed on the frame)	
EXClude=	*value1 . . valueN*	image data values to exclude from the analysis; specify numeric values and/or **SPACE**, which excludes the values of all points without navigation; see the Remarks for information about formats to use for entering multiple numeric values; this keyword is valid only with the STAT and HIST options
IMA=	image frame displaying the data to examine (default=current)	
OUTline=	**NO**	does not draw an outline of the region (default)
	YES *color*	draws an outline on the current graphics frame in the specified graphics color level (*color* default=3)
SIZe=	*line ele*	number of lines and elements defining a region centered at the cursor position; use this keyword only with a BOX *region*; see the Remarks regarding even number values for *line* and *ele* (default=cursor size)
SU=	stretch table name; use this keyword to examine modified brightness values that would be present if the image was displayed using the specified stretch table (default=stretch table currently applied to the displayed image)	
TRACKing=	**YES**	tracks image if allowed by the server; tracking allows commands to receive data as it's being ingested by the server; commands complete when their data request has been fulfilled; tracking is only supported on some servers (e.g., Area, GVAR and MTSAT) and it can be disabled by the server administrator
	NO	does not track image so server only sends data that is currently available (default=value of MCTRACK environment variable; if MCTRACK is not set, default=value set by server or server administrator)
WL=	obtains data from band with image wavelength closest to the specified value; valid only with high spectral resolution data, e.g.,	

Aqua AIRS (default=wavelength of band displayed on frame)

WN= obtains data from band with image wave number closest to the specified value; valid only with high spectral resolution data, e.g., Aqua AIRS (default=wave number of band displayed on frame)

Keywords valid with CONT option

CINt= contour interval (default=computed)

CLImit= *min max* contour interval limits (default=computed)

COLor= graphics color level of the contours (default=3)

DASh= **YES** use dashed lines for contours

NO use solid lines for contours (default)

GRA= graphics frame to display the contours (default=current)

LHGt= height of the contour labels, in pixels; specify zero for no contour labels (default=6)

LINt= contour label interval (default=2, label every other contour)

SMOoth= contour smoothing; the range is 0 to 50; larger numbers produce smoother contours (default=20)

Keywords valid with HIST and PLOToptions

COLor= graphics color level of the histogram or plot (default=3)

EG= **NO** does not erase the graphics frame before plotting HIST BOX, HIST LINE, PLOT BOX, or PLOT LINE diagrams (default=YES)

FORm= **ALL** include listing of data values and counts in text window output when specifying HIST or PLOT option (def=do not list)

GRA= graphics frame to display the histogram or plot (default=current)

LENgth= length of each class interval for HIST option's diagram and FORM=ALL output, and for PLOT option's FORM=ALL output (default=1.0)

PTYpe= **DOT** plots data as points

SOLID plots data with connecting line segments (default)

XRAnge= *min max* range of values for the x-axis

YRAnge= *min max* range of values for the y-axis

Keywords valid with Interactive and Batch Modes

DATaset= name and position of the ADDE dataset containing the image to read data from; specify as *group/descriptor.position* or *alias.position*; valid only when used with MODE=N, SIZE, and LATLON, LINELE, or STATION keywords; not valid with the CONT option, LINE region, or PLOT POINT combination (default=dataset and position of the displayed image; if a different dataset is specified the default position is 0)

ID= NEXRAD station ID (e.g., MKX) of the desired image in the dataset specified in the DATASET keyword; this keyword is needed only with NEXRAD datasets

LATlon= *lat lon* latitude and longitude for the POINT region, or to place at the BOX region location specified in the PLACE keyword; LATLON is valid only with MODE=N, and you cannot use LATLON with the LINELE or STATION keyword (default=location of the cursor on the frame)

LINele= *line ele sys* line and element numbers and their coordinate system for the POINT region, or to place at the BOX region location specified in the PLACE keyword; for the coordinate system, specify either **F** for file (area) coordinates or **I** for image coordinates; LINELE is valid only with MODE=N, and you cannot use LINELE with the LATLON or STATION keyword (default=location of the cursor on the frame)

MODE= **N** non-interactive *batch* mode; uses the cursor's position and size at time the command is run or the LATLON= or LINELE= or STATION= with SIZE= to determine the image region; not valid with LINE region

I interactive mode; you can select multiple image regions using the right mouse button or **Alt G** with commands like IMGPROBE LIST BOX, then press **Alt Q** to quit (default)

PLAce= location in the BOX region to place the point specified in the LATLON, STATION or LINELE keywords; use one of the following:

ULEFT upper-left corner of the BOX region (default when the LINELE keyword is used)

CENTER center of the BOX region (default when the LATLON or STATION keyword is used)

STAtion= station ID for the POINT region, or to place at the BOX region location specified in the PLACE keyword; specify as the ID, e.g., KMSN or YSSY, optionally followed by the station type in brackets, e.g., ARX[N] or ARX[V]; this keyword is valid is valid only with MODE=N and you cannot use it with the LATLON or LINELE keyword (default=location of the cursor on the frame)

Remarks

IMGPROBE lists or plots data for a region of the image displayed on the current frame or an image specified in the DATASET keyword. When using a displayed image, the image file must be present on disk (as opposed to workstation memory) because IMGPROBE uses data directly from the image file.

The following combinations of *option* and *region* parameters are valid:

option	*region*
CONT	**BOX**
LIST	**BOX**, **LINE** and **POINT**
HIST	**BOX** and **LINE**
PLOT	**BOX**, **LINE** and **POINT**
STAT	**BOX** and **LINE**

When you use the PLOT option with the BOX region, the x-axis of the line plot lists the pixel numbers in the box from left to right and top to bottom. Pixel number 1 is the upper-left data point, and the largest pixel number is the bottom-right data point.

When you use the LIST option with the BOX or LINE region, the output is listed in Output Blocks. There is no limit to the number of lines in an Output Block, so the output is divided into multiple Output Blocks only if it's too wide to fit in the text frame. The number of Output Blocks depends on the number of data points in each line of output and the calibration unit selected. For example, you can fit more brightness values across the text frame than IR radiance values. In the output for the BOX region, each successive Output Block is located to the right of the previous Output Block.

If you specify an even number for the *line* or *ele* parameter of the SIZE keyword, it is not possible to place an equal number of lines or elements on each side of the cursor center. The bottom line or right column of elements is omitted. For example, if the cursor is positioned at line 100 and element 200 and you specify SIZE=50 50, the data points used are those in a box covering lines 75 to 124 and elements 175 to 224.

When using a BOX region with any option, there must be less than 524,288 data points in the region. To calculate this number, you must know the image blowup or blowdown factor. If the image is displayed:

- at the image's resolution (blowup or blowdown factor of 1), use the formula: *region height* x *region width*
- with a blowup factor of *M*, use the formula:
  {[(*region height* - 1)/*M*] + 1} x {[(*region width* - 1)/*M*] + 1}
  note: if a division in the formula yields a decimal value, it is truncated before adding 1 and multiplying the terms
- with a blowdown factor of *N*, use the formula: (*region height* x *N*) x (*region width* x *N*)

For example, a 10 by 100 region size over a 1 km resolution image displayed with a blowup/blowdown factor of 1 contains 1000 data points. If the image is displayed at 2 km resolution (blowdown factor of 2), the region contains 4000 data points. If it is displayed at 0.5 km resolution (blowup factor of 2), the region contains 250 data points.

If you specify a BOX or POINT region, MODE=N, and a dataset in the DATASET keyword, you must specify the location with the LATLON,

LINELE, or STATION keyword. If it is a BOX region, you must also specify the region size with the SIZE keyword.

If you specify MODE=N but do not specify the DATASET keyword, the cursor is moved to the location specified in the LATLON, LINELE or STATION keyword.

Use the **IMGPROBE LIST POINT** command to determine valid options for the *unit* parameter.

You can enter numeric values in the EXCLUDE keyword using the list format (*value1 value2 value3* etc.) and/or range format (*x*-*y*B*n* where *x* and *y* are the beginning and ending values, and *n* is the increment). If you don't specify an increment, all real numbers from *x* through *y* are excluded. For example, EXCLUDE=0 126-129 255 excludes 0, all values from 126 through 129, and 255; EXCLUDE=0 126-129B1 255 excludes 0, 126, 127, 128, 129 and 255; EXCLUDE=0 126-130B2 255 excludes 0, 126, 128, 130 and 255; EXCLUDE=-12--1 excludes all values from -12 through -1; EXCLUDE=150-140 excludes all values greater than 150 and all values less than 140.

Examples

IMGPROBE

This entry lists information for one or more data points at the center of the cursor, including all calibration units for the image. The command is interactive, so you are prompted to select each point by moving the cursor and pressing the right mouse button or **Alt G**. Press **Alt Q** to quit.

IMGPROBE LIST LINE

This entry lists information for the data points along one or more lines. You are prompted to define each line's beginning and ending points using the mouse; then, press **Alt Q** to quit. The lines are not drawn on the frame.

IMGPROBE LIST LINE OUTLINE=YES 2

This entry lists information for the data points along one or more lines. You are prompted to define each line's beginning and ending points using the mouse; then, press **Alt Q** to quit. The lines are drawn on the frame in graphics color level 2.

IMGPROBE STAT BOX OUTLINE=YES BAND=4 EXCLUDE=255 SPACE MODE=N

This entry draws an outline around the data points in the region defined by the cursor limits and lists the statistical values for band 4 of those data points. Pixels without navigation or with a brightness value of 255 are not included in the statistical analysis.

IMGPROBE CONT BOX BRIT SIZE=200 300 CINT=10 MODE=N

This entry contours brightness values of the data points in a 200 by 300 pixel region centered at the cursor position. Contours are drawn at an interval of every 10 brightness values.

IMGPROBE PLOT LINE OUTLINE=YES 5 COLOR=4

This entry draws a line plot of the data point values along a line. You are prompted to define the line's beginning and ending points using the mouse. The plot is drawn in graphics color level 4. The line is drawn in graphics color level 5. The command is interactive, so you can draw additional line plots by selecting each line's beginning and ending points using the mouse. Note, however, that the graphics are erased each time you select a new beginning point since EG=YES is the default.

IMGPROBE HIST BOX SU=WARM COLOR=2

This entry draws a histogram of the values of the data points in the region defined by the cursor limits. The histogram is drawn in graphics color level 2. The values for the histogram are those that would be present if the image was displayed using the stretch table WARM. The command is interactive, so you can draw additional histograms by selecting new box regions using the mouse. Note, however, that the graphics are erased each time you select a new region since EG=YES is the default.

IMGPROBE HIST BOX TEMP XRANGE=200 300 YRANGE=0 100 LENGTH=.1 DATASET=GOES/IR.5 MODE=N LATLON=40 100 SIZE=50 50

This entry draws a histogram of the temperatures of the region's data points. The region is a 50- by 50-pixel box centered at 40°N, 100°W on the image in position 5 in dataset GOES/IR. The histogram's x-axis range is 200 to 300 degrees Kelvin, and the y-axis range is 0 to 100 occurrences. The length of the class interval is 0.1 degrees Kelvin.

IMGREMAP

Remaps images into different projections.

Format

IMGREMAP *sdataset ddataset [keywords]*

Parameters

sdataset source ADDE dataset name and position; specify one of the following formats:

group/descriptor.position
alias.position

a *position* greater than zero represents an absolute position in the dataset; a *position* less than or equal to zero represents a relative position in the dataset based on image time; for example, 0 is the most recent and -1 is the next most recent; to use default position, do not specify *.position* (no default for *group/descriptor* or *alias*; default=0 for *position*)

ddataset destination ADDE dataset name and absolute position; specify one of the following formats:

group/descriptor.position
alias.position

use only positive integers for *position*

Image Selection Keywords

DAY= selects from *sdataset* images with the specified day

DISt= maximum distance, in kilometers, from the satellite subpoint to include in the remap; use SMOOTH=YES and/or SPLINE= with a smaller value to round the edges of the output image

ID= selects from *sdataset* images with the specified NEXRAD station ID, e.g., MKX

MAG= *lmag emag* line and element reduction factor to apply to the source data before it's remapped; valid range is 1 through -999 where 1, 0, and -1 mean no reduction, -2 means a 2x reduction, etc; see the Remarks (*lmag* default=1, *emag* default=*lmag*)

SSIze= *line element* size of the source image region to remap; the region consists of the specified number of lines and elements centered at the location specified in the LATLON or STATION keyword (default=computed; see the Remarks)

ALL remaps the entire source image, even if only a small subset of the data is actually saved in the destination image with the SIZE keyword

TIMe= *btime etime* selects from *sdataset* images within the specified time range

TRACKing= **YES** tracks image if allowed by the server; tracking allows commands to receive data as it's being ingested by the server; commands complete when their data request has been fulfilled; tracking is only supported on some servers (e.g., Area, GVAR and MTSAT) and it can be disabled by the server administrator

NO does not track image so server only sends data that is currently available (default=value of MCTRACK environment variable; if MCTRACK is not set, default=value set by server or server administrator)

Output Keywords

BANd= image band number to remap (default=existing band for single-band images; no default for multiband images)

CF= **YES** creates a CF-compliant netCDF file (default)

NO creates a netCDF file in the previous (pre-2009.2) format, which is not CF-compliant; the CF=YES/NO keyword is valid only when writing to netCDF format destination datasets; it is ignored when writing to datasets of other formats

LATlon= *clat clon* center latitude and longitude for *ddataset* (default=center latitude and longitude of *sdataset*)

MERge= **NO** *sdataset* data overwrites the entire *ddataset* (default)

YES *sdataset* data overwrites *ddataset* only where images overlap; valid only with PRO=DEST

NAVtype= **LALO** *linc einc* use lat/lon navigation instead of the regular (native) navigation; this keyword is valid only if the source image is area or AVHRR Level 1b format; *linc* and *einc* specify the image line and element increments, e.g., NAVTYPE=LALO 4 3 requests lat/lon nav using the data from every fourth line and every third element (default=use native nav; if you specify LALO, lat/lon nav is used and the *linc* and *einc* defaults are the computed values that result in using the largest number of data points allowable without exceeding the max size of the AUX block)

PRO= **DEST** use the navigation of *ddataset* (default)

LAMB *slat1 slat2 slon*
Lambert conformal projection, where:

slat1 standard latitude (default=30)

slat2 standard latitude (default=50)

slon standard longitude (default=center longitude)

MERC *slat* Mercator projection; specify standard latitude (default=0)

MOLL *slon* Mollweide projection; specify standard longitude (default=0)

PS *slat slon*
polar stereographic projection, where:

slat standard latitude (default=60)

slon standard longitude (default=center longitude)

RADAR *rot* radar projection; specify rotation angle (default=0)

RECT rectilinear projection

SIN sinusoidal equal area projection

TANC *slat slon*
tangent cone projection, where:

slat standard latitude (default=30 if the center latitude is in the northern hemisphere; -30 if the center latitude is in the southern hemisphere)

slon standard longitude (default=center longitude)

RES= resolution of *ddataset* in km; not valid with PRO=DEST (default=1)

SAVesrc= **YES** writes *sdataset*'s directory information (including SS, DATE, TIME and BAND) to *ddataset*; valid only with PRO=DEST (default=**NO**)

SIZe= *line ele* number of lines and elements in *ddataset*; the *ele* value must be a multiple of 4; SIZE is not valid with PRO=DEST (default=480 640)

ALL creates an image large enough to include the entire geographical coverage of *sdataset*

SMOoth= *limb data*

provides limb or data smoothing

limb	**NO**	no limb smoothing (default)
	YES	provides limb smoothing
data	**NO**	no data smoothing (default)
	YES	provides data smoothing using a 2/3:1/3 weighting of the two closest pixels

SPLine= spline size in image pixels; the range is from at least 1 to 100; the range minimum may be computed based on source data (default=20)

STAtion= station ID to place at the center of the remapped image; specify as the ID, e.g., KMSN or YSSY, optionally followed by the station type in brackets, e.g., ARX[N] or ARX[V]; see the TYPE keyword in the STNLIST command for valid station types

WL= remaps band with image wavelength closest to the specified value; valid only with high spectral resolution data, e.g., Aqua AIRS

WN= remaps band with image wave number closest to the specified value; valid only with high spectral resolution data, e.g., Aqua AIRS

Remarks

IMGREMAP subdivides the destination image into boxes. The height and width of each of the boxes is determined with the SPLINE keyword, which defaults to 20 pixels. IMGREMAP performs a navigation transformation from line/element to latitutde/longitude at each of the spline box corners. It then determines the line/element coordinates in the source image that correspond to these same latitude/longitude values. Those corner values are used in a bi-linear interpolation scheme (which is much faster than a navigation transformation) to compute the line/element coordinate of each source image pixel it will use to fill the intermediate points in the spline box. This method ensures that each pixel in the destination image has an associated pixel from the source image.

If the destination image is lower resolution than the source image, the data are sampled. If the destination image is higher resolution, the pixel values are replicated. Only the 8-bit grayscale values are stored in the destination image. By default, there is no interpolation of the 8-bit values; however, the keyword SMOOTH=X YES performs an interpolation using a 2/3:1/3 weighting of the two closest pixels.

IMGREMAP uses the *sdataset* parameter and the DAY, TIME and ID keywords to select the source image to remap. It uses the *Output Keywords* to calculate the geographical region to extract from the source image for the remap. The calculation, however, sometimes underestimates or overestimates the size of the source image region needed for the remap. If that happens, use the SSIZE keyword to override the default calculation. For example, if the default calculation doesn't include the poles in the remap of a global image, specify SSIZE=ALL to ensure that they're included. The default calculation often overestimates the data region needed for a remap of polar-orbiting satellite data. To avoid this, you can use SSIZE with line and element values to choose a smaller source image region. Selecting more source image data than necessary does not affect the output image, but takes it longer to complete and adds to the cost if you pay for the data.

The MAG keyword can be used to further reduce the amount of source image data that is extracted and remapped. Use the MAG keyword only when specifying a destination image with less detail (lower resolution) than the source image. For example, if the source image is 1km resolution and you specify RES=30 for a 30km resolution destination image, you can also specify MAG=-16 to reduce the amount of data extracted and remapped by a factor of 16. It's best to pick a MAG value somewhat less than the destination image resolution because the quality of the destination image decreases as the MAG value approaches the resolution value.

When using MERGE=YES, all data in the source image (even pixels with zero data values) overwrite the collocated data in the destination image. Thus, all geographic locations in the destination image that are also in the source image will be replaced by the source image's values.

The radar projection is identical to the sinusoidal equal area projection, except that it allows you to rotate the destination image.

Display destination images with the IMGDISP command.

To create a destination dataset with a line prefix length and validity code of zero, use the IMGCOPY command with the keyword STYPE=VISR and SIZE=*line ele*; do not use SIZE=SAME.

For IMGREMAP to be successful using the PRO=DEST option, the following requirements (with the corresponding IMGLIST FORM=ALL output values in parenthesis) must be met:

- both the *sdataset* and *ddataset* must contain navigation
- the *ddataset* must be a single band of 1-byte data (*bytes per pixel:* in IMGLIST)

- the *sdataset* may be 2- or 1-byte data (*bytes per pixel:* in IMGLIST)
- the *ddataset*'s source type (*type:* in IMGLIST) must be **VISR**
- the *ddataset*'s calibration type (*cal type:* in IMGLIST) must be **BRIT**
- the *ddataset* must have a line prefix length (*PREFIX=* in IMGLIST) of **0** (zero)
- the *ddataset* must have a validity code (*valcod:* in IMGLIST) of **0** (zero)

You can use IMGREMAP to create a netCDF image file. To do so, the destination dataset must be NCDF-format and IMAGE-type. See the Remarks section in the DSSERVE command for more information.

Examples

IMGREMAP LOCAL/IMAGE.16 LOCAL/REMAP.4000

This entry remaps the image data in position 16 of dataset LOCAL/IMAGE into position 4000 of dataset LOCAL/REMAP. The source image replaces the destination image but uses its projection information.

IMGREMAP NOAA-12/HRPT NO-GOES/DATA.10 MERGE=YES PRO=DEST

This entry remaps the most recent image in dataset NOAA-12/HRPT into position 10 of dataset NO-GOES/DATA. The source image uses projection information from the destination image and replaces data only where the images overlap.

IMGREMAP G8/FDV G8/REMAP.1 RES=24 PRO=SIN

This entry remaps the most recent image in the G8/FDV dataset into the position 1 of dataset G8/REMAP. The destination image is remapped with a resolution of 24 kilometers in a sinusoidal equal area projection.

IMGREMAP G8/CONUSV G8/RADAR.2 RES=4 STATION=BOS PRO=RADAR 45

This entry remaps the most recent image in the G8/CONUSV dataset into position 2 of dataset G8/RADAR. The destination image is remapped in a radar projection, rotated 45 degrees. The image is centered at Boston, MA and has a resolution of four kilometers.

IMGREMAP WV.4 REMAP.4 PRO=LAMB 30 60 70 LATLON=34 78 RES=4

This entry remaps the image in position 4 of the dataset with the alias WV into position 4 of the dataset with the alias REMAP. The destination image is centered at 34° N and 78° W and has a resolution of 4 kilometers. The image is remapped in a Lambert conformal projection using standard latitudes of 30° and 60° N and a standard longitude of 70° W.

IMGREMAP GOES.1 A.2 PRO=PS 45 71 RES=4 SPLINE=40

This entry remaps the image in position 1 of the dataset with the alias GOES into position 2 of the dataset with the alias A. The destination image is remapped in a polar stereographic projection, with a standard latitude of 45° N and a standard longitude of 71° W. The resolution of the remapped data is four kilometers, and the spline value is 40.

IMGREMAP A/A.8016 A/A.4000 PRO=MERC 50 RES=2 SIZE=ALL

This entry remaps all of the image data in position 8016 of dataset A/A into position 4000 of dataset A/A. The destination image is as large as necessary to contain the entire remapped image: a Mercator projection, with a standard latitude of 50°N and a resolution of two kilometers.

IMGREMAP GOES/VIS LOCAL/AREA.23 BAND=1 DAY=#Y TIME=22:55 23:05 MAG=-8 RES=16 PRO=MERC LATLON=0 90

This entry remaps the image in dataset GOES/VIS with today's date and an image time between 22:55 and 23:05 into position 23 of dataset LOCAL/AREA. The destination image is in a Mercator projection centered at 0°N and 90°W with a resolution of 16 kilometers. The MAG=-8 keyword reduces the resolution of the data retrieved from the source image from 1km to 8km before remapping it to 16km resolution. This 8x reduction speeds up completion of the command with little effect on the quality of the destination image.

IMGRGB

Combines image and basemap into multi-banded RGB image.

Format

IMGRGB *source dest* **BASE= FILT=GENERIC** *[keywords]*
IMGRGB *tdiff ir dest* **BASE= FILT=TDIFF** *[keywords]*
IMGRGB *vis ir dest* **BASE= FILT=VISIR** *[keywords]*
IMGRGB *wv sst dest* **BASE= FILT=WVSST** *[keywords]*

Parameters

source source ADDE dataset name and position; specify one of the following formats:

group/descriptor.position

alias.position

a *position* greater than zero represents an absolute position in the dataset; a *position* less than or equal to zero represents a relative position in the dataset based on image time; for example, 0 is the most recent and -1 is the next most recent; to use default position, do not specify *.position* (no default for *group/descriptor* or *alias*; default=0 for *position*)

dest destination ADDE dataset name and absolute position; specify one of the following formats:

group/descriptor.position

alias.position

use only positive integers for *position*

tdiff ADDE dataset containing brightness value 0-255 that correspond to surface-ir temperature values of -10 to 100K; see the Remarks

vis ADDE dataset containing visible data

ir ADDE dataset containing infrared data

wv ADDE dataset containing water vapor data

sst ADDE dataset containing sea surface temperature values; see the Remarks

---

Keywords

BASEmap= *generic* ADDE dataset containing single-band basemap

red green blue ADDE datasets containing red, green and blue channels of basemap

BASEThreshold= maxium gray shade value from a basemap to be considered a map; if three datasets are specified in the BASEMAP= keyword, the value from the *red* channel is used for this keyword; if the gray shade value of the basemap is less than the threshold, the transparency of the source image is increased by 50%; this option is useful when a virtual graphic of a map outline has been burned into a basemap using the SVGA command (default=0)

CLOUDnocloud= minimum brightness value for *tdiff* dataset to be considered a cloud; the range is 0-255 where 0 makes the basemap disappear and 255 makes the clouds disappear (default=40)

EU= *source basemap* name of enhancement tables to apply to *source* and *basemap* images; *source* is used only with FILT=GENERIC and *basemap* is used only with BASE=*generic*

FILTer=	**GENERIC**	combines generic image and basemap; percentages of generic and basemap image are determined using the TRANSPARENCY keyword
	VISIR	combines visible, infrared and basemap imagery; percentages of visible and infrared data are determined by date and time of image
	TDIFF	combines infrared and basemap imagery using surface-ir temperature differences to determine cloud/no-cloud and transparency values; see the Remarks
	WVSST	combines water vapor, sea surface temperature and basemap imagery; transparency is determined by WV count using the enhancement table specified with WVEU keyword

IReu= name of enhancement table to apply to infrared image when using FILT=TDIFF

NAVele= number of elements between each navigation calculation; the range is 1-640 where higher numbers increase speed by decreasing the number of calculations, but also result in lower image quality (default=5)

SSTeu= name of enhancement table to apply to sea surface temperature image when using FILT=WVSST (default=**IMGRGB_SST.ET**)

TRACKing=	**YES**	tracks image if allowed by the server; tracking allows commands to receive data as it's being ingested by the server; commands complete when their data request has been fulfilled; tracking is only supported on some servers (e.g., Area, GVAR and MTSAT) and it can be disabled by the server administrator
	NO	does not track image so server only sends data that is currently available (default=value of MCTRACK environment variable; if MCTRACK is not set, default=value set by server or server administrator)

TRANSparency= *britlo brithi pctlo pcthi*

transparency values to assign low and high brightness levels when using FILT=GENERIC; if the value for *pctlo* is greater than 0, brightness values 0 to *britlo* are assigned transparency values 0 to *pctlo*; if the value of *pcthi* is less than 100, brightness values *brithi* to 255 are assigned transparency values *pcthi* to 100 (default=0 255 0 100)

WVeu= name of enhancement table to apply to water vapor image when using FILT=WVSST (default=**IMGRGB_WV.ET**)

Remarks and Examples

The topography, land/sea mask and NASA's Blue Marble Next Generation images for use with this command are available at the McIDAS Website at *http://www.ssec.wisc.edu/mcidas/basemaps/*

All source and basemap images must be the same size and cover the same geographical region.

Each option for the FILTER keyword is described below with an example.

GENERIC

This option uses a generic image with a basemap. With the TRANSPARENCY keyword the percentages of these two images is determined. The following example combines a NEXRAD image with NASA's Blue Marble Next Generation basemap.

```
IMGCOPY NEXRAD/BREF1 A/A.4000 ID=ARX SIZE=ALL
IMGCOPY A/A.4000 A/A.4001 SIZE=ALL
IMGCOPY A/A.4000 A/A.4002 SIZE=ALL
IMGCOPY A/A.4000 A/A.4003 SIZE=ALL
```

These commands copy the NEXRAD image into an AREA file, then create 3 other AREA files with the same size and geographical region.

```
IMGREMAP BBM/RED A/A.4001 PRO=DEST
IMGREMAP BBM/GREEN A/A.4002 PRO=DEST
IMGREMAP BBM/BLUE A/A.4003 PRO=DEST
```

These commands remap each Blue Marble Next Generation image component into the same region.

IMGRGB A/A.4000 A/A.4005 BASE=A/A.4001 A/A.4002 A/A.4003 FILT=GENERIC EU=BREF TRANS=16 255 95 100
IMGDISP A/A.4005 1 BAND=1
IMGDISP A/A.4005 2 BAND=2
IMGDISP A/A.4005 3 BAND=3
COMBINE 0 1 2 3

The IMGRGB combines the NEXRAD image with the basemaps with a specific transparency and a radar enhancement table. The IMGDISP commands and COMBINE command are used to display the resultant image, a radar display over a basemap.

VISIR

This option uses the date and time of the imagery to determine the percentages of visible and infrared to be used. The transparency is determined by combining the visible and infrared image information where warm/dark pixels are very transparent and cold/bright pixels are very opaque. This set of examples combines a topographic basemap with visible and infrared imagery.

IMGOPER BASEMAP/TOPOLAKES MY/AREAS.1 SIZE=2000 4000 MAG=-5 SCALE=0 6000 0 253

To create the basemap, you have to first decide on an elevation range then use IMGOPER to reduce the 2-byte data to 1-byte. The highest point in North America is roughly 6000 meters, so this IMGOPER scales the global basemap.

IMGREMAP MY/AREAS.1 MY/AREAS.2 STATION=KSTJ RES=11 PRO=SIN

This command remaps the basemap to a North American domain.

IMGREMAP EAST/FD MY/AREAS.4 STATION=KSTJ RES=11 BAND=1 TIME=17:45 PRO=SIN
IMGREMAP EAST/FD MY/AREAS.5 STATION=KSTJ RES=11 BAND=4 TIME=17:45 PRO=SIN

These commands remap the visible and infrared images over the desired domain.

IMGRGB MY/AREAS.4 MY/AREAS.5 MY/AREAS.6 BASEMAP=MY/AREAS.2 FILT=VISIR EU=X TOPO
IMGDISP MY/AREAS.6 1 BAND=1
IMGDISP MY/AREAS.6 2 BAND=2
IMGDISP MY/AREAS.6 3 BAND=3
MAP GRA=1-3
COMBINE 0 1 2 3
FRMSAVE 0 vis-ir-example.jpg TYPE=COMBINE FORM=JPG

The IMGRGB combines the basemap and satellite imagery with a topography enhancement. The IMGDISP, MAP and COMBINE commands are used to display the resultant image. The FRMSAVE saves the image as a 24-bit jpeg file.

TDIFF

This option uses the *tdiff* image to determine a cloud/no-cloud threshold and to determine the transparency of the infrared imagery on top of a basemap. The *tdiff* image, which must be created prior to running this command, must have brightness values 0-255 corresponding to surface-infrared temperature difference values -10 to 100. For example, commands like those below can be used to create the *tdiff* image.

GRDIMG RTGRIDS/NAM-USLC MY/AREAS.1 PARAM=T LEV=SFC DAY=#Y TIME=12 FHOUR=0 UNIT=K SCALE=320 200 0 255 MAG=10 10
IMGREMAP MY/AREAS.1 MY/AREAS.11 RES=8 PRO=SIN STA=KSTJ SIZE=600 800
IMGREMAP EASTL/CONUS MY/AREAS.2 BAND=4 RES=8 PRO=SIN STA=KSTJ TIME=11:45 SIZE=600 800
IMGOPER MY/AREAS.2 MY/AREAS.12 PROD=T UNIT=TEMP SCALE=320 200 0 255 SIZE=600 800
IMGOPER MY/AREAS.11 MY/AREAS.12 MY/AREAS.13 UNIT=T T FORM=ADD SCALE=-10 100 0 255 COEF=1 -1 SIZE=600 800

Once this image is created, this example can be used to create the infrared and basemap imagery using the surface-ir temperature differences.

IMGREMAP BBM/RED MY/AREAS.20 RES=8 PRO=SIN STA=KSTJ SIZE=600 800
IMGREMAP BBM/GREEN MY/AREAS.21 RES=8 PRO=SIN STA=KSTJ SIZE=600 800
IMGREMAP BBM/BLUE MY/AREAS.22 RES=8 PRO=SIN STA=KSTJ SIZE=600 800

These commands remap each Blue Marble Next Generation image component into a 600 by 800 sized AREA file in sinusoidal projection centered over St. Joseph, MO.

IMGRGB MY/AREAS.13 MY/AREAS.2 A/A.4005 FILT=TDIFF BASEMAP=MY/AREAS.20 MY/AREAS.21 MY/AREAS.22
IMGDISP A/A.4005 1 BAND=1;MAP
IMGDISP A/A.4005 2 BAND=2;MAP
IMGDISP A/A.4005 3 BAND=3;MAP
COMBINE 0 1 2 3

The IMGRGB combines the satellite imagery and temperature difference image over the basemaps. The IMGDISP commands and COMBINE command are used to display the resultant image, the GOES infrared imagery over the basemap.

WVSST

This option requires that you create a sea surface temperature image before running the command. Land values in the SST image must have a value of 0. The default enhancement table for SSTEU= assumes that SST temperature values 270 to 305 correspond to gray shades 0 to 149. Transparency values are determined by brightness values of the water vapor image and, if present, the enhancement table specified with WVEU=. The enhancement table should stretch only gray shades (i.e., do not use colors). This example uses a GOES water vapor image, the land/sea mask image, and gridded SST data.

IMGREMAP EASTS/CONUS A/A.4000 BAND=3 PRO=SIN STATION=KSTJ RES=11 SAVE=YES TIME=12 18
GRDIMG RTGRIDS/SST-GLME2 A/A.4004 MAG=5 SCALE=270 305 0 149 PARAM=TWAT DAY=#Y
IMGREMAP A/A.4004 A/A.4001 PRO=SIN STATION=KSTJ RES=11 SAVE=YES

These commands remap the GOES water vapor image and SST grid into common size and region (sinusoidal projection with the resolution of 11 km).

IMGREMAP BASEMAP/LSMASK A/A.4002 PRO=SIN STATION=KSTJ RES=11 SAVE=YES
IMGFILT A/A.4001 A/A.4002 A/A.4003 FILT=DIS 0 255 8 8 0 SCALE=0 255 0 255 SIZE=ALL

After remapping the land/sea mask image into the desired domain, the IMGFILT combines the land/sea mask with the SST image which leaves the land black and the appropriate sea surface temperatures over the water.

IMGRGB A/A.4000 A/A.4003 A/A.4005 FILT=WVSST BASEMAP=A/A.4100 A/A.4101 A/A.4102
IMGDISP A/A.4005 1 BAND=1;EG;MAP
IMGDISP A/A.4005 2 BAND=2;EG;MAP
IMGDISP A/A.4005 3 BAND=3;EG;MAP
COMBINE 0 1 2 3

The IMGRGB takes the SST image (with the land masked out) and combines it with the water vapor image. The IMGDISP commands and COMBINE command are used to display the resultant image.

IMGTRACK

Redisplays a loop of images centered at a selected feature.

Format

IMGTRACK *[keyword]*

Keywords

MAG=	*lmag emag*	line and element magnification factors to use when the images are redisplayed; see the Remarks (default=1 *lmag*)
TRACKing=	**YES**	tracks image if allowed by the server; tracking allows commands to receive data as it's being ingested by the server; commands complete when their data request has been fulfilled; tracking is only supported on some servers (e.g., Area, GVAR and MTSAT) and it can be disabled by the server administrator
	NO	does not track image so server only sends data that is currently available (default=value of MCTRACK environment variable; if MCTRACK is not set, default=value set by server or server administrator)

Remarks

IMGTRACK marks a feature on each frame of a loop set with the LS or LB command, then it uses the IMGDISP command to redisplay the images with the feature at the center of each frame. Use IMGTRACK to show a Lagrangian display of a moving feature.

The MAG keyword is used to increase or decrease the resolution when the images are redisplayed. The factors specified in MAG are relative to the originally-displayed image's resolution, not the image file's resolution. Thus MAG keyword values greater than 1 produce a blowup of the displayed image, and values less than 0 produce a blowdown. MAG=1 (the default) redisplays at the same resolution.

The keyword TRACKING= has a different meaning that the command name. The keyword refers to requesting whether the server sends only the data that was available when the command was issued (tracking=no), or to wait until all of the requested data is available before completing the request (tracking=yes). The "track" in the command name refers to tracking a feature on a loop of images and redisplaying them so that the feature appears at the center of each frame.

Examples

IMGTRACK

This entry prompts you to step through each frame of the current loop, marking the desired feature by moving the cursor to the feature's location and pressing the middle mouse button. When you've stepped through all the frames in the loop, press **Alt G** to redisplay the images with the feature at the center of each frame. Then press **Alt Q** or the center and right mouse buttons to exit the program.

IMGTRACK MAG=3 2

This entry is the same as the one above, except that it uses a line blowup factor of three and an element blowup factor of two when redisplaying the images.

ISENT

Calculates theta surface levels for upper air data.

Format

ISENT *sfile dfile day time btha etha inctha [keyword]*

Parameters

sfile	MD file number of the mandatory level data (default=last digit of the current Julian day plus 10)
dfile	destination MD file; must not exist before running ISENT (no default)
day	year and day of the upper air data, YYDDD (default=current)
time	time of the upper air data observation, HHMM (default=0 UTC if the current time is less than 13:30 UTC, else 12 UTC)
btha	beginning theta value (default=280 K)
etha	ending theta value (default=350 K)
inctha	increment of the theta value (default=10 K)

Keyword

SIG=	MD file number of the significant level data (default=depends on *sfile*; see the Remarks)

Remarks

The ISEN schema must be registered on your workstation before you can use this command. See commands LSCHE and SCHE for more information. Also see Registering MD File Schemas.

A maximum of 18 theta levels can be calculated. Theta levels contain pressure, mixing ratio, u- and v-components, delta pressure, Montgomery stream function, and the theta value.

The *sfile* parameter and the SIG keyword specify the mandatory and significant level MD file numbers respectively. If *sfile* is in the default range of 11 to 20, the default for SIG is the value of *sfile* plus ten. If *sfile* is outside the range of 11 to 20, the default for SIG is the value of *sfile* plus one. For example, if *sfile* is 14, the default for SIG is 24; if *sfile* is 2500, the default for SIG is 2501.

To view data from the created MD file, use command MDX with a sort condition like the following: SORT=TIME 12 THA 300. This sort condition is for data at 12 UTC; the isentropic surface is 300 K. See command MDX for more information.

Examples

ISENT 14 150

This entry calculates the theta surface levels for all data in MD file 14 and stores the results in MD file 150.

ISENT X 1000 X 0 290 340

This entry calculates the theta surface levels between 290 and 340 for 0 UTC on the current day, and stores this information in MD file 1000.

J

Toggles graphics frames to and from the loop control system.

Remarks

To toggle graphics frames to and from the loop control system,

Press: **Alt J**

or

Type: **J**

Press: **Enter**

Use command J only on McIDAS-X sessions with independent graphics. See the TERM command for more information.

The loop control system consists of the A, B, L, J, Y, LB, and LS commands.

You can connect or disconnect graphics frames from the loop control keys independently of the image frames.

K

Toggles image frames on and off.

Remarks

To toggle image frames on and off,

Press: **Alt K**

or

Type: **K**

Press: **Enter**

This command allows you to toggle image frames on and off to provide a better view of the graphics.

See Also

Command W toggles the graphics frame on and off.

KILL

Terminates McIDAS commands.

Format

KILL *pid1 . . pidn*

Parameter

pid1 . . pidn PID numbers of the commands to terminate

Remarks

If you don't know the PID number, use the question mark (?) command.

See Also

The / (Slash) command performs the same function as KILL.

Examples

KILL 273

This entry terminates the command running under PID number 273.

KILL 27347 27348 34272

This entry terminates the commands running under PID numbers 27347, 27348, and 24272.

L

Toggles frame looping on and off.

Remarks

To start automatic frame looping,

Press: **Alt L**

or

Type: **L**

Press: **Enter**

Pressing L a second time stops the looping at the first frame of the loop.

The L command works only if the loop control system is connected to the frame system.

See Also

The Y and J commands connect (and disconnect) the image and graphics frames to the loop control system. The LS or LB commands set the frame sequence for the loop.

Refer to *Chapter 2, Introduction*, for more information about the loop control system.

LB

Sets image or graphics frame loop bounds.

Formats

LB *bframe eframe*
LB G *bframe eframe*

Parameters

bframe	beginning frame of the loop (default=previous *bframe*)
eframe	ending frame of the loop; must be larger than *bframe* (default=half the highest frame number for image frames; the highest frame number for graphics frames)
	MAX highest frame number on the workstation
G	sets the graphics frame loop bounds; valid only in sessions with independent graphics

---

Remarks

When you start McIDAS, the loop bounds are set to 1 through the highest frame number.

See Also

Use commands A, B, and L to advance, back up, and loop through the frames. Use the LS command to create random loop sequences. See *Chapter 2, Introduction*, for more information on the loop control system.

Examples

LB

This entry lists the current loop bounds.

LB 1 6

This entry sets the loop bounds to frames 1 through 6.

LB 6 MAX

This entry sets the loop bounds to frames 6 through the highest frame number.

LB G 8 11

For a session with independent graphics, this entry sets the graphics loop bounds to frames 8 through 11.

LEO

Finds and displays a polar orbiting satellite image over a point or region.

Format

LEO *map channel [keywords]*
LEO *station channel [keywords]*
LEO *lat lon channel [keywords]*
LEO *`city,st,co' channel [keywords]*

Parameters

map map name that defines the geographic region for which to display the image; see the MAP command for a list of valid map names, e.g., FL, MID, USA, EUR (default=value specified for the LOCATE keyword in the **LEO.USER /.SITE /.CORE** text file; the value is set to USA in **LEO.CORE**)

station 3- to 5-character station ID or 5-digit ID number at which to center the image

lat latitude at which to center the image

lon longitude at which to center the image

`city,st,co' city or town at which to center the image, where *city* is the city or town name, *st* is the two-letter state or province code for the specified city and *co* is the two-letter country code for the specified city; if *city* is a U.S. city or town and appears more than once in the state, you can identify the one you want by following the city/town with the county name in brackets, e.g., 'GERMANTOWN[RICHLAND],WI'

channel satellite channel to display; valid options are VIS, VEG, ICE, SW, IR or SST, where:
VIS = visible cloud and surface features (~ 0.6 μm),
VEG = visible land vegetation, ocean aerosols (~ 0.9 μm),
ICE = near-infrared surface, cloud phase (~ 1.6 μm),
SW = shortwave infrared low cloud/fog, fire (~ 3.9 μm),
IR = infrared surface/cloud top temperature (~11.0 μm),
SST = infrared surface/cloud temperature, low level water vapor (~12.0 μm)
(default=VEG if the image time is between one hour after sunrise and one hour before sunset; otherwise default=IR)

Keywords

GRAphic= graphics frame number to display the map (default=current)

IMAge= image frame number to display the image (default=current)

MCOlor= *pol cnty* graphics color levels for map's political boundaries and county boundaries (in U.S.); use positive numbers for solid lines, negative numbers for dashed lines, and 0 to suppress drawing (default=value specified for the MCOLOR keyword in the **LEO.USER /.SITE /.CORE** text file; the value is set to **7 8** in **LEO.CORE**)

OFF= *op1 . . opN* set specified options to OFF; see the ON keyword for the list of options and their defaults

ON= *op1 . . opN* set specified options to ON; the default for all options is the value specified for the corresponding keyword in the **LEO.USER /.SITE /.CORE** text file; the options for the OFF= and ON= keywords and their default setting in **LEO.CORE** are:

- **DEBUG** list output from generated commands in text frame (default=OFF)
- **ECHO** list generated commands in text frame (default=ON)
- **FILTER** apply filter or stretch to image (default=OFF)

MAPS display map over image (default=ON)

RES= resolution, in kilometers, to display image at (default=resolution of the primary dataset if the *station*, *lat lon*, or '*city,st,co*' command format is used; default=calculated to match the geographic region if the *map* command format is used)

ZENith= maximum allowed angle, in degrees, between the satellite sub path and the specified image load point location; a value of 0 means that the satellite passed directly overhead (default=value specified for the ZENITH keyword in the **LEO.USER /.SITE /.CORE** text file; the value is set to 45 in **LEO.CORE**)

Remarks

The LEO command uses the **LEO.***, **MAP.*** and **LEODATA.*** context files (where * = **.CORE**, **.SITE** and **.USER**) to determine which data to use and how to display the data. Default parameter and keyword values are defined in the **LEO.*** files. Regional map definitions are defined in the **MAP.*** files. Satellite datasets, their instrument channel and resolution information, and the navigation datasets used to compute overpass day/time estimates are defined in the **LEODATA.*** files. The datasets in **LEODATA.CORE** may not be appropriate for users who do not access their satellite data through the SSEC Data Center. Those users may need to create a **LEODATA.USER** or **.SITE** file in order to use LEO.

The **.CORE** files are supplied with McIDAS-X and should not be modified. You can, however, change their settings to your preferences. To do so, copy the **.CORE** file from **~mcidas/data** to your **$HOME/mcidas/data** or another **MCPATH** directory, change its extension from **.CORE** to **.USER** or **.SITE**, then edit the settings as needed. The **.USER** file overrides the **.SITE** file, which overrides the **.CORE** file. This order of precedence ensures that your local settings (in the **.USER** and/or **.SITE** file) override the settings in the **.CORE** file.

All images are remapped based on a derived projection and resolution. The projection is determined by the center point latitude of the image region requested by the command. For center point latitudes -30° to 30°, the projection is Mercator (MERC). For center point latitudes -50° to -30° and 30° to 50°, the projection is Lambert Conformal (LAMB). For all other center point latitudes, the projection is Polar Sterographic (PS).

Examples

LEO

This entry finds the most recent image from the polar orbiting satellite datasets in the **LEODATA.*** file (**LEODATA.USER /.SITE /.CORE**) whose coverage includes the point or region specified in the LOCATE keyword in the **LEO.*** file (**LEO.USER /.SITE /.CORE**) and whose zenith angle to that point or region's center is within the limit set by the ZENITH keyword. It then displays the image at the base resolution of the channel if a point was specified, or at the resolution needed to cover the domain if a region was specified. For most users, the region will be the United States because the LOCATE keyword is set to USA in **LEO.CORE**. For this and all examples below, unless the *channel* parameter is specified, LEO displays the land vegetation channel (~0.9 μm) during daylight hours and displays the infrared channel (~11.0 μm) during nighttime hours.

LEO FL

This entry finds the most recent image from the polar orbiting satellite datasets in the **LEODATA.*** file whose coverage includes the region defined by the map named FL and whose zenith angle to the region's center point is within the limit set by the ZENITH keyword. It then displays the image at the resolution needed to exactly cover the region. For most users, the region will be the state of Florida because that's how FL is defined in the **MAP.CORE** file.

LEO EUR SW

This entry finds the most recent image from the polar orbiting satellite datasets in the **LEODATA.*** file whose coverage includes the region defined by the map named EUR and whose zenith angle to the region's center point is within the limit set by the ZENITH keyword. It then displays the shortwave infrared channel (~3.9 μm) at the resolution needed to exactly cover the region. For most users, the region will be Europe because that's how EUR is defined in the **MAP.CORE** file.

LEO YSSY ZENITH=20 ON=FILTER

This entry finds the most recent image from the polar orbiting satellite datasets in the **LEODATA.*** file whose coverage includes Sydney, Australia within 20° of zenith. It then displays the image at the base resolution of the channel, centered on Sydney, and applies a filter to provide optimum contrast.

LEO 27 -122

This entry finds the most recent image from the polar orbiting satellite datasets in the **LEODATA.*** file whose coverage includes 27° North, 122° East within the zenith angle limit set by the ZENITH keyword. It then displays the image at the base resolution of the channel, centered on that point.

LEO `NEW YORK,NY'

This entry finds the most recent image from the polar orbiting satellite datasets in the **LEODATA.*** file whose coverage includes New York, New York within the zenith angle limit set by the ZENITH keyword. It then displays the image at the base resolution of the channel, centered on New York.

LEO `MADRID,,ES' IR ZENITH=30 RES=3

This entry finds the most recent image from the polar orbiting satellite datasets in the **LEODATA.*** file whose coverage includes Madrid, Spain within 30° of zenith. It then displays the infrared channel (~11.0 μm) at 3 km resolution, centered on Madrid, Spain. Note that two commas appear between the city (Madrid) and country code (ES) because if only one comma is specified, it's assumed to be a state or province code.

LOGOFF

Logs you off the McIDAS session.

Remarks

The LOGOFF command logs you off the McIDAS session so the session no longer has an identifying user name and project number. LOGOFF also clears the command history so you cannot recall previously entered commands with the & (Ampersand) and ^ (Caret) commands.

See Also

The LOGON command logs you onto the McIDAS session. The EXIT command ends the McIDAS session and closes its associated windows.

LOGON

Logs you on to the McIDAS session.

Format

LOGON *user project init*

Parameters

user your user name; see the Remarks

project your 4-digit project number; see the Remarks

init initialization state to set; specify one of the following:

X logs you on and runs the GD INI command to set graphics parameters to initialized values; no other actions are performed

I logs you on and initializes your McIDAS session; see the second example below for a complete list of initialization processes

. logs you on, runs the GD INI command, restores the string table with the name *user*, and runs all strings in that table with names that begin with *user*

string logs you on, runs the GD INI command, restores the string table named *string*, and runs all strings in that table with names that begin with *string*

Remarks

You should always logon because your user name and project number are recorded with scheduler entries (SKE command), string tables (TU command), and some data files (e.g., grid, and MD files). The user name and project number are also checked when a scheduler entry is run and when you request data from a remote server. At startup or after running the LOGOFF command, the user name is set to USER and the project number is set to 0. If you fail to log on with a valid name and project number, you may be prevented from accessing data on some remote servers.

Examples

LOGON TOM 1234

This entry logs you on to the McIDAS session as user TOM with project number 1234. It also runs GD INI to set the graphics parameters (width, dash and gap) to their initialized values. Otherwise the McIDAS session is left in its present state.

LOGON TOM 1234 I

This entry logs you on to the McIDAS session as user TOM with project number 1234, and initializes the McIDAS session as follows:

- clears the string table and command history
- sets image frame loop bounds from 1 to 1/2 the highest frame number
- runs DR INI to set the dwell rates to nine units for the first frame and six units for all subsequent frames
- runs GD INI to set the graphics parameters to their initialized values
- runs GU REST X on all frames to restore the default graphics color levels
- runs EU REST X on all frames to restore the default image enhancement
- runs CUR to restore the default 31 x 31 pixel red crosshair cursor

LOGON TOM 1234 .

This entry logs you on to the McIDAS session as user TOM with project number 1234. It also runs GD INI to set the graphics parameters to their initialized values. It runs TU REST TOM to restore the string table TOM, and runs all strings in that table with names that begin with TOM, for example, TOM, TOMSDATA, TOMLIST.

LOGON TOM 1234 PLOT

This entry logs you on to the McIDAS session as user TOM with project number 1234. It also runs GD INI to set the graphics parameters to their initialized values, runs TU REST PLOT to restore the string table PLOT, and runs all strings in that table with names that begin with PLOT for example, PLOT, PLOTWX, PLOTTEMP.

LS

Creates random loop sequences for image and graphics frames.

Formats

LS *frame1 frame2 frame3 . . framen [keyword]*
LS G *frame1 frame2 frame3 . . framen [keyword]*
LS O *frame1 frame2 frame3 . . framen [keyword]*

Parameters

frame1 . . n	frame numbers to include in the loop; see the Remarks
G	sets the graphics frame loop sequence; valid only in sessions with independent graphics
O	sets the opposite frame loop sequence

Keyword

Update=	**Y**	automatically rotates the loop's frame sequence each time a new image is displayed with the IMGDISP command, moving the new image to position one of the loop
	N	does not rotate the frame sequence (default)

Remarks

For a sequential range of frames, use the format *frame***-***frame* B*inc*. *frame***-***frame* specifies the beginning and ending frames, and *inc* is the increment (the dash means "to" and the B means "by"). The default increment is 1. For example, LS 1-5 sets the loop sequence to 1, 2, 3, 4, 5; LS 1-5B2 sets the loop sequence to 1, 3, 5; LS 5-1B-2 sets the loop sequence to 5, 3, 1. You can also mix formats; LS 1 3 7-19B3 25 sets the loop sequence to 1, 3, 7, 10, 13, 16, 19, 25.

When you set the frame loop, the opposite frame loop is automatically set in the following manner. If frame *n* is less than or equal to (*max frames* /2), it corresponds to the opposite frame *n*+(*max frames* /2). If frame *n* is greater than (*max frames* /2), it corresponds to the opposite frame *n*-(*max frames* /2).

To override the opposite frame loop default, use the LS O command.

See Also

Use commands A, B, and L to advance, back up, and loop through the frame sequence. See *Chapter 2, Introduction*, for more information about the loop control system.

Examples

LS

This entry lists the current loop sequences.

LS 1 4 2 5

This entry sets the loop sequence in the order 1, 4, 2, 5.

LS 1-16

This entry sets the loop sequence to frames 1 through 16.

LS 1 16

This entry sets the loop sequence to frames 1 and 16 only.

LS 1 3 7-19B2 23

This entry sets the loop sequence to frames 1, 3, 7, 9, 11, 13, 15, 17, 19, 23.

LS O 1-4 8 9 3-1

This entry sets the opposite loop sequence to frames 1, 2, 3, 4, 8, 9, 3, 2, 1.

LS G 1-5 5-1

For a McIDAS-X session with independent graphics, this entry sets the graphics frame loop sequence to frames 1 to 5, then 5 to 1.

LSCHE

Lists an MD file schema.

Formats

LSCHE *schema version* **HED**
LSCHE FIND *key*

Parameters

schema	name of an MD file schema to list; the name is system dependent (no default)
	ALL lists all file schemas
version	schema version number to list (default=latest version)
	ALL lists all schema version numbers
key	name of a schema key; a schema entry point descriptor
HED	lists the file header only (default=lists header and keys)
FIND	lists the schemas containing the specified key

Remarks

The LSCHE command lists information about *registered* MD file schemas, i.e., present in the LW file SCHEMA. You can create local schemas using a text editor, then register them using the SCHE command. See the SCHE command for more information.

Examples

LSCHE IRAB

This entry lists the IRAB file schema.

LSCHE ISFC ALL

This entry lists all versions of the ISFC file schema.

LSCHE ISHP X HED

This entry lists the file header only for the ISHP file schema.

LSCHE FIND IDN

This entry lists all schemas that contain the key IDN (station ID number).

LVF

Lists virtual frame files.

Formats

LVF *string*
LVF *file [keyword]*

Parameters

string	lists only the virtual frame files beginning with this character string (default=lists all files)
file	name of the file to list; if no extension is specified, .PIX is automatically appended

---

Keyword

FORM=	**ALL**	lists all attributes and directory information for the virtual frame file

---

See Also

Use the SVF command to save virtual frame files. Use the RVF command to restore virtual frame files.

Examples

LVF

This entry lists all workstation virtual frame files with the extension .PIX.

LVF RAD

This entry lists the virtual frame files beginning with RAD.

LVF STORM FORM=ALL

This entry lists all attributes and directory information for the file **STORM.PIX**.

LWCON

Reverses the byte order of 4-byte values in a file to convert its format from little-endian to big-endian or vice-versa.

Format

LWCON *sfile dfile [keywords]*

Parameters

sfile source LW file to convert to the opposite format

dfile destination LW file name; if *sfile* is little-endian format then LWCON creates *dfile* in big-endian format; if *sfile* is big-endian format then LWCON creates *dfile* in little-endian format

---

Keyword

FLIp= ALL byte flips all words in the file (default=does not flip words consisting entirely of ASCII text)

Remarks

Use LWCON to convert the format of a file to the opposite format (little-endian to big-endian, or big-endian to little-endian). The file must consist of 4-byte integers and/or character data. This program will fail if the file contains integers representing valid 4-byte ASCII characters.

Examples

Suppose you create a map file named PERUMAP on a little-endian workstation. To use it in McIDAS-X you must convert it to big-endian format. To do so, enter this command:

LWCON PERUMAP OUTLPERU FLIP=ALL

This entry converts the file **PERUMAP**, which is in little-endian format, to a big-endian format file named **OUTLPERU** that can be used in McIDAS-X.

LWTODOS

Converts LW format files to ASCII text format files.

Format

LWTODOS *sfile* *dfile* **R**

Parameters

sfile	source LW file name
dfile	destination ASCII text file name; must be different than *sfile*
R	replaces the destination file if it already exists

Remarks

The LW file format is an 80-character card image, blank padded as needed, with no carriage returns or line feeds. The ASCII text file format contains variable-length lines of ASCII text with a carriage return at the end of each line.

Examples

LWTODOS SFCPLT SFCPLT.TXT

This entry converts the LW file named **SFCPLT** to an ASCII text file named **SFCPLT.TXT**.

LWU

LW file utility.

Formats

LWU COPY *sfile dfile bword eword*
LWU DEL *file*
LWU INFO *file*
LWU LIST *file bword eword*
LWU MAKE *file*
LWU POKE *file value word* **ASCII=**

Parameters

COPY	copies a file
DEL	deletes a file
INFO	lists the file and its size, in bytes
LIST	lists the words in a file
MAKE	creates a file
POKE	pokes a value into a file
sfile	source file name
dfile	destination file name
file	name of the file
bword	beginning word number to copy or list, the range is 0 to 8,000,000 (default=0)
eword	ending word number to copy or list, the range is 0 to 8,000,000 (default=last word in the file for the COPY option; default=value specified in *bword*+39 for the LIST option)
value	value to poke into the word (default=0)
word	number of the word to change (default=0)

Keyword

ASCII=	**YES**	pokes the specified *value* as ASCII characters; see the Remarks (default for all non-numeric *values*)

Remarks

Use command DMAP or LWU INFO to determine the size of the file.

By default, if the value specified with the POKE option is numeric (one or more digits, possibly prefaced with a + or a -), it is poked into the word as an integer; otherwise, it is poked into the word as ASCII characters. Use the ASCII=YES keyword to override the default when you need to poke a numeric value into the word as ASCII characters.

Caution!
Be very careful when using the POKE option; misuse can permanently damage the file.

Examples

LWU MAKE CLOUDS

This entry creates an LW file named **CLOUDS**.

LWU COPY CLOUDS NEWONE

This entry copies the LW file CLOUDS into file **NEWONE**.

LWU DEL CLOUDS

This entry deletes the LW file **CLOUDS**.

LWU INFO NEWONE

This entry lists the size of the LW file **NEWONE**.

LWU LIST LWTXT 10 20

This entry lists words 10 to 20 of the LW file **LWTXT**.

LWU POKE LWTXT 92067 64

This entry changes the value of word 64 in the LW file **LWTXT** to the integer value 92067.

LWU POKE LWTXT 92067 64 ASCII=YES

This entry changes the value of word 64 in the LW file **LWTXT** to the ASCII string 92067.

MAKFRM

Adds new frames to a McIDAS-X session.

Format

MAKFRM *nframe line ele*

Parameters

nframe	number of frames to add; the range is 1 to 500
line	number of lines; the range is 1 to 100,000 (default=number of lines in current frame)
ele	number of elements; the range is 1 to 100,000 (default=number of elements in current frame)

Remarks

You can add new frames to a session only if you allocate memory for additional frames by specifying the -e flag at startup or in your **.mcidasrc** file.

If you run MAKFRM without any parameters, it lists the amount of memory available for adding new frames.

The amount of memory required to add a new frame is slightly more than the number of lines times the number of elements. The figure is slightly higher because additional memory is required for enhancements, stretch tables, graphics tables, etc.

When you add new frames, they are appended to the end of the current frame list. They are not added to the current loop bounds. Use the LS command to add them to the current loop. If the new frames are a different size and you display them using SF, A, or B, use the R command to resize the Image Window to the size of the frames.

Examples

MAKFRM

This entry lists the amount of memory available for adding new frames.

MAKFRM 1 300 400

This entry creates a single frame that is 300 lines by 400 elements.

MAKNAV

Creates a navigation entry for an area.

Format

MAKNAV *area [keywords]*

Parameter

area area number to navigate (default=0)

Keywords

LAMB= *npl npe slat1 slat2 qlon size pole*

Lambert conformal projection

npl pole image line

npe pole image element

slat1 standard latitude 1 (default=30)

slat2 standard latitude 2 (default=60)

qlon normal or center longitude

size size of one pixel, in km, at standard latitudes

pole **N** North Pole (default)

S South Pole

MERC= *eql eqe slat qlon size*

Mercator projection

eql equator image line

eqe equator image element

slat standard latitude (default=0)

qlon normal or center longitude

size size of one pixel, in km, at standard latitude

MOLL= *eql eqe qlon size*

Mollweide projection

eql equator image line

eqe equator image element

qlon normal or center longitude

size size of one pixel, in km, at standard latitude

PLANET= planet to navigate (default=**EARTH**)

PS= *npl npe slat qlon size pole*

polar stereographic projection

npl pole image line

npe pole image element

slat standard latitude (default=60)

qlon normal or center longitude

size size of one pixel, in km, at standard latitude

pole **N** North Pole (default)

S South Pole

RADR= *cenl cene clat clon size rot*

radar projection

cenl radar site center line

cene radar site center element

clat radar site latitude

clon radar site longitude

rot image rotation (default=0)

RECT= *lin lat ele lon deglin degele*

rectilinear projection

lin image line number

lat image line latitude

ele image element number

lon image element longitude

deglin degrees latitude per image line

degele degrees longitude per image element

SIN= *cenl cene clat clon size*

sinusoidal equal area projection

cenl radar site center line

cene radar site center element

clat radar site latitude

clon radar site longitude

size size of one pixel, in km, at standard latitude

TANC= *npl npe slat glon size*

tangent cone projection

npl pole image line

npe pole image element

slat standard latitude (default=60)

qlon normal or center longitude

size size of one pixel, in km, at standard latitude

See Also

Use the IMGREMAP command to remap into the map projection created by MAKNAV.

Example

The example below copies a GOES satellite image and remaps it into a Mercator projection in local area 6000.

IMGCOPY GOES/VIS.5 LOCAL/AREA.6000 STA=KLAX SIZE=480 640 STYPE=VISR BAND=1

This entry copies the image in position 5 in the dataset GOES/VIS to position 6000 in dataset LOCAL/AREA. The STYPE=VISR keyword converts the image to 1-byte data with source type VISR and calibration type BRIT.

IMGCHA LOCAL/AREA.6000 ELE=1 LIN=1 RES=8 8

This entry sets the appropriate line and element resolutions and coordinates for the desired projection.

MAKNAV 6000 MERC=2000 2000 0 90 1

This entry generates a Mercator projection codicil entry for local area 6000.

IMGREMAP GOES/VIS.5 LOCAL/AREA.6000 PRO=DEST

This entry remaps the image in position 5 in the dataset GOES/VIS into position 6000 in dataset LOCAL/AREA.

MAP

Displays a map and latitude/longitude lines on a graphics frame.

Formats

MAP *map color1 [keywords]*
MAP *map color1* **LALO** *color2 [keywords]*

Parameters

map one of the map names listed below, or a predefined map name in the **MAP.SITE** or **MAP.USER** file; see the Remarks (default=SAT)

DEF	draws a map of the domain defined with the LAT and LON keywords using the projection specified with the PRO keyword; uses the **US*.MAP** files if the STATE, COUNTY, or ZONE keyword is specified, otherwise uses the **OUTLSUPU** map file for the continental USA, **OUTLUSAM** for North America, and **OUTLHRES** for all other areas
EUR	draws a map of Europe's coastal and political boundaries; latitude=20 80, longitude=-60 30; uses map files **OUTLSUPW** and **OUTLHPOL**
EURB	draws a map of Europe's political boundaries; latitude=20 80, longitude=-60 30; uses map file **OUTLHPOL**
EURO	draws a map of Europe's coastal boundaries; latitude=20 80, longitude=-60 30; uses map file **OUTLSUPW**
H	draws a map of North America's coastal and political boundaries contained in the frame's preexisting navigation; uses map file **OUTLUSAM**
L	draws a map of the world's coastal boundaries contained in the frame's preexisting navigation; uses map file **OUTLHRES**
LALO	draws latitude/longitude lines using the frame's preexisting navigation
MID	draws a map of the U.S. Midwest; latitude=35.7 50.4, longitude=81.2 104.6; uses map file **OUTLUSAM**
NA	draws a map of North America's coastal boundaries; latitude=5 80, longitude=55 179; uses map file **OUTLSUPW**
PO Abbrev	draws a map of the U.S. state identified by its two-letter state post office abbreviation; uses map file **OUTLUSAM** for AK, **OUTLSUPU** for all other states
POLI	draws a map of world political boundaries; uses map file **OUTLHPOL**
PRC	draws a map of Asia's coastal boundaries; latitude=15 55, longitude=-145 -65; uses map file **OUTLSUPW**
PRCB	draws a map of Asia's political boundaries; latitude=15 55, longitude=-145 -65; uses map file **OUTLHPOL**
SAT	draws a map of the locations contained in the frame's preexisting navigation; uses the **US*.MAP** files if the STATE, COUNTY, or ZONE keyword is specified, otherwise uses the **OUTLHRES** map file
USA	draws a map of the continental USA; latitude=23 56, longitude=66 126; uses map file **OUTLUSAM**
VH	draws a map of continental USA's coastal and political boundaries contained in the frame's preexisting navigation; uses map file **OUTLSUPU**
WIS	draws a map of Wisconsin; latitude=41 48, longitude=85 95; uses map file **OUTLSUPU**
WORL	draws a map of the world's coastal boundaries; used with the pseudo-Mercator projection; uses map file **OUTLSUPW**

color1 graphics color level for the map; use positive numbers for solid lines, negative numbers for dashed lines, or zero (0) to suppress map drawing (default=value specified for the MCOLOR keyword in the **MAP.USER /.SITE /.CORE** text file; the value is set to 1 in **MAP.CORE** for all maps except EUR, which is set to 1 for coastal boundaries and -1 for political boundaries)

LALO draws latitude/longitude lines with a map (default=does not draw lines)

color2 graphics color level for latitude/longitude lines; use positive numbers for solid lines and negative numbers for dashed lines (default=2)

Keywords

ACC= accuracy for the latitude/longitude lines; the range is 1 to 10 (default=4)

BOX= **YES** *color* draws a box around the map in the specified graphics color level (default for all except SAT, L, H, VH and X; default color=1)

NO does not draw a box

COUnty= *c1 . . cN* list of counties to draw; use the STATE keyword to limit counties to a specified state; the county name must be in single quotes if it contains spaces, do not include punctuation, for example, specify 'ST LUCIES' for St. Lucie's county; you cannot use the COUNTY keyword with the ZONE or MARINE keywords; see the Remarks

ALL draw all counties

NONE draw no counties (default)

DAShset= *dlength gcolor glength*

sets dashed line characteristics when a negative number is specified for *color1* or *color2*

dlength length of dashes (default=current GD command setting)

gcolor graphics color level of gaps in dashed lines; specify 255 for transparent (default=current GD command setting)

glength length of gaps (default=current GD command setting)

DOMain= **YES** draw all borders in the map's domain using the selected or specified **OUTL*** file in addition to drawing the borders specified in the COUNTY, STATE and ZONE keywords (default=NO, meaning draw only the borders specified in the COUNTY, STATE and ZONE keywords)

ELE= *min max* TV element extents (default depends on frame size)

FILe= name of the map file used to determine the map's detail and components (default=appropriate **OUTL*** file for the map's geographic region); the following map files are supplied with McIDAS-X:

File Name	Comparitive Resolution	Geographical Coverage
OUTLANT	high	Antarctica
OUTLARST	high	Argentinian states
OUTLAUSS	high	Australian states
OUTLAUST	high	Australian region
OUTLAUSW	very high	Australia's Lake Eyre
OUTLEURO	high	European region
OUTLHPOL	medium	world political boundaries
OUTLHRES	high	world coastal boundaries
OUTLSUPU	very high	USA with states and major waterways
OUTLSUPW	medium	world coastal boundaries
OUTLUSAM	high	North America with states and provinces

GRA= *f1 . . fN* graphics frame number(s) to display the output; see the Remarks (default=current)

IMA= image frame number to use for navigation (default=current)

INT= *lat lon* intervals for latitude/longitude lines (default=computed)

LABel= *scheme clat clon card*

defines the format and colors of latitude and longitude labels (default=WESTPOS 5 4 NO)

scheme labeling scheme; specify one of the five options below; see Remarks for details
WESTPOS - Western longitudes positive, Eastern longitudes negative
EASTPOS - Eastern longitudes positive, Western longitudes negative
ALLPOS - all latitudes and longitudes positive
360E - longitudes 0 - 360, with values increasing eastward
360W - longitudes 0 - 360, with values increasing westward

clat graphics color level for latitude labels (default=5)

clon graphics color level for longitude labels (default=4)

card specify YES or NO as described below
YES - includes cardinal directions (N, S, E, W) with latitude and longitude labels
NO - does not include cardinal directions (default)

NO does not label latitude and longitude lines

LAT= *lat1 lat2* latitudes of the points at the lower-right and upper-left corners of the map (default=-90 90 for DEF maps with mercator projection)

LIN= *min max* TV line extents (default depends on frame size)

LLInset= *slat nlat elon wlon lat_int lon_int line_col*
draws latitude and longitude lines within the specified inset region, where:

slat southern latitude of inset region

nlat northern latitude of inset region

elon eastern longitude of inset region

wlon western longitude of inset region

lat_int spacing interval for latitude lines (default=computed)

lon_int spacing interval for longitude lines (default=computed)

line_col graphics color level of inset lat/lon lines (default=color level of main lat/lon lines if they are requested; otherwise default=2)

LLMark= **YES** *lat lon color size sym*
draws symbols, letters or numbers at lat/lon intersections determined by *lat* and *lon*, where:

lat latitude interval (default=calculated)

lon longitude interval (default=calculated)

color graphics color level (default=color of lat/lon lines or 2 if lat/lon lines are not requested)

size height of symbols in pixels; the range is 3-30 in multiples of 3 (default=6)

sym symbol, letter or number to draw (default=**+**)

NO does not draw symbols at lat/lon intersections (default)

LON= *lon1 lon2* longitudes of the points at the lower-right and upper-left corners of the map; if you choose a CONF projection and specify SCALE, you cannot specify values for *lat2* and *lon2*; if you specify only *lat1* and *lon1*, the map is centered at that point; defaults for SAT projections are the range taken from the image or map geographical limits; for CONF and MERC projections, defaults are latitude -90 to +90, longitude -180 to +180

LSIze= height of latitude and longitude labels in pixels (default=7)

MARine= *m1 . . mN* list of marine regions to draw, or used to limit the ZONE keyword; use one or more of the 2-letter abbreviations below, ALL or NONE; you cannot use both the MARINE and COUNTY keywords; see the Remarks

valid abbreviations are:

AM	Atlantic - mid	AN	Atlantic - North
GM	Gulf of Mexico	LC	Lake St. Claire
LE	Lake Erie	LH	Lake Huron
LM	Lake Michigan	LO	Lake Ontario
LS	Lake Superior	PH	Pacific - Hawaii
PK	Pacific - Alaska	PZ	Pacific Zone
SL	St. Lawrence River		

ALL	use all marine regions
NONE	draw no marine regions (default

PRO= **CONF** draws the map in a polar stereographic or general Lambert conformal projection; see the *Projections* section in the Remarks

MERC draws the map in a pseudo-Mercator projection (default for all *map*s except SAT, H, VH, L and LALO)

SAT draws the map using the frame's preexisting navigation (default if *map* is not specified, or *map* is SAT, H, VH, L or LALO)

SCAle= map scale for a CONF projection; scale values in the range up to 1000 are multiplied by 1,000,000 (default=no scale)

SLAt= *lat1 lat2* standard latitudes for PRO=CONF projection; enter one value for a polar stereographic projection or two values for a general Lambert conformal projection (*lat1* default=60 if the center of the map is in the Northern Hemisphere, -60 if it's in the Southern Hemisphere; *lat2* default=*lat1*); see the *Projections* section in the Remarks

SLOn= standard longitude for a CONF projection only (default is computed)

SPAn= **LOW** low line-spanning tolerance; use this keyword if your MAP command draws extra lines with the map

MED medium line-spanning tolerance (default)

HIGH high line-spanning tolerance; use this keyword if your MAP command draws a map with missing line segments

STate= *s1 . . sN* list of states to draw, or used to limit the COUNTY or ZONE keyword; use the 2-letter state post office abbreviation; see the Remarks

ALL use all states (default if the COUNTY or ZONE keyword is specified and the MARINE keyword is not specified)

NONE draw no states (default in all other cases)

WIDth= graphics line width (default=value specified for the WIDTH keyword in the **MAP.USER /.SITE /.CORE** text file; the value is set to X for all maps in **MAP.CORE**, meaning use the current GD command setting)

ZONe= *z1 . . zN* list of National Weather Service state and/or marine zone numbers to draw; use the STATE and/or MARINE keyword to limit zones to the specified states and/or marine regions; you cannot use both the ZONE and COUNTY keywords; see the Remarks

ALL draw all NWS zones (default if the MARINE keyword is specified)

NONE draw no NWS zones (default if the MARINE keyword is not specified)

Remarks

The MAP command uses two sets of map files. The **US*.MAP** files (for example, **USCOUNTY.MAP**) are used only when the STATE, COUNTY, MARINE or ZONE keyword is specified. The **OUTL*** files (for example, **OUTLSUPU**) are used in all other cases. The **US*.MAP** files include all U.S. counties and National Weather Service zones and marine regions, and slightly different state boundaries than the **OUTL*** files. The **US*.MAP** files are derived from the National Weather Service AWIPS Map Database (http://www.nws.noaa.gov/geodata/).

If you want to draw a map using the frame's current navigation, specify L, H, VH or SAT (or X, since SAT is the default) for the *map* parameter. For example, running MAP with no parameters or keywords draws a map using the frame's navigation. MAP SAT STATE=KS draws the state of Kansas if it's present in the frame's navigation.

If you want to draw a map that defines new navigation for the frame, specify one of the predefined maps (for example, MID, USA, WORL) or DEF for the *map* parameter. The LAT and LON keywords define the domain for the DEF map. If you also include the STATE, COUNTY, ZONE or MARINE keyword, only the specified states, counties, zones or marine regions are drawn, if they're in the domain.

The STATE, COUNTY, ZONE, and MARINE keywords are used to draw only the specified states, counties, zones and marine regions using the **US*.MAP** files. These keywords do NOT define the map's domain. Domain is defined with the positional parameter *map*, using a predefined map, SAT, MID, USA, for example, or using DEF with the LAT and LON keywords. For example, MAP MID STATE=IA defines a domain consisting of the Midwest, but only draws the state of Iowa. If you then run MAP SAT STATE=ALL, all states in the domain are drawn using the **USSTATE.MAP** file.

When you specify the COUNTY or ZONE keyword, only the counties or zones are drawn; the outlines of the states and/or marine regions they're in are not drawn. Specify the STATE keyword without the COUNTY or ZONE keywords to draw state outlines. Specify the MARINE keyword without the ZONE keyword to draw marine region outlines and the zones within them.

You can specify a single frame or multiple frames in the GRA keyword using the list format (*frame1 frame2 frame3. .*) and the range format (*bframe-eframe*). For example, a map is displayed on frames 1 through 5 with any of the following: GRA=1 2 3 4 5, GRA=1-5, GRA=1 2-4 5.

Frame-oriented maps (H, L, LALO, SAT, or VH in the *map* parameter) always use the navigation of the single frame specified in the IMA keyword, even if multiple frames are specified in the GRA keyword.

If you have multiple panels on a frame, you can display maps on a range of panels by specifying beginning and ending panel numbers with the global keyword PAN. For example, MAP NA PAN=1 4 draws a map of North America on panels 1 through 4. If you specify a frame-oriented map (H, L, LALO, SAT, or VH) in the *map* parameter and a range in the PAN keyword, the maps may be different because each uses the navigation of the panel it's drawing to. For example, MAP H PAN=2 4 draws a map on panel 2 using the navigation for panel 2, draws a map on panel 3 using the navigation for panel 3, and draws a map on panel 4 using the navigation for panel 4.

The coverages, projections, colors and line widths for the predefined maps listed in the *map* parameter (EUR, EURB, EURO, MID, NA, POLI, PRC, PRCB, USA, WIS, WORL, and the 50 two-letter state post office abbreviations) are set in the text files **MAP.CORE**, **MAP.SITE**, and **MAP.USER**. The **.CORE** file is supplied with McIDAS-X and should not be modified. The **.USER** file overrides the **.SITE** file, which overrides the **.CORE** file. This order of precedence allows you or your site to change existing settings and/or add new maps in your **.USER** or **.SITE** file. To create a **.USER** or **.SITE** file, copy the **.CORE** file from **~mcidas/data** to your **$HOME/mcidas/data** directory (or another **MCPATH** directory) and edit the existing predefined maps and/or add new ones.

Labels

The LABEL keyword defines the format and colors of the map's latitude and longitude labels. LABEL=NO omits the labels. The five options for the *scheme* parameter are described below.

- The WESTPOS option, which can also be specified as WEST, W, YES or Y, labels Western hemisphere longitudes positive and Eastern hemisphere longitudes negative.
- The EASTPOS option, which can also be specified as EAST or E, labels Eastern hemisphere longitudes positive and Western hemisphere longitudes negative.
- The ALLPOS option, which can also be specified as ALL or A, labels all latitudes and longitudes positive.
- The 360E option labels longitudes 0 - 360, with values increasing eastward.
- The 360W option labels longitudes 0 - 360, with values increasing westward.

Projections

The pseudo-Mercator projection has equidistant latitude lines; otherwise it is the same as a true Mercator projection. The Lambert conformal projection includes polar stereographic projection as a special case. If you enter only one positive SLAT value or if you accept the default of 60°, the projection is polar stereographic with the North Pole as the origin of projection. If you enter only one negative value for SLAT, the origin of the polar stereographic projection is the South Pole. You must enter two SLAT values to obtain a general Lambert conformal projection. If SLAT1 + SLAT2 > 0, the North Pole is the origin of projection. If SLAT1 + SLAT2 < 0, the South Pole is the origin of projection.

For some values of keywords LAT, LON, SLON and SLAT, the area defined cannot be found. If this happens, an error message is displayed and the program stops. Use SLON for CONF projections only. The default value is computed by assuming that the standard meridian is located in the middle of a box set by two points and is parallel to the left and right edges of the TV screen. If a map is centered, the default SLON is equal to LON.

Examples

The examples below are divided into four groups that include satellite, pseudo-Mercator, polar stereographic and general Lambert conformal projections.

Satellite Projection

MAP VH -7 DASHSET=2 255 8

This entry draws a high-resolution map in graphics color level 7 with dashed lines. The dashes are each two pixels in length with gaps of eight pixels between the dashes.

MAP VH GRA=1-12

This entry draws a high resolution map on graphics frames 1 through 12. The map is identical in all 12 frames because it uses the navigation for the current image frame.

MAP VH IMA=1 GRA=1-12

This entry draws a high resolution map on graphics frames 1 through 12. The map is identical in all 12 frames because it uses the navigation for image frame 1.

MAP X 5 LALO 6 INT=5 10

This entry draws a map on a satellite image in graphics color level 5. It draws latitude/longitude lines in graphics color level 6 and labels them. The interval for latitude lines is 5°. The interval for longitude lines is 10°.

MAP USA X LALO LABEL=NO

This entry draws a high resolution USA map and latitude/longitude lines without labels.

MAP STATE=OH IN

This entry draws the states of Ohio and Indiana, if they're present in the current frame navigation.

MAP COUNTY=DANE 'ST CROIX' COLUMBIA STATE=WI

This entry draws Wisconsin's Dane, St. Croix, and Columbia counties, if they're present in the current frame navigation. Note that St. Croix must be specified without a period and must be enclosed in quotes because of the space in the county name.

IMGDISP MET/FD LATLON=0 0 MAG=-5 BAND=8; MAP; MAP X 2 FILE=OUTLHPOL

These entries display band 8 of the most recent image in dataset MET/FD, then draw the world coastal boundaries and the world political boundaries contained in the newly-displayed image. The coastal boundaries are drawn in the default graphics color level and the political boundaries are drawn in graphics color level 2.

Pseudo-Mercator Projection

MAP AK X LALO

This entry draws a map of Alaska and latitude/longitude lines with labels.

MAP EURO 2; MAP EURB X LALO

This entry draws a map of Europe with coastal and political boundaries.

MAP MN 3 STATE=ALL WIDTH=2
MAP X -3 COUNTY=ALL DASHSET=1 255 4

The first entry defines a map domain of Minnesota and its surrounding states, then draws the states in graphics color level 3 with a line width of two pixels. The second entry draws all counties (in all states) present in the current frame navigation (defined by the first entry) in graphics color level 3 with dotted lines.

MAP MID 3 STATE=IA ZONE=6 19 34

This entry defines a map domain of the Midwest, but only draws Iowa's National Weather Service zones 6, 19, and 34 in graphics color level 3. If DOMAIN=YES is also specified, the states in the domain (the Midwest) are also drawn, using the **OUTLUSAM** map file.

MAP WORL 7; MAP X 8 FILE=OUTLHPOL

These entries draw a map of the world coastal boundaries in graphics color level 7, and the world political boundaries in graphics color level 8.

MAP AR COUNTY=FAULKNER WHITE CRAWFORD MADISON 'VAN BUREN' CARROLL NEWTON BOONE STATE=AR

This entry defines a map domain of Arkansas and its surrounding states, but only draws the eight Arkansas counties listed with the COUNTY keyword. Note that Van Buren is enclosed in single quotes because of the space in the county name.

MAP DEF STATE=AZ COUNTY=MARICOPA GILA LAT=31 37 LON=107 117
MAP VH 2

The first entry defines a map domain of the specified latitude and longitude range, but only draws Arizona's Maricopa and Gila counties. The second entry draws all states in the map domain in graphics color level 2.

Polar Stereographic Projection

MAP WI PRO=CONF

This entry draws a map of Wisconsin in conformal projection.

MAP LALO PRO=CONF LAT=0 70 LON=90 -90

This entry draws latitude/longitude lines and labels for the specified North American region.

MAP DEF X LALO PRO=CONF LAT=30 45 LON=90 110 SLON=100 FILE=OUTLSUPU

This entry draws a map and latitude/longitude lines for a region with the specified latitude/longitude range and standard longitude. The map is drawn using the **OUTLSUPU** very high resolution map file. In this case, the map file can also be chosen by specifying VH as the first parameter (instead of DEF) and omitting the FILE=OUTLSUPU keyword.

MAP X X LALO PRO=CONF LAT=-20 -20 LON=0 180

This entry draws a map of the northern hemisphere with the North Pole as the origin of projection.

MAP X X LALO PRO=CONF LAT=20 20 LON=0 180 SLAT=-60

This entry draws a map of the southern hemisphere with the South Pole as the origin of projection.

MAP X X LALO PRO=CONF LAT=-90 SLAT=-90 LON=0 SCALE=23

This entry draws a map of Antarctica with latitude and longitude lines and a scale of 1:23,000,000. The map can be rotated 180 degrees by changing LON=0 to LON=180.

MAP X X LALO PRO=CONF LAT=-5 -35 LON=-115 -170 SLAT=-60

This entry draws a map of Australia with latitude/longitude lines and labels in the projection centered at the South Pole.

MAP X X LALO PRO=CONF LAT=-60 0 LON=-70 -145 SLAT=-60 SLON=140

This entry produces the same map of Australia but rotates it by 90° so that the South Pole is located at the right of the frame.

Lambert Conformal Projection

MAP PRO=CONF LAT=20 60 LON=70 165 SLAT=40 60

This entry draws a Lambert conformal map of the United States.

MAP X X LALO PRO=CONF LAT=40 70 LON=150 -150 SLAT=40 60

This entry draws a map of the Alaska region. In the Lambert conformal projection, the map splits but the information is complete.

MAP X X LALO PRO=CONF LAT=-15 LON=-135 SLON=180 SCALE=25 SLAT=-60 -40

This entry draws a map of Australia with latitude and longitude lines.

MCEVAL

Runs a McIDAS command substituting McIDAS string values.

Format

MCEVAL *command*

Parameter

command complete McIDAS command to run, including parameters and keywords

---

Remarks

MCEVAL substitutes the correct values for McIDAS strings specified in *command*, then runs the command. Use MCEVAL if you need to update a McIDAS string and use its new value in a concatenated command (separated by a semicolon). See the examples below. Also use MCEVAL if you run McIDAS-X commands with McIDAS string substitutions from the Unix shell. See *Appendix H, Running Commands Outside a McIDAS-X Session*, for more information.

Examples

ASK1 LL {Center latitude and longitude:}; MCEVAL IMGDISP GOES/VIS LATLON=##LL

This entry runs an ASK1 command that prompts you for a value to enter for the string LL. After you type a value for string LL, the IMGDISP command uses it to center the image at that location.

te.k STATION \"KSEA; mceval.k SFCLIST \#STATION 3

This entry, when typed from the Unix shell, runs a TE command that sets the value of the string STATION to KSEA. It then runs the SFCLIST command, using the value of the string in a SFCLIST command. That is, it runs the command SFCLIST KSEA 3.

MCMENU

Starts the F Key Menu System.

Format

MCMENU *file font*

Parameters

file name of the menu file; this is case sensitive and can include the directory with the file name; see the Examples

font name of the font to use in the F Key menu window; see your **.mcidasrc** file for lists of fonts (default=7x14)

Remarks

This command opens an F Key Menu window and loads the specified menu file. You must make the window active to use the menus.

See Also

For more information, see *Chapter 5, F Key Menu System*.

Examples

MCMENU demo.mnu

This entry opens an F Key Menu window and loads the menu file **demo.mnu**.

MCMENU /home/johndoe/mcidas/menus/IMAGE.MNU

This entry opens an F Key Menu window and loads the menu file **/home/johndoe/mcidas/menus/IMAGE.MNU**.

MDE

Edits MD files.

Formats

MDE *file* **KEYS=LIST**
MDE *file* **KEYS=DEL** *conditions*
MDE *file* **KEYS=REST** *conditions*
MDE *file* **KEYS=***schema keys* **UNIT= VALU=** *conditions*

Parameter

file MD file number to edit (default is set by MDU SET or the previously run MDE command)

Keywords

KEYS=	*schema keys*		schema keys in the MD file to edit; you can edit more than one at a time only if they contain actual (non-missing) values
	DEL		deletes the records meeting the search conditions
	LIST		lists the keywords in the MD file schema by row and column headers and data
	REST		restores a deleted record; specify ROW and COL using the *conditions* below
UNIT=	units in which to enter the new key values; specify the units if they are different from those native to the MD file		
VALU=	new value for the keys		
	MISS		enters missing data values
conditions	search conditions to meet when searching for a record to edit; use any or all of the options below		
	ROW=	*min max*	row number or range of numbers (default=1)
	COL=	*min max*	*column* number or range of numbers (default=1)
	schema keys **=**	*low high units*	range and units of the schema keyword to search for; *units* are necessary if the range is specified in units other than those native to the MD file

---

Remarks

Use command MDE to edit or restore MD files. If you delete a record accidentally, you can recover it with KEYS=REST provided you have not repacked the file.

Examples

MDE KEYS=DEL ROW=1 5 COL=15 20

This entry deletes columns 15 to 20 in rows 1 through 5.

MDE 6514 KEYS=LIST

This entry lists the keys for MD file 6514.

MDE KEYS=T TD VALU=MISS ID=KMSN TIME=12

This entry flags the temperature (T) and dew point (TD) as missing for Madison, Wisconsin, at 12 UTC.

Use the four commands below to edit a ship observation in MD file 6512.

MDU SET 6512

This entry sets the MD file pointer to 6512. All subsequent MD file commands are performed on this MD file.

MDE KEYS=T UNIT=C VALU=20 ID=KWPR TIME=12

This entry sets the search conditions ID=KWPR and TIME=12. MDE will locate ship ID KWPR at 12 UTC and change the temperature to 20° C.

MDE KEYS=DEL ROW=5 COL=10 15

This entry deletes the records at row 5 and columns 10 to 15.

MDE KEYS=REST ROW=5 COL=10

This entry restores the record at row 5 and column 10.

MDU

MD file utility.

Formats

MDU COPY *sfile dfile*
MDU DEL *bfile efile*
MDU DIR *file id "comments*
MDU LIST *bfile efile [keyword]*
MDU MAKE *file schema version nrows ncols id "comments*
MDU SET *file*

Parameters

COPY	copies the contents of the MD file
DEL	deletes an MD file
DIR	changes a file identification number and comments
LIST	lists the file directory headers
MAKE	makes a new MD file
SET	sets the current MD file number
sfile	source MD file number
dfile	destination MD file number
bfile	beginning MD file number
efile	ending MD file number (default=*bfile*)
file	MD file number
id	identification number, YYDDD (default=0 if used with the MAKE option; default=current ID if used with the DIR option)
"comments	comments added to the file; the double quote is mandatory
schema	MD file schema name, for example, IRAB, ISFC
version	schema version number; use X for schema defaults
nrows	number of rows; use X for schema defaults
ncols	number of columns; use X for schema defaults

Keyword

DAY=	Julian day to list, YYDDD

Remarks

Before using the MAKE option, be sure the schema type is registered on your workstation. See commands LSCHE and SCHE for more information. Also see *Registering MD File Schemas*.

You can use the ADDE command PTLIST in place of MDU LIST, and the ADDE command PTCOPY in place of MDU COPY.

Examples

MDU COPY 1 10

This entry copies MD file 1 into MD file 10.

MDU DEL 2 6

This entry deletes MD files 2 through 6.

MDU DIR 1 97290 "SHIP AND BUOY DATA

This entry changes the directory of MD file 1 so the ID is 97290 and the comment is *SHIP AND BUOY DATA*.

MDU LIST 1 40

This entry lists the MD file directory headers from MD files 1 to 40.

MDU MAKE 1 ISHP 3 X X 97300 "SHIP DATA

This entry creates MD file 1 of schema ISHP, version 3. It uses row and column defaults. The ID is 97300, labeled *SHIP DATA*.

MDU SET 5

This entry sets the current MD file number to 5. All subsequent MD file commands are performed on MD file 5.

MDX

Plots or grids data from MD files or grids.

Formats

MDX CLE PREFIX =
MDX LIST PREFIX=
MDX PLT *[keywords]*
MDX PLT *ugrid vgrid [keywords]*
MDX *grid [keywords]*
MDX *parm [keywords]*

Parameters

CLE restores strings with the specified prefix to their default values

LIST lists the values of the strings with the specified prefix

PLT plots one or more parameters as defined in the prefixed strings

ugrid grid number containing the u-component of the wind

vgrid grid number containing the v-component of the wind

grid one or more grid numbers of the parameters to plot or contour

parm one or more MD file keys or derived parameters; see the Remarks

---

Keywords

MDX has many keywords for fetching and displaying data. They are divided into five groups for discussion.

- Parameter keywords control the display characteristics for individual parameters, e.g., color, size, units, and scaling.
- Output device keywords control the size of the display and set the output device.
- Grid/contour keywords change the grid size, smoothing and contour interval.
- Map keywords specify maps and projections.
- Miscellaneous keywords perform various functions.

Parameter Keywords

COLOR= graphics color level of the plotted data graphics; use keyword GCOLOR for contours (default=3)

DIVIDE= divides the data values before plotting; to multiply, use a number from 0 to 1 (default=1)

FORMAT= FORTRAN format specification for the output (default=I12; character data default=A4)

LSIZE= label height in pixels; if LSIZE=5, all data is plotted; the numbers can overwrite each other (default=8)

u v if the parameter WNV is specified, a negative LSIZE plots vectors of constant length; if the parameter WIN is specified, a positive LSIZE defines a relative length of the wind barb; specify u and v lengths with the same value

MISS= missing data code (default=Z80808080)

OFFSET= operates on the data before output; data value=(data - OFFSET)/DIVIDE; the UNIT keyword uses the DIVIDE and OFFSET keywords (default=0)

PLACE= place to plot the data relative to the station location; the station is located at 41 in a 9 x 9 matrix starting at 1 in the upper-left corner, increasing along a row; use when plotting more than one parameter (default=41)

UNIT= units to plot the parameter; possible conversions are: K to C, F; MB to INHG; MPS, MPH, KT (any); KM, M, CM, MM, MI, NMI, YD, FT, IN (any) (default=the MD file original units)

SYMB plots weather symbols (default=CHAR)

Output Keywords

GELE= *beg end* graphics element extents (default=depends on the frame size)

GLINe= *beg end* graphics line extents (default=depends on the frame size)

GRA= graphics frame number (default=current)

IMA= image frame number (default=current)

OUT= **PLO** plot (default)

CON grid and contour

PELE= *beg end* CRT/printer element extents (default=1 60 for the CRT; 1 120 for the printer)

PLIN= *beg end* CRT/printer line extents (default=1 19 for the CRT; 1 60 for the printer)

Grid/Contour Keywords

CINT= grid contour interval (default=2)

DER= **WNV** plots wind vectors from the grid file data

DRAW= **YES** draws contours after gridding (default)

NO does not draw contours

GCOLor= graphics color level (default=2)

GRIDF= destination grid file to store grids (default from IGU SET)

GUESS= grid to use as first guess (default=none)

INC= grid increment in degrees (default=1.0)

SMOOTH= smoothing parameter (default=40)

Map Keywords

LAT= *min max* minimum and maximum latitude extents

LON= *min max* minimum and maximum longitude extents

MAP= **NA** North America

USA USA

MID Midwest USA

State Post Office Abbreviation

MCHAR= map character for the CRT/printer output (default= . . .)

MCOLOR= graphics map color level (default=1)

	0	for no map

PRO=	**CONF**	conformal projection for polar stereographic or Lambert conformal
	MERC	pseudo-Mercator projection (default)
	SAT	satellite projection

SCALE= map scale ratio for PRO=CONF (default=not true to scale)

SLAT= *lat1 lat2* standard latitudes; for polar stereographic, specify lat1; for Lambert conformal, specify *lat1 lat2*; used with keyword PRO=CONF (default=60, polar projection)

SLON= specifies the normal longitude for polar stereographic or Lambert conformal; used with the keyword PRO=CONF (default=centers longitude of the map limits)

Miscellaneous Keywords

DERIVE= *temp dewpoint pressure (elevation)*

if THAE, THA or MIX is displayed when the parameter does not appear in the MD file (it will be calculated); use elevation with sea level pressure

for an SVCA or ISFC file, use: DERIVE=T TD PSL ZS

for a RAOB or IRAB file, use: DERIVE=T TD P

GRIDF= grid file number to read (default=from IGU SET)

MDf= MD file number to read; you can combine up to six MD files using the PREFIX keyword (default=from MDU SET)

PREFIX= MDX stores options and defaults in the string table with a prefix at the beginning of the string name; the prefix can be any printable character except a numeric or the pound sign (#) (default=$)

You can set up two prefixed string tables using keyword PREFIX to run these arithmetic operations: - (subtract), * (multiply), + (add), / (divide), : (catenate).

For all operations except catenate, the MD file must have the same schema; if LAT/LON is not contained in the column header, only one row can be operated on, for example: PREFIX=$ - !

SORT= *keyword1 loval1 [hival] keyword2 . .*

selects data from the MD file that meets its conditions; in the SORT keyword list, do not use an equals sign (=) between keywords and *loval, hival; hival* is optional if *loval* is numeric; do not specify *hival* if *loval* is alphabetic; *keyword* is any parameter from an MD file or a ROW and COL number, for example: SORT=TIME 12 13 ST WI

SOU=	**MDF**	data source is an MD file (default)
	GRID	data source is a grid file

Remarks

MDX is a flexible routine that plots or contours from any schema MD file or grid. When plotting or contouring ISFC, SVCA, IRAB or RAOB schema MD file data, it is easier to use commands SFCPLOT, SFCCON, RAOBPLOT and RAOBCON. When contouring grid data, it's easier to use command GRDDISP.

MDX is the general MD file plotter/gridder. It is intended for use in macros, not entered directly.

With each MDX command entry, the information (i.e., parm, time, date, map boundaries, display colors) is stored in strings with a special character prefix. The default prefix is a dollar sign ($), but you can define a prefix using keyword PREFIX with any single character except a pound sign (#) or alphanumeric character.

The *parm* parameter may be any key name in the MD file (use command MDL to list the keys), or one of these derived parameters: ABV, ADV, DVG, MIX, SPD, STR, THA, THAE, VOR, WIN, WNV.

When you use the plus sign (+) as the last positional parameter, the command information is stored in the string table but not plotted. This allows you to run arithmetic operations on the data using the PLT option with the keyword PREFIX.

The most basic command to display data is MDX *parm*, where *parm* is any parameter found in an MD file. The command plots all of this parameter from the current MD file (from MDU SET) on the current graphics frame on a map as determined by the limits of the data. To do anything more with MDX, you must know something about the type of data in the MD file. For example, if you want to plot 500 mb heights for day 97250 at 12 UTC over the Midwest, you must know that the height, pressure, day and time are referenced through parameters Z, P, DAY and TIME in the MD file.

The command to enter (with IRAB schema data in MD file 5000) is:

MDX Z MAP=MID MDF=5000 SORT=P 500 DAY 97250 TIME 12

Keyword SORT provides a way of limiting the data to a specific subset of interest. This is probably the most used keyword along with a map specification keyword (MAP or LAT, LON). When using history data, the contour plot does not indicate a day unless you indicate that day as a SORT condition. Do NOT use an equals sign (=) between the keys and values in the SORT list.

Keywords LAT and LON default to limits that guarantee complete coverage of the data distribution, rounded to the nearest whole degree.

When MDX creates and interpolates data to a grid, the amount of extrapolation into data void regions is controlled via SYSKEY word 897. It has a range of 2 to 20 and a default of 10. Use command SYSVAL to list or change its value. Grid points beyond its value are assigned missing value codes, and are not contoured. Note that data density, grid spacing and smoothing can affect a data value's results.

You can use the ADDE command GRDDISP in place of MDX SOU=GRID, and the ADDE command PTDISP in place of MDX SOU=MDF OUT=PLO.

Examples

Before using this example, enter MDX CLE to clear and initialize the string entries that MDX uses.

1. Plot the weather data from MD file 4000 at 18 UTC.

MDX WX1 MDF=4000 SORT=TIME 1

2. Contour the temperature advection from MD file 3020 over the Midwest at 15 UTC.

MDX T ADV MDF=3020 MAP=MID OUT=CON SORT=TIME 15

3. Plot T in degrees Fahrenheit for MD file 4050, for Wisconsin.

MDX T + MDF=4050 UNIT=F MAP=WI

This command will not plot because + is the last positional parameter. Now plot in graphics color level 2 (MDF, UNIT and MAP are the same).

MDX T COLOR=2

4. Plot the temperature from the current surface data for 15 UTC for a lat/lon box. The MD file is the last digit of the Julian day (if zero, MD=10); today is 97322 so use MD file 2.

MDX T MDF=2 LAT=30 50 LON=90 105 SORT=TIME 15

We forgot to specify hourlies (no specials) and degrees C. So, when MDX is done plotting, enter the command below.

EG; MDX T SORT=TIME 15 TYPE 0 UNIT=C

5. Plot 1000-500 mb thickness over the USA, 12 UTC, MD file 1030, by entering the following command sequence.

MDX CLEAR

MDX Z + SORT=P 500 TIME 12 MAP=USA MDF=1030 PREFIX=$

MDX CLEAR PREFIX=!

MDX Z + SORT=P 1000 TIME 12 MAP=USA MDF=1030 PREFIX=!

MDX PREFIX=$ - !

6. Grid the winds from MD files 2020 and 2045 for streamlines.

MDX STR + MDF=2020 PREFIX=$

MDX STR + MDF=2045 PREFIX=!

MDX PREFIX=$: !

7. Use THAE, THA and MIX. It is easier to use the macros (SFCPLOT, RAOBPLOT, etc.) to display these derived parameters. When using MDX, an extra keyword, DERIVE, is necessary.

MDX MIX MDF=3 MAP=MID DERIVE=T TD PSL ZS SORT=TIME 17 DAY 92003 TYPE 0

This entry plots a mixing ratio from the surface data for day 92003.

8. Plot the grid points on the graphics for grid 10, grid file 6512, which contains temperature data.

MDX 10 SOU=GRID GRIDF=6512 MAP=MID UNIT=C

9. Plot the wind flags at the grid points for *u* grid 1 and *v* grid 2. When plotting wind flags, you cannot change the units.

MDX PLT 1 2 SOU=GRID PRO=SAT SORT=TIME 12

10. Plot parameters T and TD from MD file 4000 at 12 UTC over Wisconsin. Position the temperature above the dew points with the PLACE keyword. First, set it up with no plot.

MDX T TD + MDF=4000 SORT=TIME 12 MAP=WI PLACE=41 50

Now plot the data.

MDX PLT

11. Plot the u- and v-component vectors on the graphics from grids 1 and 2. When plotting wind vectors, you cannot change the units.

MDX PLT 1 2 SOU=GRID DER=WNV MAP=MID

12. Plot the weather symbols for 18 UTC on day 97301.

MDX WX1 UNIT=SYMB MDF=1 MAP=MID SORT=TIME 18 TYPE 0 DAY 97301

MFPLOT

Plots a frequency distribution of MD file data.

Format

MFPLOT *parm file [keywords] "label*

Parameters

parm	MD file data key to plot; see the Remarks
file	MD file number (default=current)
"label	text label for the distribution plot; the double quote is mandatory (default=DISTRIBUTION)

Keywords

conditions	conditions to meet when searching for a record to plot; use any or all of the options below		
	ROW=	*min max*	range of row numbers (default=all rows)
	COL=	*min max*	range of column numbers (default=all columns)
	schema key=	*min max units*	range and units of other schema keys used to limit the search
COLOR=	graphics color level of the plotted data (default=3)		
LENGTH=	length of each class interval, in *parm*'s units; see the Remarks		
NUMBER=	number of class intervals; see the Remarks (default=10)		
XDIVIDE=	*majr minr*		number of major and minor divisions for the x-axis (default=10 5)
XRANGE=	*min max*		range of *parm*'s values for the x-axis; see the Remarks (default=from search *conditions*)
YDIVIDE=	*majr minr*		number of major and minor divisions for the y-axis (default=10 5)
YRANGE=	*min max*		range of occurrences of *parm*'s values for the y-axis (default=computed based on the number of matches in the MD file)

Remarks

MFPLOT plots a frequency distribution of any key in the file. Use the MDL KEYS=LIST command to list the valid keys.

Specify the program keywords COLOR, LENGTH, NUMBER, XDIVIDE, XRANGE, YDIVIDE and YRANGE using the entire keyword name to avoid confusing them with keywords entered as schema search conditions.

If *parm* is used as a search condition keyword, the data is plotted in its specified units, and its minimum and maximum values are the defaults for keyword XRANGE. Otherwise, the units for the plot are the schema's default for *parm*, and the defaults for keyword XRANGE are *parm*'s low and high values among all matching records.

Do not specify both the NUMBER and LENGTH keywords. If one is specified, the other is automatically calculated using these formulas:

NUMBER=(*max-min*)/LENGTH

LENGTH=(*max-min*)/NUMBER

Max and *min* are from the keyword XRANGE. For example, if the x-axis range is from 250 K to 300 K and NUMBER=10, then LENGTH=5 (each class interval includes 5 K). If LENGTH=2, then NUMBER=25.

Keywords XDIVIDE and YDIVIDE define the number of major and minor divisions on the graph. Minor defines the number of subdivisions for each major segment. Labels are placed at each major division.

Examples

MFPLOT T 5 TIME=12 "World Temperatures at 12 UTC

This entry plots a frequency distribution of 12 UTC temperatures in MD file 5. The data is plotted in the default units with the minimum and maximum values along the x-axis set to the low and high values among all matching records. The title *World Temperatures at 12 UTC* appears at the top of the plot.

MFPLOT SPD 2131 LENGTH=1 YDIVIDE=30 2 SPD=10 60 KT

This entry plots a frequency distribution of wind speeds between 10 and 60 knots in MD file 2131. The length of each class interval is 1 knot; the y-axis has 30 major divisions, each with 2 minor divisions.

MFPLOT P 19 XRANGE=100 300 MB LEV=TRO1 TIME=0 NUMBER=20 "Tropopause Heights

This entry plots a frequency distribution of 0 UTC tropopause heights between 100 and 300 mb in MD file 19. The length of each class interval is 10 mb.

MFPLOT TD 7 ROW=1 12 COL=1100 1600 COLOR=6

This entry plots a frequency distribution of dew point temperatures in graphics color level 6 using the data in rows 1 through 12 and columns 1100 through 1600 in MD file 7.

MOSRPT

Lists MOS forecasts from the GFS, NAM and NGM models.

Formats

MOSRPT LIST *state [keywords]*
MOSRPT *id1 . . idn nfcst [keywords]*

Parameters

LIST	lists the valid MOS stations (default); in the output's "Data Types" columns, a G indicates a GFS MOS station, an E indicates an NAM MOS station, and an F indicates an NGM MOS station; the NAM MOS and NGM MOS codes are E and F, respectively, because they were previously referred to as ETA MOS and FOUS14 MOS, and because the database uses N for NEXRAD stations
state	state for the LIST option; specify a single 2-character state post office abbreviation, or **ALL** for all states (default=**ALL**)
id1 . . idn	lists forecasts for these stations and/or states; use 3- or 4-character IDs for stations and 2-character state post office abbreviations for states
nfcst	number of forecasts to list (default=1)

Keywords

DATaset=	ADDE dataset name; specify as *group/descriptor* or *alias*; this keyword is ignored when using the LIST option (default=RTWXTEXT/GFSMOS if TYPE=GFS; default=RTWXTEXT/NAMMOS if TYPE=NAM; default=RTWXTEXT/NGMMOS if TYPE=NGM)
TYPe=	MOS data type; valid entries are GFS, NAM or NGM for the *id1 . . idn* option, and GFS, NAM, NGM or ALL for the LIST option (default=GFS); this keyword is ignored if the DATASET keyword is specified

---

Remarks

Forecasts generated from Model Output Statistics (MOS) are based on multiple linear regression analysis of forecasts from the specified model: GFS (Global Forecast System), NAM (North American Mesoscale) or NGM (Nested Grid Model). The National Weather Service terminated the NGM model and its associated products on 03 March 2009 (NWS TIN08-90); the NGM options in this command are still valid for archive purposes.

Examples

MOSRPT

This entry lists the GFS MOS stations in all states.

MOSRPT KIAD 3

This entry lists the last three GFS MOS forecasts for Washington, D.C.

MOSRPT WY TYPE=NAM

This entry lists the most recent NAM MOS forecasts for all NAM MOS stations in Wyoming.

MOSRPT LIST AZ TYPE=ALL

This entry lists the GFS, NAM and NGM MOS stations in Arizona.

MOSRPT KPHX TYPE=GFS; MOSRPT KPHX TYPE=NAM; MOSRPT PHX TYPE=NGM

These entries list the most recent GFS, NAM and NGM MOS forecasts for Phoenix, Arizona. Note that the NGM MOS station ID (PHX) is different than the GFS and NAM MOS station ID (KPHX). Use MOSRPT LIST or the STNLIST command to find the correct IDs for the stations you're looking for.

MOVIE

Displays a sequence of loops.

Format

MOVIE *l1 d1 l2 d2 . . ln dn [keywords]*

Parameters

l1 . . ln the loop to display; the range for the Image Window is 1 to 99; the range is 1 to 9 when using the WIN keyword

d1 . . dn number of minutes to display the loop (default=.1)

Keyword

WIN= briefing window to display the loop; the range is 0 to 9 (default=Image Window)

Remarks

Define loops using the DEFLOOP command. Create briefing windows using the BRIEF command.

Use the appropriate procedure listed in the table below to cancel the active movie process. When you cancel the active movie process, the current loop continues running until you manually stop the loop. Use **Alt A** to advance a frame and **Alt B** to back up a frame in the loop.

Display	**Cancel Procedure**
Image Window	Run MOVIE with no parameters to cancel the active movie. Press **Alt L** to manually stop or start looping.
Briefing Window	Run MOVIE WIN= to cancel the active movie in the specified briefing window. Make the window active, then press **L** to manually stop or start looping.

Examples

MOVIE

This entry cancels the active movie on the main display.

MOVIE 2 0.5 1 1 3 2 WIN=1

This entry displays loop 2 for one-half minute, loop 1 for one minute, loop 3 for two minutes on briefing window 1, and then repeats the sequence. This movie sequence runs until you run the command MOVIE WIN=1 to cancel it.

NAVCALC

Calculates satellite subpoints and orbit times.

Formats

NAVCALC *navdataset option [keywords]*
NAVCALC *imgdataset option [keywords]*
NAVCALC *tledataset option [keywords]*

Parameters

navdataset name of the ADDE dataset containing the system navigation files for a satellite; specify as *group*/*descriptor* or *alias*

imgdataset name and position of the ADDE dataset containing the image to use as a navigation source; specify as *group*/*descriptor*.*position* or *alias*.*position*; when specifying an image dataset, you must also use the TYPE=I keyword (*position* default=0)

tledataset name of the ADDE dataset containing the NORAD Two-Line Element (TLE) data; specify as *group*/*descriptor* or *alias*; when specifying a TLE dataset, you also must use the TYPE=T keyword

option

FWD	computes the earth coordinates of a surface point based on image coordinates; not valid with TLE datasets
HGT	estimates the height of a feature based on earth coordinates and image coordinates; not valid with TLE datasets
INV	computes the image coordinates of a surface point based on earth coordinates; not valid with TLE datasets
LOS	computes the time ranges when a polar orbiting satellite is visible (in the line of sight) from a surface point
SP	computes a geostationary satellite subpoint for a specific time; or computes polar orbiting satellite subpoints for a range of times
VIEW	computes the times that a polar orbiting satellite and sensor view a surface point; not valid with TLE datasets

---

Keywords

ANTenna=	*angle*	minimum angle, in degrees, above the horizon at which a satellite signal is readable; valid only with the LOS option (default=5)
DAY=	*bday eday*	range of days for calculations; see the Remarks (default=current, or from *imgdataset*)
ELEInc=	*inc num*	increment between elements and the number of element values for multiple calculations using the FWD option
HEight=	height, in kilometers, of the surface point; valid only with the FWD, INV, LOS, and VIEW options (default=0 if the LATLON keyword is used to specify surface point; station height if the STATION keyword is used)	
HTol=	height tolerance, in kilometers, for HGT option (default=0.5)	
IMC=	**ON**	compute subpoint using navigation that has been corrected by the Image Motion Compensator; this keyword is valid only when using the SP option with GVAR (GOES-8 and later) image datasets (default)
	OFF	compute actual subpoint, ignoring the IMC
LATInc=	*inc num*	increment between latitudes and the number of latitude values for multiple calculations using the INV option
LATlon=	*lat lon*	earth coordinates of the surface point; valid only with the HGT, INV, LOS, and VIEW options

LINEle=	*lin ele*	image coordinates of the surface point; valid only with the FWD and HGT options
LINInc=	*inc num*	increment between lines and the number of line values for multiple calculations using the FWD option
LONInc=	*inc num*	increment between longitudes and the number of longitude values for multiple calculations using the INV option
SENsor=	*type*	
	POES options:	AVHRR (default), HIRS, MSU
	DMSP options:	OLS (default), SSM/I, SSM/T, or SSM/T2
	GOES and GVAR options:	IMAGER (default) and SOUNDER
	this keyword is not valid with the SP option	
STAtion=	station ID of the surface point; valid only with the HGT, INV, LOS, and VIEW options; specify as the ID, e.g., KMSN or YSSY, optionally followed by the station type in brackets, e.g., ARX[N] or ARX[V]; see the TYPE keyword in the STNLIST command for valid station types	
TIME=	*btime etime inc*	time range for calculations and interval between subpoint calculations; see the Remarks (default=0 23:59:59, or from *imgdataset*; *inc* default=*etime* minus *btime*)
TYPE=	**S**	uses the system navigation files specified with *navdataset* (default)
	I	uses the navigation of the image specified with *imgdataset*; overrides the SENSOR keyword
	T *'sat'*	uses the navigation of the TLE dataset specified in *tledataset*; *sat* is the name of the satellite in the dataset whose data you want to list; use the READ command to list the dataset and see which satellite names it contains

---

Remarks

Use both the DAY and TIME keywords to specify the time period for navigation calculations. For example, DAY=182 183 TIME=21 3 specifies a range from 21:00 UTC on day 182 through 03:00 UTC on day 183.

Specify *inc* with TIME keyword only when using the SP option with polar orbiting satellites. The SP option does not list day or time ranges for geostationary satellite navigation.

When selecting a portion of an image using the TIME keyword, specified earth or image coordinates must be present in that portion. When selecting a portion of a navigation file, specified or computed image coordinates are redefined based on that portion and not the entire file. For example, if a navigation file starts at 0 UTC but you specify TIME=6, the image coordinates 1,1 refer to the line starting at 6 UTC.

The LOS and VIEW options may not list the same number of entries: a satellite may be visible from the surface point, but not able to view the point.

The FWD option computes the earth coordinates of the northwest corner of a pixel. The INV option computes the image coordinates of a pixel containing the specified earth coordinates. Therefore, a NAVCALC INV command followed by a NAVCALC FWD command may not return the initial earth coordinates.

When working with GVAR system navigation files, navigation may not be from the time specified. NAVCALC requests navigation at the time specified with the TIME keyword. If navigation exists for that time or for earlier on that day, it is used. If not, NAVCALC requests navigation for six hours later, or 23:59:59, whichever is earlier. This request is repeated, advancing by six hours each time, until navigation is found or the end of the day is reached, meaning no navigation exists for that day.

The table below shows the meanings of column labels that appear in NAVCALC output for the FWD, HGT, and INV options.

Column Label	Description
hgt	Height of the given point.
sat	Zenith angle of the satellite from the given point.
solar	Zenith angle of the sun from the given point. If the sun is below the horizon for the given image and point, the *solar* and *relaz* values are "- - - - -".

relaz	Relative azimuth angle between the sun and the satellite, as seen from the given point. If the sun is behind the satellite, the relative azimuth is zero. If the sun is to the right, the relative azimuth is positive. If the given point is within 0.01 degree of the satellite subpoint, the *relaz* value is "undef" (undefined).

Examples

NAVCALC NAV/GOES8 SP DAY=96300

This entry lists the subpoint of the GOES-8 satellite on day 96300.

NAVCALC NAV/MODIS SP TYPE=T 'TERRA' TIME=12 15 :02

This entry lists the subpoints of the TERRA satellite every two minutes between 12 and 15 UTC of the current day.

NAVCALC NAV/NOAA12 VIEW LATLON=49 100.5 DAY=97001 97005

This entry lists the dates and times between 0:00 UTC on day 97001 and 23:59:59 UTC on day 97005 that the location 49° N, 100.5° W is in the view of the NOAA-12 satellite's AVHRR sensor.

NAVCALC A/A.8016 LOS TYPE=I STATION=KIRK TIME=18 12 DAY=93216 93219 ANTENNA=3

This entry lists the dates and times between 18 UTC on day 93216 and 12 UTC on day 93219 that the polar orbiting satellite with image data in position 8016 of dataset A/A is visible from an antenna in Kirksville, MO tracking at an angle of three degrees or greater above the horizon.

NAVCALC WALLOPS/N12HRPT.10 INV TYPE=I LATLON=43:08 89:21 HEIGHT=1

This entry computes the image line and element based on the specified latitude, longitude and height information, and the navigation of the image in position 10 of dataset WALLOPS/N12HRPT.

NAVCALC WALLOPS/N12HRPT.10 FWD TYPE=I LINELE=1223 1758 HEIGHT=1

This entry computes the latitude and longitude based on the specified line, element and height information, and the navigation of the image in position 10 of dataset WALLOPS/N12HRPT.

NAVCALC EAST/CONUSV.-2 HGT LINELE=3916 13436 LATLON=43:07:18 89:20:42 TYPE=I

This entry estimates the height of a feature based on its line, element, latitude and longitude information, and the navigation of the image in relative position -2 of dataset EAST/CONUSV.

NAVDISP

Displays polar orbital tracks and coverage.

Formats

NAVDISP *navdataset frame [keywords]*
NAVDISP *imgdataset frame [keywords]*
NAVDISP *tledataset frame [keywords]*

Parameters

navdataset name of the ADDE dataset containing the system navigation files for a satellite; specify as *group*/*descriptor* or *alias*

imgdataset name and position of the ADDE dataset containing the image to use as a navigation source; specify as *group*/*descriptor*.*position* or *alias*.*position*; when specifying an image dataset, you must also use the TYPE=I keyword (*position* default=0)

tledataset name of the ADDE dataset containing the NORAD Two-Line Element (TLE) data; specify as *group*/*descriptor* or *alias*; when specifying a TLE dataset, you also must use the TYPE=T keyword

frame frame number for display (default=current)

Keywords

AColor= *c1 . . cn* graphics color level of the range circle for each antenna site specified with the STATION and LATLON keywords; use positive numbers for solid lines and negative numbers for dashed lines; see the Remarks (default=-4)

ALabel= '*name1*' . .'*namen*'

label for each antenna site specified with the STATION and LATLON keywords; see the Remarks (defaults=A, B, etc. for locations specified with LATLON; station names or IDs for locations specified with STATION)

CColor= graphics color level of the lines displaying the geographical coverage of the satellite sensor or the boundaries of the image specified in *imgdataset*; use positive numbers for solid lines and negative numbers for dashed lines; see the Remarks (default=0, do not plot coverage)

DAY= *bday eday* range of days for display; see the Remarks (default=current, or from *imgdataset*)

HEIGHT= *hgt1 . . hgtn* height, in kilometers, of each antenna site specified with the LATLON keyword; see the Remarks (default=0)

HORizon= *ang1 . . angn* minimum angle above the horizon at which a satellite signal is readable for each antenna site specified with the STATION and LATLON keyword; see the Remarks (default=5)

INT= interval, in seconds, between the time marks on the orbital track (default=computed)

LABel= '*text*' *loc col size*

display label, location, color and size

'*text*' label or description of the display (default=from *dataset*)

loc location for the label at the bottom of the frame; specify a number 1 through 5; see the Remarks (default=1)

col graphics color level for the label (default=from the CCOLOR keyword if used; otherwise from the TRCOLOR keyword)

size height of the label characters in pixels (default=7)

LATlon= *lat1 lon1 . . latn lonn*

antenna locations; see the Remarks

NAV= *num* frame to use for navigating the output (default=display *frame*)

C use the current frame's navigation

SENsor= *type*

POES options: AVHRR (default), HIRS, MSU

DMSP options: OLS (default), SSM/I, SSM/T, or SSM/T2

STAtion= *sta1 . . stan* antenna locations; specify as station IDs, e.g., KMSN or YSSY, optionally followed by the station type in brackets, e.g., ARX[N] or ARX[V]; see the Remarks

TIME= *btime etime* time range to plot; see the Remarks (default=0 23:59:59, or from *imgdataset*)

TMColor= graphics color level of the time marks on the orbital track (default=3)

TMLabel= *color size*

size and graphics color level of the time mark labels

color graphics color level for the labels (default=from the TMCOLOR keyword if used; otherwise, default=3)

size height of the label characters in pixels (default=7)

TRColor= graphics color level of the orbital track; use positive numbers for solid lines and negative numbers for dashed lines (default=2)

TYPE= **S** uses the system navigation files specified with *navdataset* (default)

I uses the navigation of the image specified with *imgdataset*; overrides the SENSOR keyword

T '*sat*' uses the navigation of the TLE dataset specified in *tledataset*; *sat* is the name of the satellite in the dataset whose tracks you want to display; use the READ command to list the dataset and see which satellite names it contains

---

Remarks

NAVDISP displays orbital tracks, coverage edges, and antenna range circles on a navigated frame. Because there are no map drawing keywords, you must display an image or map on the frame before using NAVDISP.

Use the STATION and LATLON keywords to specify antenna locations. Colors, labels, and effective horizons specified with the ACOLOR, ALABEL, and HORIZON keywords are applied to antenna locations in the order entered, beginning with those specified with the STATION keyword.

The HEIGHT keyword is not valid with the STATION keyword. Use it with the LATLON keyword to specify the height of an antenna location. The STATION keyword uses the height defined in the **STNDB.*** station database files.

Use both the DAY and TIME keywords to specify the time period for plotting orbital tracks and coverage. For example, DAY=182 183 TIME=21 3 specifies a range from 21:00 UTC on day 182 through 03:00 UTC on day 183.

If you use the CCOLOR keyword and TYPE=I without the DAY or TIME keyword, the displayed geographical coverage is limited to the image specified in *imgdataset*. Otherwise, the entire geographical coverage of the sensor is displayed.

Labels are displayed at the bottom left corner of the frame. A default label is constructed using information from the system navigation file or the image used for navigation. The label includes the dataset name, the satellite, sensor, and day or range of days displayed. Use the LABEL keyword and the '*text*' parameter if you want to choose a different label.

Use the *loc* parameter with the LABEL keyword to place additional labels on different lines at the bottom of the frame. Location 1 is the top line; additional labels are below. The maximum number of lines depends on the frame size: five for a 480 by 640 frame.

See Also

Use the PACORB command to display orbital tracks and coverage for the Pacific Test Range. Use the USAORB command to display orbital tracks and coverage for North America.

Examples

MAP WORL

NAVDISP NAV/NOAA14 DAY=96300

These entries draw the NOAA-14 satellite orbital tracks for day 96300 over a map of the world.

MAP MID

NAVDISP A/A.8016 TYPE=I

These entries draw the orbital track for the polar orbiting satellite with image data in position 8016 of dataset A/A. The track is displayed over a map of the Midwest.

MAP DEF LAT=-40 -40 LON=-90 90 SLAT=-40 PRO=CONF

NAVDISP NAV/NOAA12 TMCOLOR=5 DAY=96300 TIME=12 18 LATLON=-77:51 -166:40 HORIZON=10 ALABEL='McMurdo' INT=300

These entries draw the NOAA-12 satellite orbital tracks between 12:00 UTC and 18:00 UTC on day 96300 over a map of Antarctica. A range circle is drawn for an antenna located at 77:51° S, 166:40° E. The antenna tracks at an angle of 10 degrees or greater above the horizon. The label *McMurdo* is plotted in the center of the range circle.

MAP NA

NAVDISP WALLOPS/N14GAC TYPE=I STATION=KTTS KE28 KEDW

These entries draw the orbital track for the polar orbiting satellite with data in the most recent image in the WALLOPS/N14GAC dataset. Antenna circles are drawn for the NASA Shuttle Landing Facility, Kennedy Space Center, FL; Northrup Landing Strip, White Sands Space Harbor, NM; and Edwards Air Force Base, CA.

MAP WORL

NAVDISP NAV/MODIS TYPE=T 'TERRA'

These entries draw the TERRA satellite orbital tracks for today over a map of the world.

NAVLIST

Lists system navigation entries.

Format

NAVLIST *dataset* *[keywords]*

Parameter

dataset	name of the ADDE dataset containing the system navigation files for a satellite; specify as *group*/*descriptor* or *alias*

Keywords

DAY=	*bday* *eday*	lists system navigation entries for a range of days (*bday* default=current; *eday* default=*bday*)
	ALL	lists all system navigation entries for the satellite or specified sensor source numbers
SENsor=	*type*	lists system navigation entries for the specified sensor; ignored if the SS keyword is used (default=first sensor listed for the satellite in the **CORE.SAT** file)
SS=	*bss* *ess*	lists system navigation entries for a range of sensor source numbers (*bss* default=number of satellite; *ess* default=*bss*)
	ALL	lists system navigation entries for all sensor sources with the day range specified in the DAY keyword

---

Remarks

Run the command SEE CORE.SAT to list the valid sensor source numbers and sensor types.

Examples

NAVLIST NAV/NOAA12

This entry lists today's navigation entries for the NOAA-12 satellite in the dataset NAV/NOAA12.

NAVLIST NAV/NOAA12 DAY=97070 97080 SS=62 64

This entry lists the navigation entries for sensor source numbers 62, 63 and 64 from days 97070 through 97080 in the dataset NAV/NOAA12.

NAVLIST NAV/GOES9 DAY=97080 97090 SENSOR=SOUNDER

This entry lists the navigation entries for the GOES-9 sounder from days 97080 through 97090 in the dataset NAV/GOES9.

O

Switches the image frame position to the opposite frame.

Remarks

To switch to the opposite image frame,

Press: **Alt O**

or

Type: **O**

Press: **Enter**

Since graphics frames have no opposites, their frame numbers do not change.

See Also

Use command LS to assign opposite frame numbers.

OBSRPT

Lists the observational text reports.

Format

OBSRPT *dataset [keywords]*

Parameter

dataset ADDE dataset name; specify as *group*/*descriptor* or *alias*; the dataset must reside on a workstation running McIDAS-XCD

Keywords

CO=	*co1 . . con*	lists observations for all stations in these countries; use the CCODE command to list valid country codes
FORm=	*tab bcol ecol*	changes the output format (default=raw observation text if FORM is not specified, defaults below if FORM is specified)
	tab	number of spaces to indent the second and subsequent lines of an observation (default=0)
	bcol	beginning column number of the text frame to list observations (default=1)
	ecol	ending column number of the text frame to list observations (default=80)
ID=	*id1 . . idn*	lists observations from stations with these IDs; see the Remarks
NEWest=	*day hour*	most recent day and hour to list; you must specify both *day* and *hour*
NHOurs=	number of hours of data to search; you must also specify the NEWEST keyword; you cannot specify both NHOURS and NPERIOD	
NPEriod=	number of time periods to search; you cannot use NPERIOD with NHOURS or NUM; see the Remarks (default=1)	
NUM=	number of observations per station to list; you cannot specify both NUM and NPERIOD	
OLDest=	*day hour*	oldest day and hour to list; you must specify both *day* and *hour*
REG=	*reg1 . . regn*	list observations from these states; use two-letter P.O. abbreviations
TYPe=	*type1 . . typen*	subtypes of observations to list; must be an integer; valid entries vary with report types; see the Remarks (default=all types)

Remarks

OBSRPT lists station-based observational text data. Other commands are easier to use with specific data types. For example, use SFCRPT to list surface hourly reports; use SYNRPT to list synoptic reports; use RAOBRPT to list upper air reports; and use TAFRPT to list terminal aerodrome forecasts.

You must specify at least one of the following keywords: ID, REG, or CO. If you specify the ID keyword, you must use the station ID format consistent with the type of observational report being listed. When listing RAOB or synoptic data, you must use the 5-digit station ID format, for example, 72645. When listing surface hourly and TAF data, you must use the 4-character station ID format, for example, KGRB.

Use the NPERIOD keyword to specify how many reporting periods, as opposed to the number of hours, to search for each station. For surface hourly reports and terminal forecasts, the reporting interval is one hour. For synoptic and RAOB data, the interval may be three, six, or twelve hours.

Use the TYPE keyword to specify subtypes of data in a single observational report. McIDAS-XCD decoders assign unique integer values to the

various types of data. The table below lists the valid integer entries for the TYPE keyword.

TYPE=	Surface Hourly Reports	Upper Air Reports	Terminal Forecasts	Synoptic Reports
0	Hourly	TTAA - mandatory temperature and wind data	--	All reports
1	Special	TTBB - significant level temperature data between the surface and 100 mb	Long term forecast	--
2	--	TTDD - significant level temperature data above 100 mb	Short term forecast	--
3	--	PPBB - significant level wind data between the surface and 100 mb	--	--
4	--	PPAA - significant level wind data above 100 mb	--	--

Examples

OBSRPT RTWXTEXT/SYNOPTIC ID=72450

This entry lists the latest synoptic report for Wichita, Kansas.

OBSRPT RTWXTEXT/SFCHOURLY CO=AU REG=WI RI ID=KLAX

This entry lists all surface hourly reports for Los Angeles and all stations in Australia, Wisconsin, and Rhode Island from the past hour.

OBSRPT RTWXTEXT/UPPERAIR ID=72201 REG=TX TYPE=0 1

This entry lists the most recent TTAA and TTBB RAOB reports for Key West, Florida, and all stations in Texas.

OBSRPT RTWXTEXT/SYNOPTIC ID=03772 72641 FORM=6 1 60 NEWEST=#Y 12 NPERIOD=2

This entry lists the last two periods of synoptic reports ending at 12:00 UTC of the current day for London, England, and Madison, Wisconsin. The first line of each report starts at column 1 and all subsequent lines start at column 6; all lines end at or before column 60. Because the interval between synoptic reports is three hours, this command lists reports from 9:00 UTC and 12:00 UTC.

OBSRPT RTWXTEXT/TERMFCST REG=DE OLDEST=1996285 10 NUM=7

This entry lists up to seven terminal aerodrome forecasts for all stations in Delaware with the oldest report no older than 10:00 UTC on day 1996285.

OBSRPT RTWXTEXT/SFCHOURLY ID=KMTW KABQ KSEA NEWEST=96300 18 NUM=5

This entry lists five reports each for Manitowoc, Wisconsin; Albuquerque, New Mexico; and Seattle, Washington, for a total of 15 reports. The most recent report is from 18:00 UTC on day 1996300.

OBSRPT RTWXTEXT/UPPERAIR REG=NV NEWEST=1996288 20 NHOURS=10 TYPE=0 NUM=5

This entry lists up to five TTAA RAOB reports for all stations in Nevada from the ten hours prior to 20:00 UTC on day 1996288.

OS

Runs an operating system command.

Format

OS *"command*

Parameter

"command operating system command to run

Remarks

The OS command lets you run a Unix operating system command from the McIDAS Text and Command Window. The command should include the correct path, if needed. Do not run a command that expects user input while running, for example, **telnet**.

See Also

You can use the ! (Exclamation Point) command to run a Unix command.

Examples

OS "ls -l /data/AREA*

This entry runs the Unix **ls** command to list all file names beginning with AREA in the **/data** directory.

OS "echo $MCPATH

This command runs the Unix **echo** command to list the current setting of the **MCPATH** environment variable.

PACORB

Displays polar orbital tracks and coverage for the Pacific Test Range.

Formats

PACORB *navdataset frame [keywords]*
PACORB *imgdataset frame [keywords]*

Parameters

navdataset	name of the ADDE dataset containing the system navigation files for a satellite; specify as *group*/*descriptor* or *alias*
imgdataset	name and position of the ADDE dataset containing the image to use as a navigation source; specify as *group*/*descriptor*.*position* or *alias*.*position*; when specifying an image dataset, you must also use the TYPE=I keyword (*position* default=0)
frame	frame number for display (default=current)

Keywords

CColor= graphics color level of the lines displaying the geographical coverage of the satellite sensor or the boundaries of the image specified in *imgdataset*; use positive numbers for solid lines and negative numbers for dashed lines; see the Remarks (default=0, do not plot coverage)

DAY= *bday eday* range of days for display; see the Remarks (default=current, or from *imgdataset*)

HORizon= *kwj vbg*

minimum angle, in degrees, above the horizon at which a satellite signal is readable

- *kwj* angle at Kwajalein atoll (default=5)
- *vbg* angle at Vandenberg AFB (default=5)

INT= interval, in seconds, between the time marks on the orbital track (default=computed)

LABel= '*text*' *loc col* display label, location, and color

- '*text*' label or description of the display (default=from *navdataset* or *imgdataset*)
- *loc* location for the label at the bottom of the frame; specify a number 1 through 5; see the Remarks (default=1)
- *col* graphics color level for the label (default=from the CCOLOR keyword if used; otherwise from the TRCOLOR keyword)

NAV= **C** use existing frame navigation; use this keyword to overlay additional orbital tracks on the same frame (default=erase frame and redraw map)

SENsor= *type*

POES options:	AVHRR (default), HIRS, MSU
DMSP options:	OLS (default), SSM/I, SSM/T, or SSM/T2

TIME= *btime etime* time range to plot; see the Remarks (default=0 23:59:59, or from *imgdataset*)

TMColor=	graphics color level of the time marks on the orbital track (default=3)	
TRColor=	graphics color level of the orbital track; use positive numbers for solid lines and negative numbers for dashed lines (default=2)	
TYPE=	**S**	uses the system navigation files specified with *navdataset* (default)
	I	uses the navigation of the image specified with *imgdataset*; overrides the SENSOR keyword

---

Remarks

Use both the DAY and TIME keywords to specify the time period for plotting orbital tracks and coverage. For example, DAY=182 183 TIME=21 3 specifies a range from 21:00 UTC on day 182 through 03:00 UTC on day 183.

If you use the CCOLOR keyword and TYPE=I without the DAY or TIME keyword, the displayed geographical coverage is limited to the image specified in *imgdataset*. Otherwise, the entire geographical coverage of the sensor is displayed.

Labels are displayed at the bottom left corner of the frame. A default label is constructed using information from the system navigation file or the image used for navigation. The label includes the dataset name, the satellite, sensor, and day or range of days displayed. Use the LABEL keyword and the '*text*' parameter if you want to choose a different label.

Specify NAV=C to overlay additional orbital tracks on the same frame. Use the *loc* parameter with the LABEL keyword to place additional labels on different lines at the bottom of the frame. Location 1 is the top line; additional labels are below. The maximum number of lines depends on the frame size: five for a 480 by 640 frame.

See Also

Use the USAORB command to display orbital tracks and coverage for North America. Use the NAVDISP command to display orbital tracks and coverage for a user-defined map.

Examples

PACORB NAV/NOAA14 DAY=96300

PACORB NAV/NOAA12 DAY=96300 NAV=C TRCOLOR=7 LABEL=X 2

The first entry draws the NOAA-14 satellite orbital tracks for day 96300 over a map of the Pacific Test Range. The second entry draws the NOAA-12 satellite orbital tracks on the same frame without erasing the NOAA-14 tracks. The NOAA-12 tracks are drawn in graphics color level 7. The label, is positioned on the text line below the first entry's label.

PACORB POESNAV/NOAA14 TIME=13:00 15:00 SENSOR=MSU CCOLOR=-7

This entry draws the NOAA-14 satellite orbital tracks from the current day between 13 and 15 UTC and displays the geographical coverage of the MSU sensor in dashed color level 7. The track data is drawn over a map of the Pacific Test Range.

PACORB VANDENBERG/N12GAC TYPE=I

This entry draws the orbital track for the polar orbiting satellite with data in most recent image in the dataset VANDENBERG/N12GAC over a map of the Pacific Test Range.

PACORB VANDENBERG/N12HRPT TYPE=I CCOLOR=10

This entry draws the orbital track and the AVHRR geographic coverage for the satellite with data in the most recent image in the dataset VANDENBERG/N12HRPT over a map of the Pacific Test Range. The geographic coverage is drawn in color level 10.

PACORB VANDENBERG/N12HRPT TYPE=I DAY=97097

This entry draws the orbital tracks for the satellite with data in the most recent image in the VANDENBERG/N12HRPT dataset. The tracks for the entire day 97097 are plotted.

PACORB KWAJ/N14GAC TYPE=I NAV=C LABEL='ORBITAL TRACK FOR NOAA-14 #Y' INT=300

This entry draws the orbital track for the satellite with data in the most recent image in the KWAJ/N14GAC dataset using the current frame's navigation. A custom label is displayed, and the time marks are drawn every five minutes along the orbital track.

PANEL

Configures frames with multiple panels.

Format

PANEL *nrow ncol bframe eframe*

Parameters

nrow	number of rows of panels (default=1)
ncol	number of columns of panels (default=*nrow*)
bframe	beginning frame to configure (default=current frame)
eframe	ending frame to configure (default=*bframe*)

Remarks

When you run PANEL, all images and graphics in the frame are erased and the frame is configured with the panel setup specified with the *nrow* and *ncol* parameters. The frame's panel configuration remains until another PANEL command is run or a virtual frame with a different panel configuration is restored with command RVF.

PANEL 1 1 configures a frame so that a single panel occupies the entire frame. Specify numbers greater than 1 for *nrow* or *ncol* to create multiple panels. You cannot make panels of different sizes on the same frame; all panels on the frame are the same size. If the frame has multiple panels, they are numbered from left to right and from top to bottom. For example, PANEL 2 2 configures a frame with panel 1 in the upper-left quadrant, panel 2 in the upper-right quadrant, panel 3 in the lower-left quadrant, and panel 4 in the lower-right quadrant.

By default, output from image and graphic commands is written to the panel where the cursor is located. Use the global keyword PANEL with commands to override the default and display the output in a different panel.

On multipanel frames, commands that use the cursor interactively may produce unexpected results if you move the cursor to a different panel while the command is running. Leave the cursor in the same panel until the command is finished.

The maximum number of panels for a frame depends on the frame's size. The panel size must be at least 32 lines by 32 elements.

Examples

PANEL

This entry configures the current frame with a single panel that consists of the entire frame. All images or graphics in the frame are erased.

PANEL 2 2

This entry configures the current frame with two rows of panels, and two panels in each row. Panels 1 and 2 are in the top row; panels 3 and 4 are in the bottom row. All images and graphics in the frame are erased.

PANEL 2 3 1 10

This entry configures frames 1 through 10 with two rows of panels, and three panels in each row. Panels 1, 2, and 3 are in the top row; panels 4, 5, and 6 are in the bottom row. All images and graphics in the frames are erased.

PC

Positions the cursor at a desired point on the frame.

Formats

PC C
PC L *station*
PC E *lat lon*
PC I *line ele*
PC T *raster pictel*

Parameters

C	positions the cursor at the center of the frame
L	positions the cursor at the station ID or synoptic ID location
E	positions the cursor at the latitude and longitude earth coordinates
I	positions the cursor at the line and element image coordinates
T	positions the cursor at the raster and pictel TV coordinates
station	station identifier, e.g., KMSN or YSSY, optionally followed by the station type in brackets, e.g., ARX[N] or ARX[V]; see the TYPE keyword in STNLIST command for valid station types
lat	latitude of the location
lon	longitude of the location
line	image or graphics line of the location
ele	image or graphics element of the location
raster	TV raster (line) of the location
pictel	TV pictel (element) of the location

Examples

PC E 36 - 98

This entry positions the cursor at 36° North and 98° East.

PC I 3300 1200

This entry positions the cursor at image line 3300 and element 1200.

PC L KMSN

This entry positions the cursor over station KMSN (Madison, Wisconsin).

PC L 72250

This entry positions the cursor over station 72250 (Brownsville, Texas).

PC T 100 356

This entry positions the cursor at TV raster 100, pictel 356.

PCMW

Computes a velocity vector.

Format

PCMW *[keywords]*

Keywords

Color=	graphics color level of the plotted wind flags (default=3)	
GLOck=	**YES**	locks the current graphics frame
	NO	does not lock the frame (default)
Size=	relative size of the plotted wind flags (default=12)	

Remarks

PCMW calculates velocity vectors by making point-to-point measurements as opposed to correlation tracking. Use the LB or LS command to properly set the loop bounds before running the PCMW command. The maximum number of frames in the loop is 10. Use keyword GLOCK to lock the current graphics frame for PCMW output.

The cursor expands and the color changes as a visual queue indicating that half of a vector measurement was made.

Units of u, v and SPD are in meters per second on the text frame and knots on the graphics frame.

Examples

PCMW

This entry computes velocity vectors using the cursor center for locations.

PCMW COLOR=2

This entry starts the local mode of PCMW and plots wind flags in graphics color level 2.

PCMW COLOR=1 GLOCK=YES SIZE=15

This entry plots 15-pixel wind flags in graphics color level 1. All wind flags are plotted on the current graphics frame.

PLAX

Calculates parallax at cursor position.

Format

PLAX *height direction [keywords]*

Parameters

height cloud height in kilometers

direction direction for computed parallax; specify one of the following:

CG cloud-to-ground parallax (default)

GC ground-to-cloud parallax

Keywords

LATlon= *lat lon* moves the cursor to the specified latitude and longitude and calculates parallax (default=current cursor position)

STAtion= moves the cursor to the specified station and calculates parallax; specify as the station ID, e.g., KMSN or YSSY, optionally followed by the station type in brackets, e.g., ARX[N] or ARX[V]; see the TYPE keyword in the STNLIST command for valid station types (default=current cursor position)

Remarks

If you don't specify *height* or *direction*, PLAX calculates the change in parallax at the cursor location as height increases. The information is listed as the change in degrees latitude and longitude per kilometer of height. The *Position error* line indicates the ground position error in kilometers at the cursor location, based on the parallax correction for a feature 10 kilometers above the earth.

If you specify *height*, and CG for *direction*, you define a point in the clouds at the specified height. PLAX calculates and plots the ground location under the given cloud point. The ground location is shown as a 7x7-pixel cross in graphics color level 3. The text output lists the latitude and longitude for the cursor point on the cloud and for the calculated ground point.

If you specify *height*, and GC for *direction*, you define a point on the ground. PLAX calculates and plots the cloud location at the specified height over the given ground point. The cloud location is shown as a 7x7-pixel cross in graphics color level 3. The text output lists the latitude and longitude for the cursor point on the ground and for the calculated cloud point.

PLAX does not work with most polar-orbiting satellite data because the navigation does not include the information needed by PLAX to do its computations.

Examples

PLAX

This entry lists the change in parallax at the cursor location as height increases. The information is given as the change in degrees latitude and longitude per kilometer of height. The *Position error* line indicates the ground position error in kilometers at the cursor location, based on the parallax correction for a feature 10 kilometers above the earth.

PLAX 10 LATLON=45 120

This entry moves the cursor to 45°N, 120°W and calculates the cloud-to-ground parallax at 10 kilometers above the earth. The command also plots the ground point that corresponds with the specified cloud point.

PRDUTIL

Calibration utility for 1-byte PRD image files

Formats

PRDUTIL ADD *dataset num type prodlo prodhi britlo brithi unit scale miss*
PRDUTIL DEL *dataset num type*
PRDUTIL LIST *dataset*

Parameters:

ADD	adds or changes a calibration block segment
DEL	deletes a calibration block segment by number and type
LIST	lists calibration block segments
dataset	ADDE image dataset name; the dataset must be local (i.e., it cannot be on a remote server) and it must be specified as *group/descriptor.position* or *alias.position*
num	calibration block segment number (1-12) to add, change or delete
type	1- to 4-character name of the product defined by the segment, e.g., P, RH, TEMP (default=PROD)
prodlo	minimum output product value (default=0)
prodhi	maximum output product value (default=255)
britlo	input brightness value assigned to *prodlo* (default=0)
brithi	input brightness value assigned to *prodhi* (default=255)
unit	1- to 4- character name of the units for the product value defined by the segment, e.g., F, MB, %
scale	divisor to apply to the product values when they are accessed by another application, e.g., D or IMGPROBE; thus output_value=product_value/scale (default=0, meaning no scaling)
miss	missing value code to assign to the product when a brightness value is outside the range defined with *britlo* and *brithi*; use decimal value -2139062144 for HEX80 (default=0)

Remarks:

Use PRDUTIL to list, add, and delete calibration block segments for a 1-byte PRD image (e.g., an image created with GRDIMG, IMGFILT or IMGOPER commands). Up to 12 segments can be defined using PRDUTIL. Multiple segments can be used to define a single product (*type*). For example, segments 2 through 6 can define the product values, corresponding brightness values, product units, scale factors and missing value codes for a product named RH while segment 7 can define the settings for a product named MIX.

Examples:

PRDUTIL LIST LOCAL/AREA.4205

This entry lists the calibration block segments for the image in position 4205 in the dataset LOCAL/AREA.

PRDUTIL DEL LOCAL/AREA.2 1 RH

This entry deletes calibration block segment number 1 from the image in position 2 in the dataset LOCAL/AREA. The segment type must be RH; otherwise, the segment is not deleted.

PRDUTIL ADD A/A.4205 6 RH 0 100 0 255 %

This entry adds or changes (if previously defined) calibration segment number 6 for the image in position 4205 in the dataset A/A. The product value is named RH with units %, and has a range of 0 through 100 that maps to image brightness values 0 through 255.

PRDUTIL ADD LOCAL/RAIN.20 1 PCP 0 200 194 255 IN 100

This entry adds or changes (if previously defined) calibration segment number 1 for the image in position 20 in the dataset LOCAL/RAIN. The product value is named PCP with units IN, and has a range of 0 through 200 that maps to image brightness values 194 through 255. The scale value of 100 means that the product values will be divided by 100 when accessed by another application, such as IMGPROBE.

PTCON

Converts point data to grids; displays contours and plots.

Format

PTCON *dataset1 . . datasetn [keywords]*

Parameters

dataset1 . . n ADDE dataset names and positions; specify one of the following formats for each dataset:

group/descriptor.position

alias.position

a *position* greater than zero represents an absolute position in the dataset; a *position* less than or equal to zero represents a relative position in the dataset based on date and time, for example, 0 is the most recent and -1 is the next most recent; the *position* ALL represents all positions in the dataset (no default for *group/descriptor* or *alias*; *position* default=ALL if DAY is specified with the SELECT keyword, otherwise default=0)

Keywords

OUT=

CON contours the gridded data (default)

PLOT plots grid points

SAVE *ddataset* saves the created grid to a destination dataset

SCON *ddataset* contours the gridded data and saves the created grid to a destination dataset

SPLOT *ddataset* plots grid points and saves the created grid to a destination dataset; see the Remarks

PARam= parameter to contour; specify a native parameter, a computed parameter or one of the four predefined parameters as listed below:

native_parameter [*unit*]

native parameter to contour and its output units; native parameters are those that are stored in the point file (no default for *parameter; unit* default=from *dataset*)

'*computed_parameter* [*unit*]=*expression*'

computed parameter to contour, its output units, and the mathematical expression used to compute it; computed parameters do not exist in the point file but are derived with the specified mathematical expression containing math operators (+ - * etc.) and the native parameters specified in the Pn keywords, e.g., PAR='TDD[F]=P1-P2'; see the Remarks and the Pn keyword description for more information (no default for *parameter* or *expression*; *unit* default=from *dataset*)

MOVIE draws streamlines with colored line segments that can be animated using the CM command

STREAML draws streamlines using u- and v-component grids

WINDB plots wind barbs using u- and v-component grids

WINDV plots wind vectors using u- and v-component grids

SELect= '*clause1*; *clause2*; . . ; *clausen*'
finds only the data matching the values in the select clauses; single quotes are mandatory; separate select clauses with semicolons;

specify *clause* as a range or a list using one of the following formats:

range:	*parameter* **[***unit***]** *bval eval*
list:	*parameter* **[***unit***]** *val1,val2, . . ,valn*

parameter	point file parameter
[*unit***]**	units for the parameter values; units are optional but must be in square brackets if specified; see the Remarks (default=from *dataset*)
bval eval	range of parameter values; see the Remarks
val1, . . ,valn	list of parameter values

Mathematical Operations Keywords

Pn= *parameter* [*unit*] '*clause1*; *clause2*; ..; *clausen*'
native parameters to use in the math expression specified in the PARAM keyword; use a number between 1 and 99 in the keyword name (P1,P2,..,P99); each keyword specifies a single native parameter and units to substitute in matching Pn values in the math expression, where:

parameter [*unit*]
is the native parameter name and units; *unit* defaults to those for the parameter in the dataset; there is no default for *parameter*

'*clause1*; *clause2*; ..; *clausen*'
are the clauses used to select data for this keyword; only data matching their criteria is used; they are specified exactly as in the SELECT keyword; the Pn keywords inherit the clauses specified in the SELECT keyword, therefore, you should specify a Pn keyword clause only if a value differs from that in the SELECT keyword or you need to add additional clauses; for example, to set the values of P1 and P2 to the 850 and 1000 mb temperatures in degrees Celsius at 0 UTC, respectively, specify the following keywords:
SEL='TIME 0; P 850' P1=T[C] P2=T[C] 'P 1000'

GINFO= *level lunit day time src fhour*
values to assign to the header of the computed parameter grid created with the PARAM and Pn keywords, where:

level	grid level
lunit	grid level units; 2 character maximum
day	grid day (default=value specified for DAY in the SELECT keyword; otherwise, default=today)
time	grid time (default=value specified for TIME in the SELECT keyword; otherwise, default=22:22:22)
src	grid source (default=MDX)
fhour	grid forecast hour (default=0)

Navigation Keywords

LAT=	*lat1 lat2*	latitudes of the points at the lower-right and upper-left corners of the map; you cannot use LAT with the NAV or MAP keyword
LON=	*lon1 lon2*	longitudes of the points at the lower-right and upper-left corners of the map; you cannot use LON with the NAV or MAP keyword

MAP= map for the contour or plot; see the MAP command for a list of valid predefined maps, for example, MID, USA, WORL; you cannot use MAP with the NAV, LAT or LON keyword

MCOlor= graphics color level of the map (default=1)

NAV=	*num*	frame to use for navigation

C use the current frame's navigation

the plot is drawn using the navigation of the specified frame; no map is drawn; you cannot use NAV with the MAP, LAT or LON keyword

PRO= **CONF** polar stereographic or Lambert conformal projection

MERC Mercator projection (default)

SLAt= *lat1 lat2* standard latitudes for a CONF projection; two values for a general Lambert conformal projection; one value for a polar stereographic projection (default=60 60 for northern hemisphere maps defined with the LAT and LON, or MAP keywords; -60 -60 for southern hemisphere maps defined with the LAT and LON, or MAP keywords)

SLOn= standard longitude for a CONF projection (default=computed)

Contour and Plot Keywords

COLor= graphics color level (default=2 for contour, 3 for plot)

ELE= *bele eele* beginning and ending frame elements for output; specify MAX for maximum element number (default=1 MAX when using the NAV keyword; default=20 MAX-20 when using the MAP keyword)

FORMAT= *designator* '*text*'

output format for the grid point values or contour labels, and associated text; valid formats are I, F, or E; see the Remarks (*designator* default=from *dataset*; no *text* default)

GRA= graphics frame for the display (default=current)

LABel= '*text*' *col line ele size*

text, color, location and size for the contour or plot title

'*text*' 72 characters maximum; specify X without quotes for the default title (default=computed)

col graphics color level of the title (default=1 if PARAM=MOVIE, otherwise default=value from the COLOR keyword)

line beginning TV line to write the title (default=below contour or plot)

ele beginning TV element to write the title (default=below contour or plot)

size height of the title text in pixels (default=8)

LINE= *blin elin* beginning and ending frame lines for output; specify MAX for maximum line number (default=1 MAX-57 when using the NAV keyword; default=20 MAX-57 when using the MAP keyword)

LSIze= height, in pixels, of the plot characters or contour labels; the range is 0-50 where 0 omits the labels (default=8)

POWer= power of 10 to multiply the grid point values by for display (default=0)

TDAy= **CAL** display calendar day on the default title; not valid if the ELE, LINE or PAN keyword is used

JUL display the Julian day on the default title (default)

Contour-Specific Keywords

CINt= *num* contour interval; specify 0 to let PTCON compute a reasonable interval; specify your own interval as an integer or decimal value greater than zero; see the Remarks for the default value

string contour values; specify a string name that contains up to 64 values; e.g., CINT=SNOWDEPTH will draw contours at the values specified in the string named SNOWDEPTH

DASh= **ALL** dash all contours

NEG dash negative contours only

POS dash positive contours only (default=solid contours)

LINT= contour label interval (default=1, meaning label every contour)

MOVie= *bcol ecol bspd espd blen elen*

change the defaults for the PARAM=MOVIE output

bcol beginning graphics color level for line segments (default=2)

ecol ending graphics color level for line segments (default=maximum color level)

bspd beginning wind speed represented by *blen*, units are from the created grid (default=computed)

espd ending wind speed represented by *elen*, units are from the created grid (default=computed)

blen beginning length of line segments (default=2)

elen ending length of line segments (default=40)

SMOoth= contour smoothing factor; the range is 0 through 50; not valid with PARAM=MOVIE or PARAM=STREAML; larger numbers produce smoother contours (default=20)

Plot-Specific Keyword

PINt= *row col* row and column intervals of grid points to plot with OUT=PLOT, PARAM=WINDB or PARAM=WINDV; for example, PINT=4 3 plots every fourth grid point in the row direction and every third grid point in the column direction (*row* default=1; *col* default=*row*)

Grid Conversion Keywords

BPAr= smoothing factor for the conversion of point data to grids; a larger number results in a data point influencing a greater number of grid points, and the generation of a smoother grid; BPAR is related to the Barnes parameter used in the grid analysis; the suggested range is 5 to 500 (default=40)

DISt= distance in grid points to extrapolate into areas without data; the range is 1 to 25; grid points beyond the specified distance are assigned missing value codes and therefore are not contoured; the results will vary for a given value depending on data density, grid spacing and smoothing (default=10)

LLInc= *latinc loninc* latitudinal and longitudinal incremements, in degrees, between the grid points in the grid that PTCON creates; use this keyword to control the smoothing and extents of the output contours (*latinc* default=1 if the grid is 10° or more in the latitudinal and longitudinal directions, *latinc* default=computed value <1 if the grid is less than 10° in either direction, *loninc* default=*latinc*)

Grid Output Keywords

DEL= **YES** delete the destination grid file before copying

NO do not delete the destination grid file (default)

DGRid= beginning grid number in the destination grid file to place the copied grids; this keyword may overwrite existing grids; see the Remarks (default=append after the last grid in the grid file)

GLEV= *level unit* grid level to use in the grid header and on the display (default=value entered for LEV, THA, or P in the SELECT keyword *clause*; otherwise, no default)

MAXgrd= maximum number of grids the newly created destination grid file can store; see the Remarks (default=1000)

TITLE= *'text'* title of the newly created destination grid file; 32 characters maximum

---

Text Output Keyword

BYTecount= **YES** lists the number of bytes received from the server

NO does not list the number of bytes (default)

Remarks

PTCON gathers data from one or more point files in one or more datasets. The SELECT keyword defines which records to match and the PARAM keyword defines which parameters in the matching records to use. The selected point data is used to produce grid data.

The grid data may be saved, displayed, or both. The grid is saved in a Mercator projection with the native units or in the units specified with the PARAM keyword.

You must specify a parameter using the PARAM keyword, and a type of navigation using the NAV, MAP, or LAT and LON keywords. If the point file contains multiple times and/or days, you should also specify the TIME and/or DAY parameter in a SELECT keyword clause. Otherwise, all times and/or days will be used.

To file the correct day information in the output grid header, you must specify the DAY parameter in a SELECT keyword *clause*. If you don't specify a day, the value stored in the grid header is today's Julian date.

If you specify MOVIE, STREAML, WINDB, or WINDV with the PARAM keyword, grids of both the u- and v-component of the wind are saved or contoured.

The default contour interval for a grid is taken from the text file **CONTOUR.DEF**. If no file entry matches the grid's parameter, units and level, the default is calculated to produce a reasonable number of contours, usually between 10 and 20. To change, add or delete entries in **CONTOUR.DEF**, first back up the current version, then modify it using the format described near the top of the file.

If you specify DEL=YES or a position number of a nonexistent grid file in the *ddataset* parameter, the grids are copied into a newly created grid file with the attributes specified in keywords MAXGRD and TITLE.

Using the DGRID keyword may result in grids being overwritten in the destination grid file. For example, if the destination grid file has 50 grids (in grids 1 to 50) and you specify NUM=10 DGRID=25, the new grids will overwrite grids 25 to 34 in the destination grid file.

Use the FORMAT keyword options below to format output of the grid point values or contour labels based on FORTRAN designators for data output. To add text to the labels, follow the designator with a text in single quotes.

Designator	Description
I	integer
F	floating point
E	exponential

To specify a parameter's output format, use a designator followed by a number. For example, I4 specifies an integer of up to four digits; F10.1 specifies a floating point decimal of up to 10 digits (including the decimal point) with one digit to the right of the decimal; E10.3 specifies an exponential number of up to 10 characters (including the decimal point and the four-character exponential flag) with three digits to the right of the decimal. Specify an X to use the parameter's default format. If the specified integer or floating point format is too small for the parameter's values, a series of asterisks (***) is plotted.

When using the SELECT and PARAM keywords, you can use the following units:

Unit type	UNIT=	Definition
Distance	CM	centimeters
	DM	decameters
	FT	feet
	GPM	geopotential meters
	IN	inches

	KM	kilometers
	M	meters
	MI	miles
	MM	millimeters
	NMI	nautical miles
	YD	yards
Pressure	HPA	hectopascals
	INHG	inches of mercury
	MB	millibars
Speed	KT or KTS	knots
	MPH	miles per hour
	MPS	meters per second
Temperature	C	Celsius
	F	Fahrenheit
	K	Kelvin

The PARAM keyword contains the parameter to contour. It can be a native parameter, a computed parameter, or a predefined parameter (STREAML, WINDB, WINDV, or MOVIE).

Native parameters are those stored in the point file. They are specified in the PARAM keyword in the format *parameter* [*unit*], where *parameter* is the parameter name in the file, and [*unit*] is the output units, which are optional but must be in square brackets if specified.

Computed parameters do not exist in the point file but are derived with a mathematical expression. Computed parameters are specified in the PARAM keyword in the format '*parameter* [*unit*]=*expression*', where *parameter* is the user-defined parameter name; [*unit*] is the label units, which are optional but must be in square brackets if specified; and *expression* is the mathematical expression consisting of math operators and one or more native parameters defined in the Pn keywords.

The table below lists the math operators valid in the computed parameter expression in the PARAM keyword.

Math Operator	**Function**
+	addition
–	subtraction
*	multiplication
/	division
**	power
SQRT	square root
EXP	exponential (the number e raised to a power)
LOG	natural logarithm
LOG10	base 10 logarithm
SIN	sine
COS	cosine
TAN	tangent
ASIN	arcsine
ACOS	arccosine
ATAN	arctangent
ABS	absolute value
MIN	minimum of two grids
MAX	maximum of two grids

To compute a power of ten, such as 10^{2} or 10^{-3} in the math expression of a computed parameter in the PARAM keyword, specify it in either the 1e*power* or 10**(*power*) format. For example, both 1e-5 and 10**(-5) represent 10^{-5}. You must include the parenthesis with 10**(-5) for it to be interpreted correctly.

The following character limits exist for the PARAM keyword:

- 256 characters maximum for the entire PARAM keyword list
- four characters maximum for the computed parameter name
- four characters maximum for the computed parameter units

When using point/MD files with CYD (ccyyddd) date units (e.g., FO14, IRAB, ISEN, ISFC, ISHP, PIRP and SYN schema MD files created with McIDAS-XCD 7.5 and later), commands PTCON, PTCOPY, PTDISP and PTLIST default to the current year if you don't specify a four-digit year and the nearest century if you specify only the last two digits of the year. For example, if the file uses CYD units for the DAY parameter and its value is 2004260, then you can specify the DAY in the SELECT keyword as 2004260 or 04260. If the current year is 2004, you can also specify it as 260. Use the PTLIST command with FORM=PARAM to list the file's parameters and units.

The PTCON, PTCOPY, PTDISP and PTLIST commands continue to work with point/MD files using the old SYD (*ssyyddd*) date units (e.g., FO14, IRAB, ISEN, ISFC, ISHP, PIRP and SYN schema MD files created with versions of McIDAS-XCD prior to 7.5), with the following exception: If you want to specify DAY in the SELECT keyword and the *ss* portion of the MD file's DAY value is nonzero, then you must specify the units with DAY in the SELECT keyword. For example, if the MD file uses SYD units for DAY and its value is 7197240, then it must appear as SELECT='DAY[SYD] 7197240'. If the MD file uses SYD units for DAY but has a zero (blank) *ss* value, you don't have to specify the units in the SELECT keyword. For example, if the MD file uses SYD units for DAY and its value is 97240 then it can appear as either SELECT='DAY 97240' or SELECT='DAY[SYD] 97240'.

Examples

PTCON SURFACE/HOURLY PARAM=T[F] SELECT='TIME 6' MAP=USA

This entry draws a contour of today's 6:00 UTC surface temperature data (in degrees Fahrenheit) over the United States. The data is taken from the most recent point file in dataset SURFACE/HOURLY. The grid points used to draw the contour are not saved.

PTCON RTPTSRC/SFCHOURLY RTPTSRC/SYNOPTIC RTPTSRC/SHIP PARAM=TD[C] SELECT='TIME 6; DAY 96311' LAT=0 70 LON=-20 120 PRO=CONF GRA=3 LABEL='DEW POINT TEMPS' FORMAT=F6.1 'degrees C' DASH=NEG

This entry draws a contour of the 6:00 UTC dew point temperature data (in degrees Celsius) for day 96311. All data in datasets RTPTSRC/SFCHOURLY, RTPTSRC/SYNOPTIC, and RTPTSRC/SHIP is searched. The contours are drawn on graphics frame 3 using a map with a latitude range of 0° to 70° North and a longitude range of 20° East to 120° West. The label on the image window is *DEW POINT TEMPS*. The contour values are labeled with a floating point format with *degrees C* after each value. Contours with negative values are dashed. The grid points used to draw the contour are not saved.

PTCON SURFACE/SYNOPTIC.7 PARAM=PCP[IN] POW=2 BPAR=20 OUT=SCON TEST/GRIDS.2 MAP=MID SELECT='TIME 18; DAY 96317' DGRID=21 CINT=10 LINT=2

This entry draws contours for the 18:00 UTC six-hour cumulative precipitation (in inches) for day 96317. The data is taken from the point file in position 7 of the dataset SURFACE/SYNOPTIC. The contour is drawn on a map of the Midwest. The values are multiplied by one hundred times their original value. A contour interval of 0.10" is used with labels on every other contour. The grid point conversion is filtered using a smoothing factor of 20. The gridded analysis is saved as grid number 21 of the grid file in position 2 of the dataset TEST/GRIDS.

PTCON SURFACE/HOURLY.5 SURFACE/SYNOPTIC.5 SURFACE/SHIP.5 PARAM=T[C] SELECT='TIME 00; DAY 96315' LAT=-84 -68 LON=20 -171 SLAT=-75 SLON=160 PRO=CONF OUT=SPLOT MYGRIDS.51 DGRID=13 COLOR=3 LLINC=1 4 DIST=6 PINT=2 5

This entry plots surface temperature (in degrees Celsius) from 0:00 UTC for day 96315 using data from the point file in position 5 of datasets SURFACE/HOURLY, SURFACE/SYNOPTIC, and SURFACE/SHIP. Grid point values are plotted on a map of West Antarctica in graphics color level 3 with data from every second row and fifth column displayed. The grid point increment is one degree of latitude and four degrees of longitude. Data is extrapolated up to a distance of six grid points into data void regions. The gridded analysis is saved in grid number 13 of the grid file in position 51 of the dataset with the alias name MYGRIDS.

PTCON SURFACE/HOURLY SELECT='TIME 0; DAY #Y' PARAM=MOVIE MAP=USA MOVIE=2 8

TE MOVIECOLORS "GU MAKE !1 0 !2 0

REPEAT MOVIECOLORS 2 TO 8 BY 1 0 BY 40

CM 2 8 GO=YES

The PTCON entry draws today's 00 UTC surface streamlines with colored line segments on a map of the United States. The TE and REPEAT entries set graphics color levels 2 to 8 to increase linearly from black (level 2) to bright green (level 8). The CM entry then circulates the graphics color levels to animate the wind flow.

PTCON RTPTSRC/SFCHOURLY SEL='DAY #Y;TIME 12' PAR='DELT[F]=P1-P2' P1=T[F] P2=T[F] 'TIME 0' MAP=USA

This entry draws contours of today's 00 UTC to 12 UTC temperature change in degrees Fahrenheit on a map of the United States. The data is taken from the most recent point file in dataset RTPTSRC/SFCHOURLY.

PTCON RTPTSRC/SFCHOURLY SEL='DAY #Y;TIME 12' MAP=NA PARAM='THA[K]=P1/((P2/1000).286)' P1=T[K] P2=PSL[MB]**

This entry contours potential temperature in degrees Kelvin at 12 UTC today on a map of North America. The potential temperature values are calculated using the equation in the PARAM keyword, known as Poisson's Equation.

PTCOPY

Copies point data from one dataset to another.

Format

PTCOPY *sdataset ddataset [keywords]*

Parameters

sdataset source ADDE dataset name and position; specify one of the following formats:

group/descriptor.position
alias.position

a *position* greater than zero represents an absolute position in the dataset; a *position* less than or equal to zero represents a relative position in the dataset based on date and time, for example, 0 is the most recent and -1 is the next most recent; the *position* ALL represents all positions in the dataset (no default for *group/descriptor* or *alias*; *position* default=ALL if DAY is specified with the SELECT keyword, otherwise default=0)

ddataset destination ADDE dataset name and absolute position; specify one of the following formats:

group/descriptor.position
alias.position

use only positive integers for *position*

Keywords

BYTecount= **YES** lists the number of bytes received from the server

NO does not list the number of bytes (default)

DEL= **YES** deletes the destination file before copying; see the Remarks

NO does not delete the destination file (default)

DROw= beginning row number in the destination file to write data; this keyword may overwrite existing rows; see the Remarks (default=append after the last row with data)

NROws= number of rows to allocate when creating an MD file; the valid range is 1 to 1000 (default=schema default)

SELect= '*clause1*; *clause2*; . . ; *clausen*'
finds only the data matching the values in the select clauses; single quotes are mandatory; separate select clauses with semicolons; specify *clause* as a range or a list using one of the following formats:

range: *parameter* **[***unit***]** *bval eval*
list: *parameter* **[***unit***]** *val1,val2, . . ,valn*

parameter point file parameter

[*unit***]** units for the parameter values; units are optional but must be in square brackets if specified; see the Remarks (default=from *sdataset*)

bval eval range of parameter values; see the Remarks

val1, . ., *valn* list of parameter values

TITLE=	'*text*'	title of the destination point file; 32 characters maximum (default=current title if the file already exists; default=title of the source file if the file is new)

Remarks

PTCOPY copies records from the source point data file to the destination point data file. Use the SELECT keyword to define a subset of records to copy. If you don't specify SELECT, all records in the source file are copied.

If you use the range format for the SELECT keyword and *bval* is less than *eval*, all values between and including *bval* and *eval* are selected. For example, if you specify SELECT='LON -177 177', the longitude values -177, -176, . . , 176, 177 are selected. If *bval* is greater than *eval*, all values greater than *bval* or less than *eval*, including *bval* and *eval*, are selected. For example, if you specify SELECT='LON 177 -177', the longitude values 177, 178, 179, 180, -179, -178 and -177 are selected. This feature also allows you to select only wind directions from northwest to northeast with SELECT='DIR 315 45'.

If you specify the DROW keyword, the values of the row header parameters (for example, DAY, TIME and TYPE in ISFC schema MD files) are checked. If the values of the copied data's row header parameters are the same as the values in the destination row, the data is added to the row. Otherwise, the destination row's data is overwritten with the new data. If DROW is not specified, the data is copied to the rows following the last row containing data in the destination file.

If you specify DEL=YES or a position number of a nonexistent point data file in the *ddataset* parameter, the records are copied into a newly created MD file; change the title with the TITLE keyword.

When using the SELECT keyword, you can specify the following units:

Unit type	UNIT=	Definition
Distance	CM	centimeters
	DM	decameters
	FT	feet
	GPM	geopotential meters
	IN	inches
	KM	kilometers
	M	meters
	MI	miles
	MM	millimeters
	NMI	nautical miles
	YD	yards
Pressure	HPA	hectopascals
	INHG	inches of mercury
	MB	millibars
Speed	KT or KTS	knots
	MPH	miles per hour
	MPS	meters per second
Temperature	C	Celsius
	F	Fahrenheit
	K	Kelvin

When using point/MD files with CYD (ccyyddd) date units (e.g., FO14, IRAB, ISEN, ISFC, ISHP, PIRP and SYN schema MD files created with McIDAS-XCD 7.5 and later), commands PTCON, PTCOPY, PTDISP and PTLIST default to the current year if you don't specify a four-digit year and the nearest century if you specify only the last two digits of the year. For example, if the file uses CYD units for the DAY parameter and its value is 2004260, then you can specify the DAY in the SELECT keyword as 2004260 or 04260. If the current year is 2004, you can also specify it as 260. Use the PTLIST command with FORM=PARAM to list the file's parameters and units.

The PTCON, PTCOPY, PTDISP and PTLIST commands continue to work with point/MD files using the old SYD (ssyyddd) date units (e.g., FO14, IRAB, ISEN, ISFC, ISHP, PIRP and SYN schema MD files created with versions of McIDAS-XCD prior to 7.5), with the following exception: If you want to specify DAY in the SELECT keyword and the *ss* portion of the MD file's DAY value is nonzero, then you must specify the units with DAY in the SELECT keyword. For example, if the MD file uses SYD units for DAY and its value is 7197240, then it must appear as SELECT='DAY[SYD] 7197240'. If the MD file uses SYD units for DAY but has a zero (blank) *ss* value, you don't have to specify the units in the SELECT keyword. For example, if the MD file uses SYD units for DAY and its value is 97240 then it can appear as either SELECT='DAY 97240'

or SELECT='DAY[SYD] 97240'.

Examples

PTCOPY RAOB/MANDATORY LOCAL/PT.1 DEL=YES

This entry copies the entire contents of the most recent point file in dataset RAOB/MANDATORY to position 1 of dataset LOCAL/PT. The DEL=YES keyword deletes the existing destination point file and creates a new one with the same title as the source file. Because point files may be very large, copying an entire point file may take a long time. Use the SELECT keyword to copy only the data you need, as shown in the following examples.

PTCOPY SURFACE/ISFC.5 LOCAL/PT.2 SELECT='CO US,CN ; TIME 0 2 ; TYPE 0' DEL=YES TITLE='15 May 96 US and Canada Sfc Data'

This entry copies the 00 through 02 UTC surface hourly (type 0) data for all stations in the United States and Canada in position 5 of dataset SURFACE/ISFC to position 2 of dataset LOCAL/PT. The DEL=YES and TITLE keywords delete the existing destination point file and create a new one with the title *15 May 96 US and Canada Sfc Data*. Because each hour's data in ISFC schema point files is assigned to a separate row, the 00 UTC data is copied to row 1, the 01 UTC data is copied to row 2, and the 02 UTC data is copied to row 3 in the new destination file.

PTCOPY SURFACE/ISFC.5 LOCAL/PT.2 SELECT='CO US,CN ; TIME 6,12 ; TYPE 0' DROW=2

If this entry follows the previous example, it replaces the destination point file's 01 UTC data in row 2 with 06 UTC data, and replaces the 02 UTC data in row 3 with 12 UTC data.

PTCOPY SURFACE/ISFC.5 LOCAL/PT.2 SELECT='CO US,CN ; TIME 18 ; TYPE 0' TITLE='15 May 96 0,6,12,18 UTC Sfc Data'

If this entry follows the previous two examples, it copies the 18 UTC data to row 4 of the destination point file (position 2 of dataset LOCAL/PT) and changes its title to *15 May 96 0,6,12,18 UTC Sfc Data*. The data is placed in row 4 because without DROW being specified, the data is copied to the row after the last row containing data. The destination point file now has the 00, 06, 12 and 18 UTC data in rows 1, 2, 3 and 4.

PTDISP

Displays point data.

Format

PTDISP *dataset frame [keywords]*

Parameters

dataset ADDE dataset name and position; specify one of the following formats:

group/descriptor.position
alias.position

a *position* greater than zero represents an absolute position in the dataset; a *position* less than or equal to zero represents a relative position in the dataset based on date and time, for example, 0 is the most recent and -1 is the next most recent; the *position* ALL represents all positions in the dataset (no default for *group/descriptor* or *alias*; *position* default=ALL if DAY is specified with the SELECT keyword, otherwise default=0)

frame frame number to display the output; specify OPP to display the output on the frame opposite the current frame (default=current)

Keywords

PARam= *p1 . . pn*

list of parameters to plot; the list can contain native and/or computed parameters in the following formats:

native_parameter[*unit*]
native parameter and its output units; native parameters are those that are stored in the point file (no default for *parameter; unit* default=from *dataset*)

'*computed_parameter*[*unit*]=*expression*'
computed parameter, its output units, and the mathematical expression used to compute it; computed parameters do not exist in the point file but are derived with the specified mathematical expression containing math operators (+ - * etc.) and the native parameters specified in the P*n* keywords, e.g., PAR=ID T[F] TD[F] 'TDD[F]=P1-P2' PSL[MB]; see the Remarks and the P*n* keyword description for more information (no default for *parameter* or *expression*; *unit* default=from *dataset*)

SELect= '*clause1*; *clause2*; . . ; *clausen*'
finds only the data matching the values in the select clauses; single quotes are mandatory; separate select clauses with semicolons; specify *clause* as a range or a list using one of the following formats:

range: *parameter***[***unit***]** *bval eval*

list: *parameter***[***unit***]** *val1*,*val2*, . . ,*valn*

parameter point file parameter

[*unit***]** units for the parameter values; units are optional but must be in square brackets if specified; see the Remarks (default=from *dataset*)

bval eval range of parameter values; see the Remarks

val1,. .,*valn* list of parameter values

Mathematical Operations Keyword

P*n*= *parameter*[*unit*] '*clause1*; *clause2*; ..; *clausen*'
native parameters to use in the math expressions specified in the PARAM keyword; use a number between 1 and 99 in the keyword name (P1,P2,..,P99); each keyword specifies a single native parameter and units to substitute in matching P*n* values in the math expressions, where:

parameter[*unit*]
is the native parameter name and units; *unit* defaults to those for the parameter in the dataset; there is no default for *parameter*

'*clause1*; *clause2*; ..; *clausen*'
are the clauses used to select data for this keyword; only data matching their criteria is used; they are specified exactly as in the SELECT keyword; the P*n* keywords inherit the clauses specified in the SELECT keyword, therefore, you should specify a P*n* keyword clause only if a value differs from that in the SELECT keyword or you need to add additional clauses; for example, to set the values of P1 and P2 to the 850 and 1000 mb temperatures in degrees Celsius at 0 UTC, respectively, specify the following keywords: SEL='TIME 0; P 850' P1=T[C] P2=T[C] 'P 1000'

Navigation Keywords

LAT=	*lat1 lat2*	latitudes of the points at the lower-right and upper-left corners of the map; you cannot use LAT with the NAV or MAP keyword
LON=	*lon1 lon2*	longitudes of the points at the lower-right and upper-left corners of the map; you cannot use LON with the NAV or MAP keyword
MAP=		map for the plot; see the MAP command for a list of valid predefined maps, for example, MID, USA, WORL; you cannot use MAP with the NAV, LAT or LON keyword
MCOlor=		graphics color level of the map (default=1)
NAV=	*num*	frame to use for navigation
	C	use the current frame's navigation; the plot is drawn using the navigation of the specified frame; no map is drawn; you cannot use NAV with the MAP, LAT or LON keyword
PRO=	**CONF**	polar stereographic or Lambert conformal projection
	MERC	Mercator projection (default)
SLAt=	*lat1 lat2*	standard latitudes for a CONF projection; two values for a general Lambert conformal projection; one value for a polar stereographic projection (default=60 60 for northern hemisphere maps defined with the LAT and LON, or MAP keywords; -60 -60 for southern hemisphere maps defined with the LAT and LON, or MAP keywords)
SLOn=		standard longitude for a CONF projection (default=computed)

Plot Keywords

COLor=	*c1 . . cn*	graphics color level for each parameter specified with the PARAM keyword (default=3); each of the *c1 . . cn* values can be specified as a single number, e.g., COLOR=3 5 6, or in format *color*[*min-max*B*inc*] to assign multiple colors for the parameter, based on its values; see the Remarks for details
DEClut=	**YES**	declutters the plot; does not plot overlapping values (default)
	NO	does not declutter the plot
ELE=	*bele eele*	beginning and ending frame elements for output; specify MAX for maximum element number (default=1 MAX when using the NAV keyword; default=20 MAX-20 when using the MAP keyword)
FORMAT=	*f1 . . fn*	output format for each parameter specified with the PARAM keyword; valid formats are A, I, F, E, K, VECT, FLAG, SYMB, WXCM, WXCA, WPCM, and WPCA; see the Remarks (default=from *dataset*)
LINe=	*blin elin*	beginning and ending frame lines for output; specify MAX for maximum line number (default=1 MAX-57 when using the NAV keyword; default=20 MAX-57 when using the MAP keyword)

LOC= *symbol color size*
mark the location of each observation with a centered symbol in the specified graphic color level and size; valid symbols are **+**, **.**, ***** and **o** (default=+ 3 6 if LOC keyword is specified; otherwise no symbol)

LSIze= *l1 . . ln* plotted height, in pixels, of each parameter specified with the PARAM keyword (default=8)

MULtiply= *m1 . . mn* multipliers for each parameter specified with the PARAM keyword; each parameter is multiplied by the specified number before plotting; use a number between 0 and 1 to divide (default=1)

OFFset= *l1 e1 . . ln en* frame line and element offset from the center plot position; useful when plotting multiple parameters; specify the offset for each parameter using minus (-) for up and left and plus (+) for down and right; the center of the text or symbol is plotted at the specified offset position (default=0 0)

TDAy= **CAL** displays the calendar day on the default title; not valid if the ELE, LINE or PAN keyword is used

JUL displays the Julian day on the default title (default)

TITLE= '*text*' *col line ele size*
text, color, location and size for the title

'*text*' 72 characters maximum; specify X without quotes for the default title (default=computed)

col graphics color level of the title (default=value from the COLOR keyword)

line beginning TV line to write the title (default=lower-left corner of frame; exact line number depends on frame size and the value specified with the COLOR keyword)

ele beginning TV element to write the title (default=20)

size height of the title text in pixels (default=8)

WINd= *type par1 par2*
wind type (speed and direction, or u and v components) and the corresponding parameter names in the point file; use this keyword only with PARAM=WIND; specify *type*, *par1* and *par2* as follows:

type **SPDDIR** if the two parameters define speed and direction, or **UV** if the two params define u and v

par1, par2 corresponding parameter names in the point file

(default=SPDDIR SPD DIR; if you specify WIND=UV, then the defaults for *par1* and *par2* are U and V)

Output Keyword

BYTecount= **YES** lists the number of bytes received from the server

NO does not list the number of bytes (default)

Remarks

PTDISP displays data from records in a point data file. The SELECT keyword defines which records to match and the PARAM keyword defines which parameters in the matching records to display. If PTDISP has started to plot data, you can stop it by pressing **Alt Q**.

If you use the range format for the SELECT keyword and *bval* is less than *eval*, all values between and including *bval* and *eval* are selected. For example, if you specify SELECT='LON -177 177', the longitude values -177, -176, . . , 176, 177 are selected. If *bval* is greater than *eval*, all values greater than *bval* or less than *eval*, including *bval* and *eval*, are selected. For example, if you specify SELECT='LON 177 -177', the longitude values 177, 178, 179, 180, -179, -178 and -177 are selected. This feature also allows you to select only wind directions from northwest to northeast with SELECT='DIR 315 45'.

You can assign output plot color based on the parameter's values by using the COLOR keyword format *color*[*min-max*B*inc*], where *color* is the beginning graphics color level, *min* and *max* indicate the range of parameter values to color-code, and *inc* is the increment (the dash means "to" and the B means "by"). For example, when plotting integer values, COLOR=2[30-59B10] plots values 30-39 in level 2, values 40-49 in level 3, and values 50-59 in level 4. Values less than 30 are in level 1, and values greater than 59 are in level 5. When plotting floating point values (e.g.,

FORMAT=F5.2), COLOR=2[30-59.99B10] plots values 30.00-39.99 in level 2, values 40.00-49.99 in level 3, and values 50.00-59.99 in level 4. Values less than 30.00 are in level 1, and values greater than 59.99 are in level 5. Also, when plotting color-coded wind flags (PAR=WIND with FORMAT=FLAG), the units of the *min* and *max* values are knots, even if you specify PAR=WIND[MPS].

Use the FORMAT keyword options below to format output based on FORTRAN designators for data output:

Designator	**Description**
A	character
I	integer
F	floating point
E	exponential

To specify a parameter's output format, use a designator followed by a number. For example, I4 specifies an integer of up to four digits; A4 specifies up to four characters, which is the maximum; F10.1 specifies a floating point decimal of up to 10 digits (including the decimal point) with one digit to the right of the decimal; E10.3 specifies an exponential number of up to 10 characters (including the decimal point and the four-character exponential flag) with three digits to the right of the decimal. Specify an X to use the parameter's default format. If the specified integer or floating point format is too small for the parameter's values, a series of asterisks (***) is plotted.

Use the K format to "clip" leading integer digits and drop the decimal portion of an integer or floating point value. Follow the K with a number to specify the number of integer digits to display. For example, if a pressure value is 1016.26 and you specify FORMAT=K3, *016* is plotted. If you also specify MULTIPLY=10, *163* is plotted.

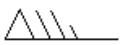
Specify FORMAT=SYMB to translate weather report types (R-, S+, for example) into equivalent weather symbols. Only the first type in the report is plotted.

Specify FORMAT=WXCM, WXCA, WPCM or WPCA to translate WMO numeric codes into equivalent weather symbols. The table below shows the ISFC and SYN MD file schema parameter combinations that correspond to each format.

Format	**Schema**	**Parameters (and values if required)**
WXCM	ISFC SYN	WXC1, WXC2, WXC3 and WXC4 WXC1 with STYP=1, 2 or 3
WXCA	SYN	WXC1 with STYP=4, 5, 6 or 7
WPCM	SYN	WPC1 and WPC2 with STYP=1, 2 or 3
WPCA	SYN	WPC1 and WPC2 with STYP=4, 5, 6 or 7

If you specify FORMAT=VECT or FLAG, you must also specify PARAM=WIND, which requests wind speed (SPD) and direction (DIR) from the point file. The VECT format produces a wind vector display. The FLAG format produces a wind flag display. The vector or flag is placed at the center plot position; using the OFFSET keyword has no effect.

Wind flag symbols use triangular flags, and long and short lines, or barbs, to represent wind speed. For example, the wind flag shown here represents a wind speed of 75 knots or 37.5 meters/second.



- a flag represents 50 knots or 25 meters/second
- a long barb represents 10 knots or 5 meters/second
- a short barb represents 5 knots or 2.5 meters/second.

A default title is constructed using the information specified with the PARAM and SELECT keywords. For example, if your command contains the keywords PARAM=T[F] TD[C] SELECT='DAY 96325; TIME 12; TYPE 0', the default title is: *T[F] TD[C] DAY=96325 TIME=12 TYPE=0*. Use the TITLE keyword if you want to choose a different title.

When using the SELECT and PARAM keywords, you can use the following units:

Unit type	**UNIT=**	**Definition**
Distance	CM	centimeters
	DM	decameters
	FT	feet
	GPM	geopotential meters
	IN	inches
	KM	kilometers

	M	meters
	MI	miles
	MM	millimeters
	NMI	nautical miles
	YD	yards
Pressure	HPA	hectopascals
	INHG	inches of mercury
	MB	millibars
Speed	KT or KTS	knots
	MPH	miles per hour
	MPS	meters per second
Temperature	C	Celsius
	F	Fahrenheit
	K	Kelvin

The PARAM keyword contains the parameters to plot. The list can contain native and/or computed parameters.

Native parameters are those stored in the point file. They are specified in the PARAM keyword list in the format *parameter*[*unit*], where *parameter* is the parameter name in the file, and [*unit*] is the output units, which are optional but must be in square brackets if specified.

Computed parameters do not exist in the point file but are derived with a mathematical expression. They are specified in the PARAM keyword list in the format '*parameter*[*unit*]=*expression*', where *parameter* is the user-defined parameter name; [*unit*] is the label units, which are optional but must be in square brackets if specified; and *expression* is the mathematical expression consisting of math operators and one or more native parameters defined in the P*n* keywords.

The table below lists the math operators valid in the computed parameter expression in the PARAM keyword.

Math Operator	**Function**
+	addition
–	subtraction
*	multiplication
/	division
**	power
SQRT	square root
EXP	exponential (the number e raised to a power)
LOG	natural logarithm
LOG10	base 10 logarithm
SIN	sine
COS	cosine
TAN	tangent
ASIN	arcsine
ACOS	arccosine
ATAN	arctangent
ABS	absolute value
MIN	minimum of two grids
MAX	maximum of two grids

To compute a power of ten, such as 10^{2} or 10^{-3} in the math expression of a computed parameter in the PARAM keyword, specify it in either the 1e*power* or 10**(*power*) format. For example, both 1e-5 and 10**(-5) represent 10^{-5}. You must include the parenthesis with 10**(-5) for it to be interpreted correctly.

The following character limits exist for the PARAM keyword:

- 256 characters maximum for the entire PARAM keyword list
- 12 characters maximum for the computed parameter name
- 12 characters maximum for the computed parameter units

When using point/MD files with CYD (ccyyddd) date units (e.g., FO14, IRAB, ISEN, ISFC, ISHP, PIRP and SYN schema MD files created with McIDAS-XCD 7.5 and later), commands PTCON, PTCOPY, PTDISP and PTLIST default to the current year if you don't specify a four-digit year and the nearest century if you specify only the last two digits of the year. For example, if the file uses CYD units for the DAY parameter and its value is 2004260, then you can specify the DAY in the SELECT keyword as 2004260 or 04260. If the current year is 2004, you can also specify it as 260. Use the PTLIST command with FORM=PARAM to list the file's parameters and units.

The PTCON, PTCOPY, PTDISP and PTLIST commands continue to work with point/MD files using the old SYD (ssyyddd) date units (e.g., FO14, IRAB, ISEN, ISFC, ISHP, PIRP and SYN schema MD files created with versions of McIDAS-XCD prior to 7.5), with the following exception: If you want to specify DAY in the SELECT keyword and the *ss* portion of the MD file's DAY value is nonzero, then you must specify the units with DAY in the SELECT keyword. For example, if the MD file uses SYD units for DAY and its value is 7197240, then it must appear as SELECT='DAY[SYD] 7197240'. If the MD file uses SYD units for DAY but has a zero (blank) *ss* value, you don't have to specify the units in the SELECT keyword. For example, if the MD file uses SYD units for DAY and its value is 97240 then it can appear as either SELECT='DAY 97240' or SELECT='DAY[SYD] 97240'.

Examples

PTDISP SURFACE/ISFC SELECT='TIME 12' PARAM=T

This entry plots 12 UTC temperatures from the most recent point file in dataset SURFACE/ISFC. The temperatures are plotted using the navigation of the current frame and are in the units and format that the file uses for storing temperature.

PTDISP SURFACE/SYNOPTIC SELECT='TIME 18 ; DAY 96001' PARAM=T[C] FORMAT=I3 DECLUT=NO LAT=-40 -40 LON=-90 90 SLAT=-40 PRO=CONF TITLE='Antarctic Temperatures (deg C) at 18 UTC on 1 Jan 1996'

This entry plots 18 UTC temperatures in degrees Celsius from the point file storing the data for day 96001 in dataset SURFACE/SYNOPTIC. The temperatures are plotted as integers of up to three digits over a southern hemisphere map centered near the South Pole. All matching temperatures are plotted since DECLUT=NO is specified. The default title below the plot is replaced with the text *Antarctic Temperatures (deg C) at 18 UTC on 1 Jan 1996*.

PTDISP SURFACE/ISFC.5 OPP SELECT='TIME 12 ; T[F] 20 29' PARAM=T[F] FORMAT=I3 MAP=USA

This entry plots 12 UTC temperatures between 20 and 29 degrees Fahrenheit from the point file in position 5 of dataset SURFACE/ISFC. The temperatures are plotted on the frame opposite the current frame and are in degrees Fahrenheit over a map of the United States. The specified FORMAT keyword plots the temperatures as integers of up to three digits.

PTDISP SURFACE/ISFC.10 SELECT='TIME 17' PARAM=WIND T[F] TD[F] PSL[MB] ID FORMAT=FLAG I3 I3 K3 MULTIPLY=X X X 10 LSIZE=11 6 6 6 5 COLOR=7 5 4 3 2 OFFSET=0 0 -8 -13 3 -13 -8 3 3 3 MAP=MID TITLE='US Station Plot, 1700Z'

This entry displays a station plot of 17 UTC data from the point file in position 10 of dataset SURFACE/ISFC. The station plot includes wind flags, temperatures (in degrees Fahrenheit), dew point temperatures (in degrees Fahrenheit), altimeter settings (in millibars) and station IDs over a map of the Midwest. The LSIZE, COLOR and OFFSET keywords determine the plot size, graphics color level and offset from the center plot position of each of the five parameters. The altimeter settings are displayed in the standard meteorological format by using the K format to "clip" all but the last three integer digits after the observed values are multiplied by 10 with the MULTIPLY keyword. For example, an observed value of 1002.71 is displayed as 027; an observed value of 999.33 is displayed as 993.

PTDISP RAOB/MANDATORY.8 SELECT='LEV 850 ; TIME 0' PARAM=WIND T[C] TD[C] FORMAT=VECT I3 I3 LSIZE=12 7 7 COLOR=7 5 4 OFFSET=0 0 -12 2 4 2 MAP=OK PAN=1
PTDISP RAOB/MANDATORY.8 SELECT='LEV 500 ; TIME 0' PARAM=WIND T[C] TD[C] FORMAT=VECT I3 I3 LSIZE=12 7 7 COLOR=7 5 4 OFFSET=0 0 -12 2 4 2 MAP=OK PAN=2
PTDISP SURFACE/ISFC.8 SELECT='TIME 0' PARAM=WIND T[C] TD[C] FORMAT=VECT I3 I3 LSIZE=12 7 7 COLOR=7 5 4 OFFSET=0 0 -12 2 4 2 MAP=OK PAN=2 2 1

These entries create a three-panel display of 00 UTC wind vectors, temperatures and dew point temperatures from the point files in position 8 of datasets RAOB/MANDATORY and SURFACE/ISFC. The first entry displays the 850 millibar data in the upper-left quarter of the frame. The second entry displays the 500 millibar data in the upper-right quarter of the frame. The third entry displays the surface data in the bottom half of the frame. The FORMAT, LSIZE, COLOR and OFFSET keywords determine the format, plot size, graphics color level and offset from the center plot position of each of the three parameters.

PTDISP RTPTSRC/SFCHOURLY SEL='DAY #Y;TIME 18' PAR='DELP[MB]=P2-P1' P1=PSL[MB] P2=PSL[MB] 'TIME 21' FORMAT=F4.1 NAV=C

This entry plots today's 18 UTC to 21 UTC pressure change in millibars using the navigation of the current frame. The data is taken from the most recent point file in dataset RTPTSRC/SFCHOURLY.

PTDISP RTPTSRC/UPPERMAND SEL='DAY #Y;TIME 12' MAP=USA PARAM='THIK[M]=(287/9.8)*((P1+P2)/2)*LOG(1000/500)' P1=T[K] 'P 1000' P2=T[K] 'P 500'

This entry plots 1000-500 mb thickness in meters at 12 UTC today on a map of the United States. The thickness values are calculated using the equation in the PARAM keyword, known as the Hypsometric Equation.

PTDISP RTPTSRC/SFCHOURLY SEL='DAY #Y; TIME 20' MAP=USA PARAM=T[F] FORMAT=I4 COLOR=2[20-89B10]

This entry plots today's 20 UTC temperatures on a map of the United States. The temperatures are color coded: less than 20°F in graphics color level 1, 20-29°F in level 2, 30-39°F in level 3, 40-49°F in level 4, 50-59°F in level 5, 60-69°F in level 6, 70-79°F in level 7, 80-89°F in level 8, and 90°F and greater in level 9.

PTDISP RTPTSRC/SFCHOURLY SEL='DAY #Y; TIME 15' MAP=IL PARAM=WIND[KTS] FORMAT=FLAG COLOR=2[1-20B5]

This entry plots today's 15 UTC wind flags on a map of Illinois. The flags are color coded by speed: less than 1 knot in graphics color level 1, 1-5 knots in level 2, 6-10 knots in level 3, 11-15 knots in level 4, 16-20 knots in level 5, and greater than 20 knots in level 6. Note that each flag symbol may appear in two different colors because the speeds are rounded to the nearest five knot increment (i.e., 2.5-7.4 knot speeds are displayed as 5 knot flags, 7.5-12.4 knot speeds are displayed as 10 knot flags, 12.5-17.4 knot speeds are displayed as 15 knot flags, etc.). Thus, 5 knot flags can be in graphics color level 2 or 3, 10 knot flags can be in level 3 or 4, 15 knot flags can be in level 4 or 5, etc.

PTLIST

Lists point data in a dataset.

Format

PTLIST *dataset [keywords]*

Parameter

dataset ADDE dataset name and position; specify one of the following formats:

group/descriptor.position
alias.position

a *position* greater than zero represents an absolute position in the dataset; a *position* less than or equal to zero represents a relative position in the dataset based on date and time, for example, 0 is the most recent and -1 is the next most recent; the *position* ALL represents all positions in the dataset (no default for *group/descriptor* or *alias*; *position* default=ALL if DAY is specified with the SELECT keyword, otherwise default=0)

Keywords

PARam= *p1 . . pn*

list of parameters to plot; the list can contain native and/or computed parameters in the following formats:

native_parameter [*unit*]
native parameter and its output units; native parameters are those that are stored in the point file (no default for *parameter; unit* default=from *dataset*)

'*computed_parameter* [*unit*]=*expression*'
computed parameter, its output units, and the mathematical expression used to compute it; computed parameters do not exist in the point file but are derived with the specified mathematical expression containing math operators (+ - * etc.) and the native parameters specified in the Pn keywords, for example,
PAR=ID T[F] TD[F] 'TDD[F]=P1-P2' PSL[MB];
see the Remarks and the Pn keyword description for more information (no default for *parameter* or *expression*; *unit* default=from *dataset*)

SELect= '*clause1*; *clause2*; . . ; *clausen*'
finds only the data matching the values in the select clauses; single quotes are mandatory; separate select clauses with semicolons; specify *clause* as a range or a list using one of the following formats:

range: *parameter* **[***unit***]** *bval eval*

list: *parameter* **[***unit***]** *val1*,*val2*, . . ,*valn*

parameter point file parameter

[*unit***]** units for the parameter values; units are optional but must be in square brackets if specified; see the Remarks (default=from *dataset*)

bval eval range of parameter values; see the Remarks

val1, . .,*valn* list of parameter values

Mathematical Operations Keyword

Pn= *parameter* [*unit*] '*clause1*; *clause2*; . . ; *clausen*'
native parameters to use in the math expressions specified in the PARAM keyword; use a number between 1 and 99 in the keyword name (P1,P2,..,P99); each keyword specifies a single native parameter and units to substitute in matching Pn values in the math expressions, where:

parameter [*unit*]
is the native parameter name and units; *unit* defaults to those for the parameter in the dataset; there is no default for *parameter*

'*clause1*; *clause2*; ..; *clausen*'
are the clauses used to select data for this keyword; only data matching their criteria is used; they are specified exactly as in the SELECT keyword; the Pn keywords inherit the clauses specified in the SELECT keyword, therefore, you should specify a Pn keyword clause only if a value differs from that in the SELECT keyword or you need to add additional clauses; for example, to set the values of P1 and P2 to the 850 and 1000 mb temperatures in degrees Celsius at 0 UTC, respectively, specify the following keywords: SEL='TIME 0; P 850' P1=T[C] P2=T[C] 'P 1000'

Output Keywords

BYTecount= **YES** lists the number of bytes received from the server

NO does not list the number of bytes (default)

FORM= **DATA** lists parameter values in the point file (default)

FILE *bpos epos*

lists only the point file directories; all other keywords are ignored; *bpos* is the beginning position number or ALL; *epos* is the ending position number or LAST; *bpos* and *epos* override the position specified in the *dataset* parameter (*bpos* default=position specified in the *dataset* parameter, *epos* default=*bpos*)

ALL lists all point file directories

PARAM lists the parameter names, units and default formats in the point file; this option is useful for determining parameters valid with the SELECT and PARAM keywords

FORMAT= *f1 . . fn* output format for each parameter specified with the PARAM keyword; valid formats are A, I, F, E, DM and DMS; see the Remarks (default=from *dataset*)

HEAder= '*textline1*' . . '*textlinen*'
header lines at the top of the data output; replaces the default header, which consists of a line of parameter names followed by a line of dashes

NUM= number of records matching the criteria defined with the SELECT keyword to list; if you specify NUM=50, the first 50 records matching the SELECT keyword are listed (default=1)

ALL list all records matching the SELECT keyword; you may get thousands of lines of output with this option; see the Remarks

SUPpress= *s1 . . sn* suppress text from the default output; specify one or more of the following options:

DASH suppress the line of dashes from the header

HEADER suppress the entire header: parameter names and dashes

MISS suppress printing "missing" as a record value; applies only when the PARAM keyword is not specified

NUMMATCH suppress the "Number of matches found" line

Remarks

PTLIST lists data from records in one or more point data files in a dataset. The SELECT keyword defines which records to match and the PARAM keyword defines which parameters in the matching records to list. The NUM keyword defines how many matching records to list. Since each file

may contain thousands of records, you may get thousands of lines of output if you specify NUM=ALL but don't narrow the search criteria enough with the SELECT keyword.

If you use the range format for the SELECT keyword and *bval* is less than *eval*, all values between and including *bval* and *eval* are selected. For example, if you specify SELECT='LON -177 177', the longitude values -177, -176, . . , 176, 177 are selected. If *bval* is greater than *eval*, all values greater than *bval* or less than *eval*, including *bval* and *eval*, are selected. For example, if you specify SELECT='LON 177 -177', the longitude values 177, 178, 179, 180, -179, -178 and -177 are selected. This feature also allows you to select only wind directions from northwest to northeast with SELECT='DIR 315 45'.

If you do not specify the PARAM keyword, PTLIST lists all the parameters in each matching record in two columns. If you use the default listing (FORM=DATA), PTLIST reorders the output so the data is listed chronologically.

Use the FORMAT keyword options below to format output based on FORTRAN designators for data output:

Designator	**Description**
A	character
I	integer
F	floating point
E	exponential

To specify a parameter's output format, use a designator followed by a number. For example, I4 specifies an integer of up to four digits; A4 specifies up to four characters; F10.1 specifies a floating point decimal of up to 10 digits (including the decimal point) with one digit to the right of the decimal; E10.3 specifies an exponential number of up to 10 characters (including the decimal point and the four-character exponential flag) with three digits to the right of the decimal. Specify an X to use the parameter's default format. Use a number preceding an X (e.g., 0X, 1X, 4X) to specify spaces between parameters in the output. The default is one space (1X) between each parameter. If the specified integer or floating point format is too small for the parameter's value, a series of asterisks (***) is listed.

Use the DM and DMS formats to list latitude and longitude values in degrees:minutes and degrees:minutes:seconds, respectively. The default format for latitude and longitude values is decimal (which can be altered with the F format).

When using the SELECT and PARAM keywords, use the following units:

Unit type	**UNIT=**	**Definition**
Distance	CM	centimeters
	DM	decameters
	FT	feet
	GPM	geopotential meters
	IN	inches
	KM	kilometers
	M	meters
	MI	miles
	MM	millimeters
	NMI	nautical miles
	YD	yards
Pressure	HPA	hectopascals
	INHG	inches of mercury
	MB	millibars
Speed	KT or KTS	knots
	MPH	miles per hour
	MPS	meters per second
Temperature	C	Celsius
	F	Fahrenheit
	K	Kelvin

The PARAM keyword contains the parameters to output. The list can contain native and/or computed parameters.

Native parameters are those stored in the point file. They are specified in the PARAM keyword list in the format *parameter* [*unit*], where *parameter* is the parameter name in the file, and [*unit*] is the output units, which are optional but must be in square brackets if specified.

Computed parameters do not exist in the point file but are derived with a mathematical expression. Computed parameters are specified in the PARAM keyword list in the format '*parameter* [*unit*]=*expression*', where *parameter* is the user-defined parameter name; [*unit*] is the label units, which are optional but must be in square brackets if specified; and *expression* is the mathematical expression consisting of math operators and one or more native parameters defined in the Pn keywords.

The table below lists the math operators valid in the computed parameter expression in the PARAM keyword.

Math Operator	**Function**
+	addition
–	subtraction
*	multiplication
/	division
**	power
SQRT	square root
EXP	exponential (the number e raised to a power)
LOG	natural logarithm
LOG10	base 10 logarithm
SIN	sine
COS	cosine
TAN	tangent
ASIN	arcsine
ACOS	arccosine
ATAN	arctangent
ABS	absolute value
MIN	minimum of two grids
MAX	maximum of two grids

To compute a power of ten, such as 10^{2} or 10^{-3} in the math expression of a computed parameter in the PARAM keyword, specify it in either the 1e*power* or 10**(*power*) format. For example, both 1e-5 and 10**(-5) represent 10^{-5}. You must include the parenthesis with 10**(-5) for it to be interpreted correctly.

The following character limits exist for the PARAM keyword:

- 256 characters maximum for the entire PARAM keyword list
- 12 characters maximum for the computed parameter name
- 12 characters maximum for the computed parameter units

When using point/MD files with CYD (ccyyddd) date units (e.g., FO14, IRAB, ISEN, ISFC, ISHP, PIRP and SYN schema MD files created with McIDAS-XCD 7.5 and later), commands PTCON, PTCOPY, PTDISP and PTLIST default to the current year if you don't specify a four-digit year and the nearest century if you specify only the last two digits of the year. For example, if the file uses CYD units for the DAY parameter and its value is 2004260, then you can specify the DAY in the SELECT keyword as 2004260 or 04260. If the current year is 2004, you can also specify it as 260. Use the PTLIST command with FORM=PARAM to list the file's parameters and units.

The PTCON, PTCOPY, PTDISP and PTLIST commands continue to work with point/MD files using the old SYD (ssyyddd) date units (e.g., FO14, IRAB, ISEN, ISFC, ISHP, PIRP and SYN schema MD files created with versions of McIDAS-XCD prior to 7.5), with the following exception: If you want to specify DAY in the SELECT keyword and the *ss* portion of the MD file's DAY value is nonzero, then you must specify the units with DAY in the SELECT keyword. For example, if the MD file uses SYD units for DAY and its value is 7197240, then it must appear as SELECT='DAY[SYD] 7197240'. If the MD file uses SYD units for DAY but has a zero (blank) *ss* value, you don't have to specify the units in the SELECT keyword. For example, if the MD file uses SYD units for DAY and its value is 97240 then it can appear as either SELECT='DAY 97240' or SELECT='DAY[SYD] 97240'.

Examples

PTLIST SURFACE/ISFC FORM=FILE ALL

This entry lists the directories for all point files in dataset SURFACE/ISFC.

PTLIST SURFACE/SHIP.1 FORM=PARAM

This entry lists the names, units and default formats for all parameters in the point file in position 1 of dataset SURFACE/SHIP. Use this

information to determine the valid parameters for the SELECT and PARAM keywords.

PTLIST SURFACE/ISFC

This entry lists all the parameters and their values in the first record of the most recent point file in dataset SURFACE/ISFC.

PTLIST SURFACE/ISFC SELECT='ID KSAN ; TIME 0,6,12,18' PARAM=ID ST DAY TIME HMS T TD DIR SPD WX NUM=ALL

This entry lists San Diego's 0, 6, 12 and 18 UTC data from the most recent point file in dataset SURFACE/ISFC. The station ID, state, day, time, observation time, temperature, dew point temperature, wind direction, wind speed and weather type are listed from all matching records.

PTLIST SURFACE/ISFC.5 SELECT='ID KSAN ; TIME 0,6,12,18' PARAM=ID ST DAY TIME HMS T TD DIR SPD WX1 NUM=ALL

This entry requests the same information as the previous entry, but searches for data in the point file in position 5 of dataset SURFACE/ISFC.

PTLIST SURFACE/ISFC.ALL SELECT='ID KSAN ; TIME 0,6,12,18' PARAM=ID ST DAY TIME HMS T TD DIR SPD WX1 NUM=ALL

This entry requests the same information as the previous two entries, but searches for data in all the point files in dataset SURFACE/ISFC.

PTLIST SURFACE/ISFC SELECT='ID KSAN ; TIME 0,6,12,18 ; DAY 96120 96122' PARAM=ID ST DAY TIME HMS T TD DIR SPD WX1 NUM=ALL

This entry lists San Diego's 0, 6, 12 and 18 UTC data from days 96120 through 96122; it searches for data in all the point files in the dataset SURFACE/ISFC. You do not need to specify .ALL as the dataset position because you specified DAY in the SELECT keyword.

PTLIST SURFACE/ISFC SELECT='TIME 10 13 ; CO US,CA,MX ; T[F] -100 32' PARAM=CO ID DAY HMS T[F] T[C] WX1 FORMAT=X 2X X 2X X 2X X 2X F5.1 2X F5.1 2X NUM=ALL

This entry lists data for all stations in the United States, Canada and Mexico reporting a temperature below 32 degrees Fahrenheit between 10 and 13 UTC; it searches for data in the most recent point file in dataset SURFACE/ISFC. The country, station ID, day, observation time, temperature and weather type are listed. Temperature is listed twice: in degrees Fahrenheit and Celsius. The values specified in the FORMAT keyword result in two spaces between each parameter and both temperatures being listed to the nearest tenth degree.

PTLIST RTPTSRC/SFCHOURLY SEL='DAY #Y;TIME 18;ST CA' PAR=ID T[F] TD[F] 'DEPR[F]=P1-P2' P1=T[F] P2=TD[F] FORMAT=X I4 I5 I7 NUM=ALL

This entry lists the temperature, dew point temperature, and dew point depression for stations in California at 18 UTC today. The wider I5 and I7 formats for dew point temperature and dew point depression are specified so the units (F) for those parameters appear in the header at the top of the output.

PTLIST RTPTSRC/SFCHOURLY SEL='DAY #Y;TIME 12;ST MI' PARAM=ID 'RH[%]=(EXP(5423*(1/P1-1/P2))*100)' T[K] TD[K] P2=TD[K] P1=T[K] FORMAT=X I5 F5.1 F5.1 NUM=ALL

This entry lists the relative humidity, temperature, and dew point temperature for stations in Michigan at 12 UTC today. The relative humidity is calculated using the RH equation in the PARAM keyword.

R

Resizes the Image Window to the size of the current frame.

Remarks

To resize the Image Window,

Press: **Alt R**

or

Type: **R**

Press: **Enter**

The R command resizes the Image Window to show the entire displayed frame. This command is most useful if you manually changed the size of the Image Window, switched to a different-sized frame, or used the Z command to zoom in on the frame.

If you specified the **-autoResize** flag at startup or in your **.mcidasrc** file, the Image Window is automatically resized to show the entire frame when you switch to a different-sized frame or zoom frames.

RAOBCON

Contours upper-air data from IRAB-schema point files.

Formats

RAOBCON *param level map time day [keywords]*
RAOBCON *par1-par2 lev1-lev2 map time1-time2 day1-day2 [keywords]*

Parameters

param	parameter to contour; valid parameters are those defined in the **RAOBCON.CORE**, **RAOBCON.SITE** and **RAOBCON.USER** files; see the Remarks for more information and a list of the parameters in **RAOBCON.CORE** (default=T)
par1-par2	parameter difference to contour, e.g., T-TD for dewpoint depression; see the Remarks
level	level of the data; valid options are the mandatory levels: SFC, TRO1, TRO2, MAXW, 1000, 925, 850, 700, 500, 400, 300, 250, 200, 150, 100, 70, 50, 30, 20, 10 (default=500)
lev1-lev2	level difference to contour, e.g., 1000-500 to subtract 500mb data from 1000mb data; see the Remarks
map	map for the contour; see the MAP command for a list of valid predefined maps, for example, FL, MID, USA, WORL (default=do not draw a map; use the navigation of the frame specified with NAV keyword)
time	observation time (default=0 if current time is between 0 UTC and 11:59 UTC; otherwise default=12)
time1-time2	time difference to contour, e.g., 12-6 to subtract 6 UTC data from 12 UTC data; see the Remarks
day	observation date; valid formats include YYYYDDD, YYYY/MM/DD, and DD/MON/YYYY where YYYY, DDD, MM, and DD are integers and MON is three or more letters, for example, JAN or JANUARY (default=current)
day1-day2	day difference to contour, e.g., 2000110-2000109 to subtract day 2000109 data from day 2000110 data; see the Remarks

---

Keywords

CINt=	*num*	contour interval; specify 0 to let RAOBCON compute a reasonable interval; specify your own interval as an integer or decimal value greater than zero (default=from the **CONTOUR.DEF** text file; if no file entry matches the parameter, level, and units, default=0)
	string	contour values; specify the name of a string that contains up to 64 values; e.g., CINT=SNOWDEPTH will draw contours at the values specified in the string named SNOWDEPTH
COLor=		graphics color level for the display (default=2)
DASh=	**ALL**	dash all contours
	NEG	dash negative contours only
	POS	dash positive contours only
DATaset=		ADDE point dataset name and position; specify as *group/descriptor.position* or *alias.position* where *position* is a positive integer (absolute position), 0 or a negative integer (relative position), or ALL (default=RTPTSRC/UPPERMAND.ALL)
GRA=		graphic frame number for the display (default=current)
LSIze=		contour label height, in pixels (default=8)
NAV=	*num*	frame to use for navigation

C use the current frame's navigation (default)

you cannot use the NAV keyword if you specify a map with the *map* parameter

UNIt= units for the specified parameter; see the Remarks for defaults; run the CU LIST command to list the valid units and abbreviations for the parameters that can be changed

---

Remarks

RAOBCON builds and runs a PTCON command to display mandatory-level upper-air data from IRAB-schema point files. RAOBCON accepts all the keywords in the PTCON command except LAT, LON, MAP, PRO, SLAT, SLON and PARAM because they may conflict with RAOBCON's parameters or keywords. You can use the PTCON SELECT keyword to limit the analysis to a specified list or range of values, but you cannot specify a different parameter, level, time, day or unit than those defined by the RAOBCON parameters or keywords.

The table below lists the parameter name, default units and a description of each parameter defined in the **RAOBCON.CORE** file. Default units enclosed in parenthesis cannot be changed. The **.CORE** file is supplied with McIDAS-X and should not be modified. The **.USER** file overrides the **.SITE** file, which overrides the **.CORE** file. This order of precedence allows you or your site to change existing parameters and/or add new parameters in your **.USER** or **.SITE** file. To create a **.USER** or **.SITE** file, copy the **.CORE** file from **~mcidas/data** to your **$HOME/mcidas/data** directory (or another **MCPATH** directory), rename it, then edit the existing predefined parameters and/or add new ones.

param	Default Units	Description
ELEV	M	station elevation
P	MB	pressure
Z	M	height from surface
T	C	temperature
TD	C	dew point temperature
MIX	(G/KG)	mixing ratio
THA	(K)	potential temperature
THAE	(K)	equivalent potential temperature
DIR	(degrees)	wind direction
SPD	KTS	wind speed
STREAML		streamlines
ABV	(sec^{-1})	absolute vorticity
VOR	(sec^{-1})	vorticity
DVG	(sec^{-1})	divergence

Parameter advection or divergence is done by appending ADV or DVG to the parameter name, e.g., TADV and SPDDVG. Parameters with their default units enclosed in parenthesis in the table above (e.g., MIX, THA and ABV) and advected or diverged parameters cannot be used with the differencing options (*par1-par2*, *lev1-lev2*, *time1-time2* and *day1-day2*).

If you specify both a time difference (*time1-time2*) and a day difference (*day1-day2*), RAOBCON contours the values equal to the parameter and level at *time1,day1* minus the parameter and level at *time2,day2*. For example, RAOBCON Z 500 USA 0-12 2000123-2000122 contours the values equal to the 500-mb heights at 0 UTC, day 2000123 minus the 500-mb heights at 12 UTC, day 2000122.

If you specify a parameter difference (*par1-par2*) with a time difference and day difference, RAOBCON contours the values equal to the parameter difference at *time1,day1* minus the parameter difference at *time2,day2*. For example, RAOBCON T-TD 500 USA 0-12 2000123-2000122 contours the values equal to the 500-mb dewpoint depression at 0 UTC, day 2000123 minus the 500-mb dewpoint depression at 12 UTC, day 2000122. If you also include a level difference (*lev1-lev2*), RAOBCON does the computations for both levels. For example, RAOBCON T-TD 850-500 USA 0-12 2000123-2000122 contours the values computed by the equation [A-B]-[C-D], where:

A = 850-mb dewpoint depression at 0 UTC, day 2000123
B = 500-mb dewpoint depression at 0 UTC, day 2000123
C = 850-mb dewpoint depression at 12 UTC, day 2000122
D = 500-mb dewpoint depression at 12 UTC, day 2000122

Examples

RAOBCON

This entry contours 500-mb temperatures for the current day and time using the navigation of the current frame.

RAOBCON P SFC USA 12 UNIT=INHG COLOR=7 LABEL='Surface Pressure, in inches of mercury' 7 420 150

This entry contours surface pressure in inches of mercury at 12 UTC over a map of the United States. The contours are drawn in graphics color level 7. The default label below the plot is replaced with the text *Surface Pressure, in inches of mercury* near the center of the frame in graphics color level 7. Label is a PTCON keyword

TE DEWPOINTS "-20 -10 0 10 20 25 30 32 35

RAOBCON TD 700 CINT=DEWPOINTS DASH=NEG UNIT=F

The TE command creates a string called DEWPOINTS containing contour values for the RAOBCON entry. The RAOBCON entry draws the 700-mb 20, -10, 0, 10, 20, 25, 30, 32, and 35° Fahrenheit dew-point temperature contours for the current day and time using the navigation of the current frame. The -20° and -10° contours are dashed; the others are solid. Do not specify a pound sign (#) before the string name in the CINT keyword.

RAOBCON Z 500 USA 12 17/JAN/1996 DATASET=CLIMATE/1996

This entry plots 500-mb temperatures over a map of the United States. The data is from 12 UTC on 17 January 1996 in dataset CLIMATE/1996.

RAOBCON VOR 500 USA 0

This entry contours today's 0 UTC 500-mb relative vorticity over a map of the United States.

RAOBCON Z 500 EURO 0 2000131-2000130

This entry contours the 24-hour 500-mb height change from 0 UTC, day 2000130 to 0 UTC, day 2000131 over a map of Europe.

RAOBPLOT

Plots upper-air data from IRAB-schema point files.

Formats

RAOBPLOT *param level map time day [keywords]*
RAOBPLOT *par1-par2 lev1-lev2 map time1-time2 day1-day2 [keywords]*

Parameters

param	parameter to plot; valid parameters are those defined in the **RAOBPLOT.CORE**, **RAOBPLOT.SITE** and **RAOBPLOT.USER** files; see the Remarks for more information and a list of the parameters in **RAOBPLOT.CORE** (default=PLOT)
par1-par2	parameter difference to plot, e.g., T-TD for dewpoint depression; see the Remarks
level	level of the data; valid options are the mandatory levels: SFC, TRO1, TRO2, MAXW, 1000, 925, 850, 700, 500, 400, 300, 250, 200, 150, 100, 70, 50, 30, 20, 10 (default=500)
lev1-lev2	level difference to plot, e.g., 1000-500 to subtract 500mb data from 1000mb data; see the Remarks
map	map for the contour; see the MAP command for a list of valid predefined maps, for example, FL, MID, USA, WORL (default=do not draw a map; use the navigation of the frame specified with NAV keyword)
time	observation time (default=0 if current time is between 0 UTC and 11:59 UTC; otherwise default=12)
time1-time2	time difference to plot, e.g., 12-6 to subtract 6 UTC data from 12 UTC data; see the Remarks
day	observation date; valid formats include YYYYDDD, YYYY/MM/DD, and DD/MON/YYYY where YYYY, DDD, MM, and DD are integers and MON is three or more letters, for example, JAN or JANUARY (default=current)
day1-day2	day difference to plot, e.g., 2000110-2000109 to subtract day 2000109 data from day 2000110 data; see the Remarks

---

Keywords

COLor= graphics color level (default=3); specify as a single number, e.g., COLOR=5, or in the format *color*[*min-max*B*inc*] to assign multiple colors for the parameter, based on its values; e.g., COLOR=2[30-59B10] plots values 30-39 in level 2, values 40-49 in level 3, values 50-59 in level 4, values less than 30 in level 1, and values greater than 59 in level 5

DATaset= ADDE point dataset name and position; specify as *group*/*descriptor*.*position* or *alias*.*position* where *position* is a positive integer (absolute position), 0 or a negative integer (relative position), or ALL (default=RTPTSRC/UPPERMAND.ALL)

DEClut= **YES** declutter the plot, meaning do not plot overlapping values (default)

NO do not declutter the plot

GRA= graphic frame number for the display (default=current)

LSIze= height of the plot characters, in pixels (default=8)

NAV= *num* frame to use for navigation

C use the current frame's navigation (default)

you cannot use the NAV keyword if you specify a map with the *map* parameter

UNIt= units for the specified parameter; see the Remarks for defaults; run the CU LIST command to list the valid units and abbreviations for the parameters that can be changed

Remarks

RAOBPLOT builds and runs a PTDISP command to display mandatory-level upper-air data from IRAB-schema point files. RAOBPLOT accepts all the keywords in the PTDISP command except LAT, LON, MAP, PRO, SLAT, SLON and PARAM because they may conflict with RAOBPLOT's parameters or keywords. You can use the PTDISP SELECT keyword to limit the plot to a specified list or range of values, but you cannot specify a different parameter, time, day or unit than those defined by the RAOBPLOT parameters or keywords.

The table below lists the parameter name, default units and a description of each parameter defined in the **RAOBPLOT.CORE** file. Default units enclosed in parenthesis cannot be changed. The **.CORE** file is supplied with McIDAS-X and should not be modified. The **.USER** file overrides the **.SITE** file, which overrides the **.CORE** file. This order of precedence allows you or your site to change existing parameters and/or add new parameters in your **.USER** or **.SITE** file. To create a **.USER** or **.SITE** file, copy the **.CORE** file from **~mcidas/data** to your **$HOME/mcidas/data** directory (or another **MCPATH** directory), rename it, then edit the existing predefined parameters and/or add new ones.

param	**Default Units**	**Description**
IDN	(number)	five-digit station ID
OBS	(dots)	observation locations
ELEV	M	station elevation
OBTIME	(HMS)	actual time of observation
PLOT	M	station plot of the IDN, T, TD, Z, and WINDB parameters; units are M, for metric, or A, for American
P	MB	pressure
Z	M	height from surface
T	C	temperature
TD	C	dew point temperature
MIX	(G/KG)	mixing ratio
THA	(K)	potential temperature
THAE	(K)	equivalent potential temperature
DIR	(degrees)	wind direction
SPD	KTS	wind speed
WIND	(deg) KTS	two-digit wind direction and two- or three-digit speed, in the form DDSSS
WINDB	(barb)	wind barb (DIR and SPD); flag=25 mps (~50 kts), long barb=5 mps (~10 kts), short barb=2.5 mps (~5 kts) kts
WINDV	(vector)	wind vector (DIR and SPD)

If you specify both a time difference (*time1-time2*) and a day difference (*day1-day2*), RAOBPLOT plots the values equal to the parameter and level at *time1,day1* minus the parameter and level at at *time2,day2*. For example, RAOBPLOT Z 500 USA 0-12 2000123-2000122 plots the values equal to the 500-mb heights at 0 UTC, day 2000123 minus the 500-mb heights at 12 UTC, day 2000122.

If you specify a parameter difference (*par1-par2*) with a time difference and day difference, RAOBPLOT plots the values equal to the parameter difference at *time1,day1* minus the parameter difference at *time2,day2*. For example, RAOBPLOT T-TD 500 USA 0-12 2000123-2000122 plots the values equal to the 500-mb dewpoint depression at 0 UTC, day 2000123 minus the 500-mb dewpoint depression at 12 UTC, day 2000122. If you also include a level difference (*lev1-lev2*), RAOBPLOT does the computations for both levels. For example, RAOBPLOT T-TD 850-500 USA 0-12 2000123-2000122 plots the values computed by the equation [A-B]-[C-D], where:

A = 850-mb dewpoint depression at 0 UTC, day 2000123

B = 500-mb dewpoint depression at 0 UTC, day 2000123

C = 850-mb dewpoint depression at 12 UTC, day 2000122

D = 500-mb dewpoint depression at 12 UTC, day 2000122

Parameters with their default units enclosed in parenthesis in the table above (e.g., IDN, MIX and THA) cannot be used with the differencing options (*par1-par2*, *lev1-lev2*, *time1-time2* and *day1-day2*).

Examples

RAOBPLOT

This entry plots 500-mb station model data for the current day and time using the navigation of the current frame.

RAOBPLOT IDN SFC USA LSIZE=6

This entry plots station ID numbers over a map of the United States with a plot size of six pixels.

RAOBPLOT T 850 NA DECLUT=NO UNIT=K LSIZE=6

This entry plots 850-mb temperatures in degrees Kelvin over a map of North America for the current day and time. The data is drawn with a plot size of six pixels. It is not decluttered, so all values are plotted and some may overlap.

RAOBPLOT WINDB 200 WORL 0 SELECT='SPD[KTS] 80 300' COLOR=5 LSIZE=10 TITLE='0 UTC 200-mb winds of at least 80 knots' 7 420 150

This entry plots wind barbs for all 0 UTC 200-mb winds with speeds of 80 knots or greater over a map of the world. The barbs are drawn in graphics color level 5 with a plot size of 10 pixels. The default title below the plot is replaced with the text *0 UTC 200-mb winds of at least 80 knots* near the center of the frame in graphics color level 7. SELECT and TITLE are PTDISP keywords.

RAOBPLOT TD 700 USA 12 17/JAN/1996 DATASET=CLIMATE/1996

This entry plots 700-mb dew point temperatures over a map of North America. The data is from 12 UTC on 17 January 1996 in dataset CLIMATE/1996.

RAOBPLOT Z 500 USA 12 2000130-2000129

This entry plots the 24-hour 500-mb height change from 12 UTC, day 2000129 to 12 UTC, day 2000130 over a map of the United States.

RAOBPLOT Z 500-1000 NA 0

This entry plots today's 0 UTC 1000/500-mb thickness over a map of North America.

RAOBRPT

Lists upper-air (RAOB) observations.

Format

RAOBRPT *id1 . . idN nhours [keywords]*

Parameters

id1 . . idN station IDs or states to list; use the two-character state P.O. abbreviation for states

nhours number of hours to search for observations starting from DAY and TIME (default=12)

Keywords

CO= *co1 . . coN* lists observations for all stations in these countries; use the CCODE command to list valid country codes

DATaset= ADDE dataset name; specify as *group/descriptor* or *alias* (default=RTWXTEXT/UPPERAIR)

DAY= Julian date to start search; use either the 5- or 7-digit format, for example, 99300 or 1999300 (default=current)

TIMe= UTC time to start search; valid times are 0, 3, 6, 9, 12, 15, 18 and 21 (default=current)

TYPe= *type1 . . typeN*

list the upper-air data by type

- **ALL** all types of upper-air data (default)
- **PPAA** mandatory-level wind data between the surface and 100 mb
- **PPBB** significant-level wind data between the surface and 100 mb
- **PPDD** significant-level wind data above 100 mb
- **TTAA** mandatory temperature and wind data
- **TTBB** significant-level temperature data between the surface and 100 mb
- **TTCC** mandatory-level temperature data above 100 mb
- **TTDD** significant-level temperature data above 100 mb

Remarks

You must specify at least one station ID, state, or country.

If you use the DAY keyword to specify a day other than the current day, you must also use the TIME keyword.

Examples

RAOBRPT 72435

This entry lists the latest upper-air observations of all types for Paducah, Kentucky, from the past 12 hours.

RAOBRPT SC 18 CO=FR TYPE=TTAA

This entry lists all TTAA upper-air observations for South Carolina and France from the past 18 hours.

RAOBRPT 72645 72201 LA 12 DAY=96300 TIME=12 TYPE=TTAA TTBB

This entry lists all TTAA and TTBB upper-air observations for Green Bay, Wisconsin; Key West, Florida; and Louisiana from 12 hours prior to 12:00 UTC on day 96300.

READ

Lists the contents of an ADDE text dataset.

Format

READ *dataset*

Parameter

dataset ADDE dataset name; specify one of the following formats:

group/descriptor
alias

Remarks

The READ command lists the contents of a text dataset. A text dataset points to a single text file. If the dataset is on a remote server, use the DATALOC command to map the group name to the remote server workstation.

Example

READ ADMIN/GOES9

This entry lists the contents of the text dataset ADMIN/GOES9.

REDIRECT

Writes the LW file path redirection memory block.

Formats

REDIRECT ADD *file "path*
REDIRECT CLEAR *type*
REDIRECT DELETE *file*
REDIRECT LIST *type*
REDIRECT MAKE
REDIRECT REST *rfile*
REDIRECT SAVE *rfile*

Parameters

ADD	adds a redirection entry to the file **LWPATH.NAM** and dynamic memory	
CLEAR	clears the redirection entries	
DELETE	deletes a redirection entry from dynamic memory and the file **LWPATH.NAM**	
LIST	lists the file redirection entries	
MAKE	writes the redirection entries from **LWPATH.NAM** into dynamic memory (default)	
REST	restores a saved redirection file into file **LWPATH.NAM** and dynamic memory	
SAVE	saves the redirection entries to a file	
file	file name, e.g., **MDXX3074**, **AREA011*** ; 12 characters maximum including the extension; if you include an extension, it must be three or less characters (use a wildcard to match extensions longer than three characters)	
"path	desired directory path, 59 characters maximum; the double quote is mandatory	
type	**FILE**	the file **LWPATH.NAM**
	MEM	dynamic memory (default for the LIST option)
	ALL	both **LWPATH.NAM** and dynamic memory; use with the CLEAR option only (default for CLEAR)
rfile	redirection file name; if an extension is not specified, .NAM is automatically appended	

Remarks

File redirection is used to define which directories McIDAS commands search when looking for specified data and help files. When a command requests data from a file (for example, when IMGLIST displays image data from the file AREA0001), the active redirect table is checked to see if one of its entries has a matching file name. If so, the command searches for the file in the directory specified in the entry.

The active redirect table resides in memory. You can create multiple redirect files, which can be individually restored to the active redirect table. Redirect files are created with a text editor or with the command REDIRECT ADD. Redirect files consist of one or more entries, each containing a file name and a directory or path. The entry's file name can contain wild card characters (for example, A*, AREA*, MDXX05*, *.GIF). Lines beginning with a double quote (") are comments (for example, "This is a comment). Comments can only be added with a text editor; not with REDIRECT ADD. Redirect files cannot contain blank lines.

The maximum length of an entry is 72 characters, 12 for the file name and 60 for the directory. The 60th character can only be a slash (/).

If your workstation is hung and you suspect file redirection is causing it, first exit McIDAS, then delete the file **LWPATH.NAM** and boot the system. This file is located in the **$HOME/mcidas/data** or another **MCPATH** directory.

Caution! It is safest to redirect only data files, for example, areas and MD files. Do not redirect the source file specified with the DOSTOLW command or the destination file specified with the LWTODOS command. Redirecting McIDAS system files may cause problems. Some of the systems files you should NOT redirect are listed below.		
ALLOC.WWW	**FRNVEXP.***nnn*	**STRTABLE**
CONTEXT.SLT	**LWPATH.NAM**	**SYSIMAGE.PM**
EXPORT.*nnn*	**MCRGB.TXT**	**TERMCHAR.***nnn*
FRAMED.*nnn*	**MENU**	**UCTERM.***nnn*
FRAMENH.*nnn*	**NUM.FRM**	**VERSION.TXT**
FRNV.*nnn*	**SKEDFILE**	**WINQUFIL**
FRNVDIR.001	**STARTUP.SYS**	

Examples

REDIRECT

This entry places the redirected files and paths from the active redirection file (**LWPATH.NAM**) into dynamic memory.

REDIRECT LIST FILE

This entry lists the redirection entries in the file **LWPATH.NAM**.

REDIRECT CLEAR

This entry clears the file **LWPATH.NAM** and dynamic memory.

REDIRECT ADD AREA137* "/u/smith/volume/mcidas/AREAS

This example adds the redirection path **/u/smith/volume/mcidas/AREAS/AREA137*** to dynamic memory and the file **LWPATH.NAM**.

REDIRECT REST TSTVL1

This entry brings the file **TSTVL1.NAM** into dynamic memory and the redirection file **LWPATH.NAM**.

REDIRECT ADD AREA* "/u/johndoe/mcidas/areas

REDIRECT ADD GRID8* "/u/johndoe/mcidas/gridfiles

REDIRECT ADD MDXX1000 "/u/johndoe/mcidas/mdfiles

REDIRECT ADD *.ET "/u/johndoe/mcidas/enhancements

REDIRECT SAVE DATA.NAM

REDIRECT REST DATA.NAM

These entries create redirect entries in the active redirect table and then saves the table as the file **DATA.NAM**. The last entry makes the file the active redirect table so it can be used.

To create the same **DATA.NAM** redirect file using a text editor, enter the lines below in **DATA.NAM** and save the file.

```
AREA* /u/johndoe/mcidas/areas GRID8* /u/johndoe/mcidas/gridfiles MDXX1000
/u/johndoe/mcidas/mdfiles *.ET /u/johndoe/mcidas/enhancements
```

To use the file once it has been created, restore it to the active redirect table by running the command REDIRECT REST DATA.NAM.

REPEAT

Repeats a McIDAS command.

Format

REPEAT *string start* **TO** *end* **BY** *inc start* **BY** *inc*
. . start **BY** *inc [keyword]*

Parameters

string	string name containing the commands to run; 1 to 12 characters; any string beginning with these characters is run
start	starting parameter value (default=0)
end	ending parameter value (default=0)
inc	value to increment the parameter with each repetition (default=1)
TO	mandatory parameter indicating the range between *start* and *end*
BY	mandatory parameter indicating the increment between *start* and *end*

Keyword

MAXN=	maximum number of commands to run (default=200)

Remarks

Before running REPEAT, set up your *string* using **!** *number* for each parameter to be incremented; *number* must begin at 1 and increment by 1. An exclamation point (**!**) followed by an integer in a string will increment the parameter on each run. Parameters may be decremented using a minus sign (–).

Examples

If the string named LOAD contains the entry "IMGDISP A.!1 !2", then entering the following command, displays image A.1 in frame 3, A.2 in frame 5, A.3 in frame 7, and A.4 in frame 9.

REPEAT LOAD 1 TO 4 BY 1 3 BY 2

RUN

Runs a McBASI program.

Formats

RUN *p1 . . pn* **FILE=** *[keywords]*
' *filename p1 . . pn [keywords]*
RUN "*statement*
' "*statement*

Parameters

p1 . . pn	positional parameters read by the INPUT statement
filename	name of the McBASI program to run; the extension defaults to .MCB
"*statement*	McBASI statement to run immediately; 256 characters maximum

---

Keywords

Clear=	**YES**	resets the values of the variables to zero or null before starting the program (default when running a program with the FILE keyword)
	NO	variables retain the values from the previous RUN (default when running the statement immediately)
File=		name of the McBASI program to run; the extension defaults to .MCB; this keyword is required when using RUN, but is invalid when using the single quote format (')
Var=		name of the LW file to store the variable values in after running a program; the file contents are used to initialize the variables unless CLEAR=YES is specified (default=BASICVV1)
	SHARE	shares variables when one McBASI program runs another

Remarks

RUN will run a tiny BASIC language program prepared using an ASCII editor. The BASIC reserved words are: KEYIN, PRINT, INPUT, LET, GOTO, GOSUB, RETURN, SLEEP, STOP, IF, OPEN, CLOSE, REM, PLOT, PTABLE and POKE. (See Chapter 4, *McBASI Interpreter,* for more information.)

The arithmetic operators are: + – * / **

These are the available functions and pseudofunctions: CHR$, SIN, COS, TAN, ASIN, ACOS, ATAN, EXP, LOG, LN, ABS, BIN, BIN$, INDEX, LEN, VAL, STR$, MID$, TIME$, DATE$, DAY$, EOF$, TFILE$, TFILD$, TFILC$, TFILB$, TFILA$, MOD, NINT, INT, SQRT, PEEK, PEEK$, KSYS$, KSYS and TABLE$.

See Also

See Chapter 4, *McBASI Interpreter*, for more information on McBASI interpreter commands, functions and sample programs.

Examples

RUN "?SQRT (5.7)

This entry calculates and prints the value of the square root of 5.7. The question mark is a substitute for PRINT.

'MYPROG 27 #H

This entry runs the program **MYPROG.MCB** and passes the value 27 and the current time (the string #H) to the INPUT statements in the program.

RUN 27 #H FILE=MYPROG

This entry performs the same function as the previous one. Use this format for programs started from the system scheduler.

'MYPROG CLEAR=NO VAR=MYVARS.VAR

This entry runs the program **MYPROG.MCB** but recovers the values for all variables from the file **MYVARS.VAR**. The values are written back to this file when the program ends.

RUN VAR=MYVARS.VAR FILE=MYPROG

This entry runs the program **MYPROG.MCB**, initializes all variables to zero, and writes the values of the variables into the file **MYVARS.VAR** when the program ends.

RUN "PTABLE MID$(TIME$,1,2),"HR":KEYIN "TL HR

This entry sets the value of the string HR to the current hour, then lists the value of the string. It can be run once an hour from the scheduler (SKE command) to keep the string's value current for use in commands like PTLIST and PTDISP.

RUN "T=VAL(MID$(TIME$,1,2)):M=T+100:KEYIN"PTCOPY SFC/HOURLY LOCAL/MD.";M" SEL,'CO US;TIME ";T"' DEL,YES

This entry sets the value of T to the current hour and the value of M to the current hour plus 100. It then copies the data for TIME=T for the United States from the most recent point file in the dataset SFC/HOURLY to the dataset LOCAL/MD in the position number equal to the value of M.

For example, if the current time is 12:50 UTC, this entry copies the 12 UTC data for the United States to position 112 of dataset LOCAL/MD. If this entry is run hourly throughout the day, each hour's data is copied to the corresponding position between 100 and 123 in dataset LOCAL/MD.

RVF

Restores virtual frames saved with command SVF.

Format

RVF *bframe eframe name*

Parameters

bframe	beginning frame number to load (default=1)
eframe	ending frame number to load (default=*bframe*)
name	file name used to save the frames with the SVF command; if no extension is specified, .PIX is automatically appended

---

Remarks

RVF restores the image, graphics, frame directory, navigation, image enhancement and the graphics table that were present on each of the frames when they were saved with command SVF. Full image and graphics functionality is retained so you can use commands like EG, ERASE, EU, GU, K and W. If you're running a session with independent graphics, you can also use commands like SF, SG, J and Y to display or loop the restored image and graphics frames independently.

Virtual frames can only be restored to frames the same size as the original. Use command LVF with FORM=ALL to list the size of each of the virtual frames so you can restore them to frames of the correct size with RVF.

The frames specified in RVF acquire the panel configuration that the virtual frames had when they were saved with the SVF command. For example, if a virtual frame with four panels is restored to a single-panel frame, the frame is reconfigured to have four panels.

Frames saved with SVF on little endian workstations (e.g., PC systems) can only be restored with RVF on the same workstation or other little endian workstations. Likewise, frames saved with SVF on big endian workstations (e.g., Sun SPARCstations and IBM RS/6000s) can only be restored with RVF on the same workstation or other big endian workstations.

See Also

Use the SVF command to save virtual frame files. Use LVF to list virtual frame files.

Examples

RVF 1 5 PICTURE

This entry reloads frames 1 through 5 from the file **PICTURE.PIX**.

S

Lists Metar, Synoptic and RAOB stations within the cursor bounds.

Remarks:

To list the coordinates at the center of the cursor and the Metar, Synoptic and RAOB stations within the cursor bounds,

Press: **Alt S**

or

Type: **S**

Press: **Enter**

The S command is provided as a fast way to enter (a "shortcut" to) the commonly-run command STNLIST TYPE=METAR SYNOPTIC RAOB.

See Also

Use the STNPLOT command to plot stations on the graphics frame. Use the STNLIST command for more sophisticated station listing options.

SCHE

Adds a schema to the schema file.

Format

SCHE *name*

Parameter

name name of the text or LW format schema file

Remarks

You can create a text format schema file using a text editor and place it in the **mcidas/data** subdirectory. Then register the new schema file using the SCHE command. Use the LSCHE command to list registered schemas.

You can also register LW format schema files like the DC* files (e.g., **DCISFC** and **DCSYN**) distributed with McIDAS-X. You don't, however, need to convert your locally created text format schema files to LW format, as the SCHE command works with both text and LW format schema files.

See Also

See *Registering MD File Schemas* for more information.

Examples

SCHE DCISFC

This entry adds schema ISFC, which is the schema named in LW file DCISFC, to the workstation schema file. The current version of this schema is included with each upgrade.

SCHE MYSCHEMA.TXT

This entry adds the schema named in the text file **MYSCHEMA.TXT** to the workstation schema file.

SEE

Lists the text in an LW file.

Format

SEE *file nchars [keywords]*

Parameters

file name of the LW file to list

nchars number of characters per line (default=80)

Keywords

BEGin= first line number to list; if negative, the number of the first byte to list (default=0)

LASt= number of lines from the end of the file to list

NLInes= number of lines to list (default=500)

ALL lists all lines

STEp= number of lines skipped between the lines listed; if negative, it steps backwards; this keyword only works on 80-character ASCII LW files (default=1)

Remarks

Command SEE attempts to list both LW files (80-character records with no linefeeds) and text files (up to 80 characters with imbedded linefeeds). However, the keywords may not work properly with text files.

You can use the ADDE command READ in place of SEE.

Examples

SEE LOGON

This entry lists file LOGON starting at line zero (the beginning) stepping forward one line until it reaches the end of the file.

SEE MYFILE BEGIN=25

This entry lists the text in LW file MYFILE beginning with line 25.

SEE SCHEMA BEGIN=99 NLINES=15 STEP=-1

This entry lists file SCHEMA starting at line 99, stepping backwards by one line for 15 lines.

SF

Displays a frame.

Format

SF *frame*

Parameter

frame frame number to display (default=1)

Remarks

On McIDAS-X sessions with independent graphics, SF displays the specified image frame. Use the SG command to display a graphic frame. See the TERM command for more information.

See Also

Use the F command to list the current frame number.

Examples

SF

This entry displays frame 1.

SF 12

This entry displays frame 12.

SFCCON

Contours surface hourly data from ISFC-schema point files.

Formats

SFCCON *param map time day [keywords]*
SFCCON *par1-par2 map time1-time2 day1-day2 [keywords]*

Parameters

param	parameter to contour; valid parameters are those defined in the **SFCCON.CORE**, **SFCCON.SITE** and **SFCCON.USER** files; see the Remarks for more information and a list of the parameters in **SFCCON.CORE** (default=T)
par1-par2	parameter difference to contour, e.g., T-TD for dewpoint depression; see the Remarks
map	map for the contour; see the MAP command for a list of valid predefined maps, for example, FL, MID, USA, WORL (default=do not draw a map; use the navigation of the frame specified with NAV keyword)
time	observation time; see the Remarks (default=current or most recent reporting hour)
time1-time2	time difference to contour, e.g., 12-6 to subtract 6 UTC data from 12 UTC data; see the Remarks
day	observation date; valid formats include YYYYDDD, YYYY/MM/DD, and DD/MON/YYYY where YYYY, DDD, MM, and DD are integers and MON is three or more letters, for example, JAN or JANUARY (default=current)
day1-day2	day difference to contour, e.g., 2000110-2000109 to subtract day 2000109 data from day 2000110 data; see the Remarks

---

Keywords

CHIlltype=	**CURRENT**	compute the wind chill values for the CHILL option using the formula implemented by the National Weather Service in November 2001 (default)
	OLD	compute the wind chill values using the pre-2001 formula
CINt=	*num*	contour interval; specify 0 to let SFCCON compute a reasonable interval; specify your own interval as an integer or decimal value greater than zero (default=from the **CONTOUR.DEF** text file; if no file entry matches the parameter and units, default=0)
	string	contour values; specify the name of a string that contains up to 64 values; e.g., CINT=SNOWDEPTH will draw contours at the values specified in the string named SNOWDEPTH
COLor=	graphics color level for the display (default=2)	
DASh=	**ALL**	dash all contours
	NEG	dash negative contours only
	POS	dash positive contours only
DATaset=	ADDE point dataset name and position; specify as *group/descriptor.position* or *alias.position* where *position* is a positive integer (absolute position), 0 or a negative integer (relative position), or ALL (default=RTPTSRC/SFCHOURLY.ALL)	
GRA=	graphic frame number for the display (default=current)	
LSIze=	contour label height, in pixels (default=8)	
NAV=	*num*	frame to use for navigation

	C	use the current frame's navigation (default)

you cannot use the NAV keyword if you specify a map with the *map* parameter

PTYpe=	**ALT**	use altimeter setting when computing station pressure for use in the equations that generate the contoured values if the PSTN, MIX, THA or THAE option is specified (default); see the Remarks
	PMSL	use sea-level pressure when computing station pressure for use in the equations that generate the contoured values if the PSTN, MIX, THA or THAE option is specified
UNIt=		units for the specified parameter; see the Remarks for defaults; run the CU LIST command to list the valid units and abbreviations for the parameters that can be changed

---

Remarks

SFCCON builds and runs a PTCON command to display surface hourly data from ISFC-schema point files. SFCCON accepts all the keywords in the PTCON command except LAT, LON, MAP, PRO, SLAT, SLON and PARAM because they may conflict with SFCCON's parameters or keywords. You can use the PTCON SELECT keyword to limit the analysis to a specified list or range of values, but you cannot specify a different parameter, time, day or unit than those defined by the SFCCON parameters or keywords.

The table below lists the parameter name, default units and a description of each parameter defined in the **SFCCON.CORE** file. Default units enclosed in parenthesis cannot be changed. The **.CORE** file is supplied with McIDAS-X and should not be modified. The **.USER** file overrides the **.SITE** file, which overrides the **.CORE** file. This order of precedence allows you or your site to change existing parameters and/or add new parameters in your **.USER** or **.SITE** file. To create a **.USER** or **.SITE** file, copy the **.CORE** file from **~mcidas/data** to your **$HOME/mcidas/data** directory (or another **MCPATH** directory), rename it, then edit the existing predefined parameters and/or add new ones.

param	Default Units	Description
ELEV	FT	station elevation
T	F	temperature
CHILL	(F)	wind chill index
HEAT	(F)	heat index
TD	F	dew point temperature
MIX	(G/KG)	mixing ratio
THA	(K)	potential temperature
THAE	(K)	equivalent potential temperature
DIR	(degrees)	wind direction
SPD	KTS	wind speed
GUS	KTS	wind gusts
STREAML		streamlines
ALTIM	MB	altimeter setting
PMSL	MB	sea-level pressure
PSTN	(MB)	station pressure
VIS	MI	visibility
PCP	IN	three- or six-hour precipitation total; three-hour total reported only at 3, 9, 15, 21 UTC; six-hour total reported only at 0, 6, 12, 18 UTC; see the Remarks
PCP3	IN	three-hour PCP total; reported only at 3, 9, 15, 21 UTC; see the Remarks
PCP6	IN	six-hour PCP total; reported only at 0, 6, 12, 18 UTC; see the Remarks
PCP24	IN	24-hour precipitation total; reported at 12 UTC; see the Remarks
SNO	IN	snow depth; reported only at 0, 6, 12, 18 UTC; see the Remarks
CAC	(code)	ceiling cloud amount; 0=clear, 5=few, 1=scattered, 2=broken, 3=overcast
CA1	(code)	first non-ceiling cloud amount; 0=clear, 5=few, 1=scattered, 2=broken, 3=overcast
CA2	(code)	second non-ceiling cloud amount; 0=clear, 5=few, 1=scattered, 2=broken, 3=overcast

CZC	FT	height of cloud ceiling
CZ1	FT	height of first non-ceiling cloud layer
CZ2	FT	height of second non-ceiling cloud layer
ABV	(sec^{-1})	absolute vorticity
VOR	(sec^{-1})	vorticity
DVG	(sec^{-1})	divergence

The *time* default is the current hour for all parameters except PCP, PCP3, PCP6, PCP24 and SNO. Because PCP and SNO are not reported every hour in METAR reports, the default for these parameters is the most recent hour storing the data. PCP is reported every three hours starting at 0 UTC. PCP3 represents the three-hour totals in PCP; it is reported at 3, 9, 15, and 21 UTC. PCP6 represents the six-hour totals in PCP; it is reported at 0, 6, 12, and 18 UTC. PCP24 represents the 24-hour precipitation totals and is reported at 12 UTC only. SNO is reported every six hours at 0, 6, 12, and 18 UTC.

This means, for example, that the *time* default for PCP is 0 if the current time is between 0:00 and 2:59 UTC; the *time* default for PCP3 is 3 if the current time is between 3:00 and 8:59 UTC; the *time* default for SNO and PCP6 is 6 if the current time is between 6:00 and 11:59 UTC. The time default for PCP24 is 12 UTC today if the current time is between 12:00 and 23:59 UTC, and is 12 UTC yesterday if the current time is between 0:00 and 11:59 UTC.

Parameter advection or divergence is done by appending ADV or DVG to the parameter name, e.g., TADV and SPDDVG. Parameters with their default units enclosed in parenthesis in the table above (e.g., CHILL, MIX and THA) and advected or diverged parameters cannot be used with the differencing options (*par1-par2*, *time1-time2* and *day1-day2*).

If you specify both a time difference (*time1-time2*) and a day difference (*day1-day2*), SFCCON contours the values equal to the parameter at *time1*,*day1* minus the parameter at *time2*,*day2*. If you specify a parameter difference (*par1-par2*) with a time difference and day difference, SFCCON contours the values equal to the parameter difference at *time1*,*day1* minus the parameter difference at *time2*,*day2*. For example, SFCCON T-TD MID 0-12 2000123-2000122 contours the values equal to the dewpoint depression at 0UTC,day2000123 minus the dewpoint depression at 12UTC,day2000122.

In the PTYPE keyword, specify either PMSL or SLP for sea-level pressure, and ALT, ALTM or ALTIM for altimeter setting. The keyword is valid only with the calculated options that contain station pressure in their equations (PSTN, MIX, THA and THAE). Altimeter setting is the default because it's correct to use it (rather than sea-level pressure) when computing station pressure for use in the equations.

Examples

SFCCON

This entry contours the temperature for the current day and time using the navigation of the current frame.

SFCCON PCP3 UNIT=MM

This entry contours the most recent three-hour precipitation totals in millimeters using the navigation of the current frame. For example, if the current time is 19 UTC, the 15 UTC data is plotted since three-hour precipitation totals are reported only at 3, 9, 15 and 21 UTC.

SFCCON SNO MID 12 UNIT=CM COLOR=7 LABEL='12 UTC Midwest snow depth, in centimeters' 7 420 150

This entry contours snow depth in centimeters at 12 UTC over a map of the Midwest. The contours are drawn in graphics color level 7. The default label below the plot is replaced with the text *12 UTC Midwest snow depth, in centimeters* near the center of the frame in graphics color level 7. LABEL is a PTCON keyword.

TE COLD "-20 -10 0 10 20 25 30 32 35

SFCCON T CINT=COLD DASH=NEG

The TE command creates a string called COLD containing contour values for the SFCCON entry. The SFCCON entry draws the -20, -10, 0, 10, 20, 25, 30, 32 and 35° Fahrenheit temperature contours for the current day and time using the navigation of the current frame. The -20° and -10° contours are dashed; the others are solid. Do not specify a pound sign (#) before the string name in the CINT keyword.

SFCCON PMSL NA 6 12/JAN/1996 DATASET=CLIMATE/1996

This entry contours sea-level pressure over a map of North America at 6 UTC on 12 January 1996 using the data in the dataset CLIMATE/1996.

SFCCON TDADV KY

This entry contours dewpoint temperature advection over a map of Kentucky.

SFCCON VIS MID 12-0 2000131-2000130

This entry contours the 36-hour visibility change from 0 UTC, day 2000130 to 12 UTC, day 2000131 over a map of the Midwest.

SFCCON THA USA PTYPE=PMSL

This entry calculates and contours potential temperature over a map of the United States for the current day and hour. Mean sea-level pressure is used when computing station pressure for use in the potential temperature calculation. This may yield some error in the values, with higher station elevations usually having larger errors.

SFCLIST

Lists surface hourly data from point files.

Format

SFCLIST *id1 . . idn nhours [keywords]*

Parameters

id1 . . idn	station IDs or states to list; use the two-character state P.O. abbreviations for states; specify either IDs or states, but not a combination; see the Remarks
nhours	number of hours to search for observations starting from the current time and going backwards; use the TIME keyword to specify a range of times; see the Remarks (default=1)

Search Keywords

CO= *co1 . . con* lists observations for all stations in these countries; use CCODE command to list valid country codes; see the Remarks

DATaset= ADDE point dataset name and position; specify as *group/descriptor.position* or *alias.position* where *position* is a positive integer (absolute position), 0 or a negative integer (relative position), or ALL (default=RTPTSRC/SFCHOURLY.ALL)

DAY= *bday eday* range of days to list; uses the TIME keyword, not *nhours* parameter to list times (default=current *bday*)

SELect= '*clause1*; *clause2*; . . ; *clausen*'
finds only the data matching the values in the select clauses; single quotes are mandatory; separate select clauses with semicolons; specify *clause* as a range or a list using one of the following formats:

range: *parameter* **[***unit***]** *bval eval*
list: *parameter* **[***unit***]** *val1,val2, . . ,valn*

parameter	point file parameter; you cannot use ID, ST, CO, TIME or DAY in a clause
[*unit***]**	units for the parameter values; units are optional but must be in square brackets if specified (default=from point dataset)
bval eval	range of parameter values
val1, . ., valn	list of parameter values

SPAn= **TIME** lists all data in the TIME keyword range, *btime* to *etime*, for each day of the DAY keyword range, *bday* to *eday* (default)

DAY lists all data from *btime* on *bday* through *etime* on *eday*

TIMe= *btime etime* range of times to list; overrides *nhours* parameter (default=current *btime*)

Output Keywords

OPTion= *o1 . . on*

optional output fields to display; choose from the following options

NONE doesn't display any optional fields (default)

	ALL	displays all optional fields listed below
	CHILL	displays the derived wind chill index, WCI
	CLOUD	displays the first and second non-ceiling cloud parameters, Cld1 and Cld2
	HEAT	displays the derived heat index, HI
	HUMID	displays the relative humidity, RH, and mixing ratio, MIX
	PRECIP	displays liquid precipitation, Pcp, snow depth, Sno, and 24-hour precipitation, P24
	SLP	displays the sea-level pressure, SLP
	STP	displays derived station pressure, STP
	THETA	displays derived potential temperature, Th, equivalent potential temperature, ThE, and wetbulb potential temperature, ThW; displayed in Kelvin regardless of value specified with the UNIT keyword
	VAPOR	displays derived vapor pressure, e, and saturation vapor pressure, es
	VIRTUAL	displays derived virtual temperature, Tv
	WETBULB	displays derived lifted condensation level, LCL, and wet bulb temperature, Tw
CHIlltype=	**CURRENT**	compute the wind chill values listed with OPT=CHILL using the formula implemented by the National Weather Service in November 2001 (default)
	OLD	compute the wind chill values using the pre-2001 formula
PTYpe=	**ALT**	use altimeter setting in computations of derived parameters listed in OPT=HUMID, STP, THETA, VIRTUAL and WETBULB options; see the Remarks
	PMSL	use sea-level pressure in computations of derived parameters listed in OPT=HUMID, STP, THETA, VIRTUAL and WETBULB options (default)
SORt=	**TIME**	sorts output by nominal hour, HH; for each hour, stations are listed alphabetically, regardless of the actual reporting time (default)
	OBTIME	sorts output by the exact time of observation, HHMM; for each time, stations are listed alphabetically
	STN	sorts output alphabetically by station; for each station, observations are listed chronologically
UNIt=	**A**	lists output in American units: temperatures in Fahrenheit, wind speed and gusts in knots, visibility in miles, cloud heights in 100s of feet, precipitation and snow depth in inches (default)
	M	lists output in metric units: temperatures in Celsius, wind speed and gusts in meters/second, visibility in kilometers, cloud heights in hectometers, precipitation and snow depth in millimeters

---

Remarks

SFCLIST displays point file data in an easy-to-read format that includes columns for temperature, dewpoint temperature, wind direction and speeds, pressure, visibility, current weather, and cloud ceiling. Special (i.e., nonhourly) observations are indicated by an *S* in the first column of the output. This data is obtained from the same raw observational text reports displayed by the command SFCRPT, but formatted in columns.

The cloud ceiling (Ceil) field on the right side of the output is displayed as *coverage/height*. Coverage is listed in eighths; height is listed in hundreds of feet if UNIT=A, or in hectometers if UNIT=M. For example, 3/111 represents 3/8ths cloud coverage at a height of 11,100 feet if UNIT=A, or 11,100 meters if UNIT=M. If OPTION=CLOUD or ALL is specified, the coverage and height values for the first and second non-ceiling levels (Cld1 and Cld2) are also listed.

You must specify at least one station ID, state, or country. Normally, you should not specify a combination of station IDs, states or countries. Because SFCLIST uses a logical AND between different search conditions, only data that satisfies all of the specified conditions is listed. For example, the command SFCLIST KBOS FL will not find any data because Boston (KBOS) is not in the state of Florida; SFCLIST NY CO=AU will not find any data because there is no Australian state with the abbreviation NY. If you specify multiple values for a single search condition

however, SFCLIST uses a logical OR to list all data. For example, the command SFCLIST KBOS KMIA lists all data for both Boston and Miami.

When using the *nhours* parameter, the current time determines which data is listed. For example, if you run the command SFCLIST KMSN 1 between 11:00 and 11:59 UTC, Madison's 11 UTC data is listed. If you run it between 12:00 and 12:59 UTC, the 12 UTC data is listed. If the current time is before 13 UTC (12:55, for example) but the 13 UTC data is available, you cannot list it using *nhours*; you must specify TIME=13.

In the PTYPE keyword, specify either PMSL or SLP for sea-level pressure pressure, and ALT, ALTM or ALTIM for altimeter setting. The keyword is valid only with the calculated options that contain a pressure value in their equations (OPT=HUMID, STP, THETA, VIRTUAL and WETBULB). Sea-level pressure is the default because it's the correct pressure type for those equations. However, since altimeter setting is usually much more common (than sea-level pressure) in surface hourly reports, you can use it by specifying PTYPE=ALT. But if you do so, be aware that the values of the calculated parameters won't be correct. The amount of error depends on the equation (e.g., mixing ratio values are usually about the same whether using sea-level pressure or altimeter setting) and the station elevation (higher elevations usually result in larger differences).

See Also

See the PTLIST command for additional information about select clauses.

Examples

SFCLIST TX

This entry lists all surface data for Texas from the past hour.

SFCLIST TX OK KS 2

This entry lists all surface data for Texas, Oklahoma, and Kansas from the past two hours.

SFCLIST KMIA KLAX TIME=6 12

This entry lists all surface data for Miami and Los Angeles from 6 to 12 UTC today.

SFCLIST KMSN TIME=0 12 DAY=96100 96105 SPAN=DAY DATASET=CLIMATE/1996

This entry lists all archived surface data for Madison from 0 UTC on day 96100 to 12 UTC on day 96105.

SFCLIST KMSN TIME=0 12 DAY=96100 96105 DATASET=CLIMATE/1996

This entry lists all archived surface data for Madison from 0 UTC to 12 UTC on each day in the range 96100 through 96105.

SFCLIST X 6 CO=IT ES FR

This entry lists all surface data for Italy, Spain, and France from the past six hours.

SFCLIST MS 3 CO=US

This entry is an exception to the rule stated in the Remarks above. It lists all surface data for Mississippi from the past three hours. The CO=US keyword eliminates data from stations in other countries that use MS as the state/province/territory abbreviation. It is rarely necessary to specify the CO keyword when listing data for a U.S. state because very few stations outside the United States use an abbreviation that matches a U.S. state abbreviation.

SFCLIST CO=US SELECT='T[F] -100 -5 ; SPD[KTS] 20 40' OPTION=CHILL

This entry lists surface data, including the wind chill index, for all stations in the United States reporting a temperature below -5°F and wind speed between 20 and 40 knots from the past hour.

SFCMG

Displays a surface meteorogram.

Format

SFCMG *id nhours [keywords]*

Parameters

id station ID

nhours number of hours to include in meteorogram; specify the ending time with the TIME keyword (default=24)

Keywords

COLor= *t td p cld win wx pcp vis bg*

graphics color levels for each meteorogram component: temperature (*t*), dew point (*td*), pressure (*p*), clouds (*cld*), wind (*win*), current weather (*wx*), precipitation (*pcp*), visibility (*vis*), background grid (*bg*) (default=5 4 2 7 3 3 4 3 8)

CRAnge= *min max unit* minimum and maximum cloud levels, and units (*min* default=0; *max* default=maximum cloud height; *unit* default=FT)

DATaset= ADDE point dataset name and position; specify as *group/descriptor.position* or *alias.position* where *position* is a positive integer (absolute position), 0 or a negative integer (relative position), or ALL (default=RTPTSRC/SFCHOURLY.ALL)

DAY= ending day to plot (default=current)

EG=
- **NO** does not erase graphics frame before plotting
- **YES** automatically erases graphis frame before plotting (default)

GRA= graphic frame number to display meteorogram (default=current)

INT= hour interval for plotting data; see the Remarks (default=1)

LATest=
- **LEFT** plot the most recent data on the left side of the meteorogram
- **RIGHT** plot the most recent data on the right side of the meteorogram (default)

PRAnge= *min max unit* minimum and maximum pressure, and units (*min* and *max* default=calculated; *unit* default=MB)

PTYpe=
- **ALT** use altimeter setting for pressure values (default)
- **PMSL** use sea-level pressure for pressure values

TDAy=
- **CAL** display the calendar day in the title and day labels (default)
- **JUL** display the Julian day in the title and day labels

TIMe= ending hour to plot (default=current for current day; default=23 if another day is specified with the DAY keyword)

TRAnge= *min max unit* minimum and maximum temperature, and units (*min* and *max* default= calculated; *unit* default=F)

TYPe=
- **ALL** plot both hourly and special observations (default=plot only hourly observations)

UNIt=
- **A** plot in American units: visibility in miles and precipitation in inches (default)

M	plot in metric units: visibility in kilometers and precipitation in millimeters

---

Remarks

SFCMG displays a meteorogram (time-series plot) of surface hourly data. The meteorogram displays temperature, dew point and pressure traces. It also plots cloud coverage and heights; wind direction, speed, and gusts; current weather; precipitation; and visibility.

To avoid plotting data that overlaps the meteorogram's borders and legends, SFCMG displays an x-axis longer than the plotted data. The fact that the data's starting and ending times fall inside the diagram's borders does not mean data is missing.

SFCMG plots the first weather reported in the weather group. For example, if the weather group consists of heavy snow and blowing snow (+SN, BLSN), only the heavy snow symbol is plotted.

Use the INT keyword to declutter the plotted data. For example, the command below plots every sixth hour of data from 00 UTC on day 97097 through 00 UTC on day 97100.

SFCMG KDCA 72 DAY=97100 TIME=18 INT=6

Without INT=6, every hour's data would be displayed and the wind flags and gusts, weather symbols, precipitation totals, and visibility values would overlap.

Examples

SFCMG KMSN

This entry displays a meteorogram showing the surface data for Madison from the past 24 hours.

SFCMG EGLL 48 UNIT=M TRANGE=0 20 C

This entry displays a meteorogram showing the surface data for London from the past 48 hours. The grid containing the temperature trace ranges from 0° C to 20° C. Visibility and precipitation are plotted in metric units (kilometers and millimeters).

SFCMG KMIA 36 TIME=12 DAY=96105 DATASET=CLIMATE/1996

This entry displays a meteorogram showing 36 hours of surface data for Miami from 00 UTC on day 96104 through 12 UTC on day 96105.

SFCPLOT

Plots surface hourly data from ISFC-schema point files.

Formats

SFCPLOT *param map time day [keywords]*
SFCPLOT *par1-par2 map time1-time2 day1-day2 [keywords]*

Parameters

param	parameter to plot; valid parameters are those defined in the **SFCPLOT.CORE**, **SFCPLOT.SITE** and **SFCPLOT.USER** files; see the Remarks for more information and a list of the parameters in **SFCPLOT.CORE** (default=PLOT)
par1-par2	parameter difference to plot, e.g., T-TD for dewpoint depression; see the Remarks
map	map for the plot; see the MAP command for a list of valid predefined maps, for example, FL, MID, USA, WORL (default=do not draw a map; use the navigation of the frame specified with NAV keyword)
time	observation time; see the Remarks (default=current or most recent reporting hour)
time1-time2	time difference to plot, e.g., 12-6 to subtract 6 UTC data from 12 UTC data; see the Remarks
day	observation date; valid formats include YYYYDDD, YYYY/MM/DD, and DD/MON/YYYY where YYYY, DDD, MM, and DD are integers and MON is three or more letters, for example, JAN or JANUARY (default=current)
day1-day2	day difference to plot, e.g., 2000110-2000109 to subtract day 2000109 data from day 2000110 data; see the Remarks

---

Keywords

CHIlltype=	**CURRENT**	compute the wind chill values for the CHILL option using the formula implemented by the National Weather Service in November 2001 (default)
	OLD	compute the wind chill values using the pre-2001 formula
COLor=	graphics color level (default=3); specify as a single number, e.g., COLOR=5, or in format *color*[*min-max*B*inc*] to assign multiple colors for the parameter, based on its values; e.g., COLOR=2[30-59B10] plots values 30-39 in level 2, values 40-49 in level 3, values 50-59 in level 4, values less than 30 in level 1, and values greater than 59 in level 5	
DATaset=	ADDE point dataset name and position; specify as *group*/*descriptor*.*position* or *alias*.*position* where *position* is a positive integer (absolute position), 0 or a negative integer (relative position), or ALL (default=RTPTSRC/SFCHOURLY.ALL)	
DEClut=	**YES**	declutter the plot, meaning do not plot overlapping values (default)
	NO	do not declutter the plot
GRA=	graphic frame number for the display (default=current)	
LSIze=	height of the plot characters, in pixels (default=8)	
NAV=	*num*	frame to use for navigation
	C	use the current frame's navigation (default)
	you cannot use the NAV keyword if you specify a map with the *map* parameter	
PTYpe=	**ALT**	plot altimeter setting for pressure values if the PLOT option is specified, or use altimeter setting when computing station pressure for use in the equations that generate the plotted values if the PSTN, MIX, THA or THAE option is

specified (default); see the Remarks

PMSL plot sea-level pressure for pressure values if the PLOT option is specified, or use sea-level pressure when computing station pressure for use in the equations that generate the plotted values if the PSTN, MIX, THA or THAE option is specified

UNIt= units for the specified parameter; see the Remarks for defaults; run the CU LIST command to list the valid units and abbreviations for the parameters that can be changed

---

Remarks

SFCPLOT builds and runs a PTDISP command to display surface hourly data from ISFC-schema point files. SFCPLOT accepts all the keywords in the PTDISP command except LAT, LON, MAP, PRO, SLAT, SLON and PARAM because they may conflict with SFCPLOT's parameters or keywords. You can use the PTDISP SELECT keyword to limit the plot to a specified list or range of values, but you cannot specify a different parameter, time, day or unit than those defined by the SFCPLOT parameters or keywords.

The table below lists the parameter name, default units and a description of each parameter defined in the **SFCPLOT.CORE** file. Default units enclosed in parenthesis cannot be changed. The **.CORE** file is supplied with McIDAS-X and should not be modified. The **.USER** file overrides the **.SITE** file, which overrides the **.CORE** file. This order of precedence allows you or your site to change existing parameters and/or add new parameters in your **.USER** or **.SITE** file. To create a **.USER** or **.SITE** file, copy the **.CORE** file from **~mcidas/data** to your **$HOME/mcidas/data** directory (or another **MCPATH** directory), rename it, then edit the existing predefined parameters and/or add new ones.

param	**Default Units**	**Description**
ID	(text)	three- or four-character ICAO station ID
OBS	(dots)	observation locations
ELEV	FT	station elevation
OBTIME	(HMS)	actual time of observation
PLOT	A	station plot of the T, ALTIM or PMSL (set with the PTYPE keyword), CAC, ID, TD, WX1, WINDB parameters, in clockwise order from top left of plot; units are A, for American, or M, for metric
WX1	(text)	weather type (first four characters)
WX2	(text)	weather type (next four characters)
WX	(text)	weather type (combined WX1, WX2)
WXS	(symbols)	weather type (combined WX1, WX2)
WXC	(symbols)	WMO codes for present weather (combined WXC1, WXC2, WXC3, WXC4)
T	F	temperature
CHILL	(F)	wind chill index
HEAT	(F)	heat index
TD	F	dew point temperature
MIX	(G/KG)	mixing ratio
RH	(%)	relative humidity
THA	(K)	potential temperature
THAE	(K)	equivalent potential temperature
DIR	(degrees)	wind direction
SPD	KTS	wind speed
GUS	KTS	wind gusts
WIND	(deg) KTS	two-digit wind direction and two- or three-digit speed, in the form DDSSS
WINDB	(barb)	wind barb (DIR and SPD); flag=25 mps (~50 kts), long barb=5 mps (~10 kts), short barb=2.5 mps (~5 kts)
WINDV	(vector)	wind vector (DIR and SPD)
ALTIM	MB	altimeter setting
PMSL	MB	sea-level pressure
PSTN	(MB)	station pressure
VIS	MI	visibility

PCP	IN	three- or six-hour precipitation total; three-hour total reported only at 3, 9, 15, 21 UTC; six-hour total reported only at 0, 6, 12, 18 UTC; see the Remarks
PCP3	IN	three-hour PCP total; reported only at 3, 9, 15, 21 UTC; see the Remarks
PCP6	IN	six-hour PCP total; reported only at 0, 6, 12, 18 UTC; see the Remarks
PCP24	IN	24-hour precipitation total; reported at 12 UTC; see the Remarks
SNO	IN	snow depth; reported only at 0, 6, 12, 18 UTC; see the Remarks
CAC	(code)	ceiling cloud amount; 0=clear, 5=few, 1=scattered, 2=broken, 3=overcast
CA1	(code)	first non-ceiling cloud amount; 0=clear, 5=few, 1=scattered, 2=broken, 3=overcast
CA2	(code)	second non-ceiling cloud amount; 0=clear, 5=few, 1=scattered, 2=broken, 3=overcast
SKY	(code)	sky cloud amount (combined CAC, CA1, CA2)
CZC	FT	height of cloud ceiling
CZ1	FT	height of first non-ceiling cloud layer
CZ2	FT	height of second non-ceiling cloud layer

The *time* default is the current hour for all parameters except PCP, PCP3, PCP6, PCP24 and SNO. Because PCP and SNO are not reported every hour in METAR reports, the default for these parameters is the most recent hour storing the data. PCP is reported every three hours starting at 0 UTC. PCP3 represents the three-hour totals in PCP; it is reported at 3, 9, 15, and 21 UTC. PCP6 represents the six-hour totals in PCP; it is reported at 0, 6, 12, and 18 UTC. PCP24 represents the 24-hour precipitation totals and is reported at 12 UTC only. SNO is reported every six hours at 0, 6, 12, and 18 UTC.

This means, for example, that the *time* default for PCP is 0 if the current time is between 0:00 and 2:59 UTC; the *time* default for PCP3 is 3 if the current time is between 3:00 and 8:59 UTC; the *time* default for SNO and PCP6 is 6 if the current time is between 6:00 and 11:59 UTC. The time default for PCP24 is 12 UTC today if the current time is between 12:00 and 23:59 UTC, and is 12 UTC yesterday if the current time is between 0:00 and 11:59 UTC.

If you specify both a time difference (*time1-time2*) and a day difference (*day1-day2*), SFCPLOT plots the values equal to the parameter at *time1*,*day1* minus the parameter at *time2*,*day2*. If you specify a parameter difference (*par1-par2*) with a time difference and day difference, SFCPLOT plots the values equal to the parameter difference at *time1*,*day1* minus the parameter difference at *time2*,*day2*. For example, SFCPLOT T-TD MID 0-12 2000123-2000122 plots the values equal to the dewpoint depression at 0UTC, day 2000123 minus the dewpoint depression at 12UTC, day 2000122.

Parameters with their default units enclosed in parenthesis in the table above (e.g., ID, CHILL and THA) cannot be used with the differencing options (*par1-par2*, *time1-time2* and *day1-day2*).

In the PTYPE keyword, specify either PMSL or SLP for sea-level pressure, and ALT, ALTM or ALTIM for altimeter setting. The default is ALT for the PLOT option and the calculated options that contain station pressure in their equations (PSTN, MIX, THA and THAE). Altimeter setting is the default in the calculated options because it's correct to use it (rather than sea-level pressure) when computing station pressure for use in the equations.

Examples

SFCPLOT

This entry plots station model data for the current day and time using the navigation of the current frame.

SFCPLOT T MID DECLUT=NO

This entry plots temperature over a map of the Midwest for the current day and time. The data is not decluttered, so all data is plotted and some values may overlap.

SFCPLOT PCP6 UNIT=MM

This entry plots the most recent six-hour precipitation totals in millimeters using the navigation of the current frame. For example, if the current time is 17 UTC, the 12 UTC data is plotted since six-hour precipitation totals are reported only at 0, 6, 12 and 18 UTC.

SFCPLOT WINDB USA 22 COLOR=7 LSIZE=12 SELECT='SPD[KTS] 25 200' DECLUT=NO TITLE='22 UTC winds of at least 25 knots' 7 420 150

This entry plots wind barbs for all 22 UTC wind speeds of 25 knots or greater over the United States. The barbs are drawn in graphics color level 7 with a plot size of 12 pixels. The default title below the plot is replaced with the text *22 UTC winds of at least 25 knots* near the center of the frame in graphics color level 7. SELECT and TITLE are PTDISP keywords.

SFCPLOT SNO NA 12 17/JAN/1996 DATASET=CLIMATE/1996 DECLUT=NO

This entry plots all snow depth reports over a map of North America at 12 UTC on 17 January 1996 using the data in the dataset CLIMATE/1996. The data is not decluttered, so all data is plotted and some values may overlap.

SFCPLOT T OK 18 2000130-2000129

This entry plots the 24-hour temperature change in degrees Celsius from 18 UTC, day 2000129 to 18 UTC, day 2000130 over a map of Oklahoma.

SFCPLOT MIX FL

This entry plots mixing ratio over a map of Florida for the current day and time. It uses the default for the PTYPE keyword, meaning that altimeter setting is used when computing station pressure for use in the mixing ratio calculation.

SFCPLOT PSTN CO 12 PTYPE=PMSL

This entry plots station pressure over a map of Colorado at 12 UTC of the current day. Mean sea-level pressure is used instead of altimeter setting when calculating the station pressure values. This may yield some error in the values, with higher station elevations usually having larger errors.

SFCRPT

Lists surface hourly observations.

Format

SFCRPT *id1 . . idn nhours [keywords]*

Parameters

id1 . . idn	station IDs or states to list; use the two-character state P.O. abbreviation for states
nhours	number of hours to search for observations starting from DAY and TIME (default=1)

Keywords

CO=	*co1 . . con*	lists observations for all stations in these countries; use the CCODE command to list valid country codes
DATaset=	ADDE dataset name; specify as *group/descriptor* or *alias* (default=RTWXTEXT/SFCHOURLY)	
DAY=	Julian date to start search; use either the 5- or 7-digit format, for example, 96300 or 1996300 (default=current)	
TIMe=	UTC time to start search (default=current)	
TYPe=	**ALL**	lists both hourly and special reports (default)
	H	lists only hourly reports
	S	lists only special reports

Remarks

You must specify at least one station ID, state, or country.

If you use the DAY keyword to specify a day other than the current day and do not use the TIME keyword, the search begins at the current hour on the specified day and works backwards.

Examples

SFCRPT KSTL

This entry lists all surface observations for St. Louis from the past hour.

SFCRPT KMIA KBTV VT 2

This entry lists all surface observations for Miami; Burlington, Vermont; and all stations in Vermont from the past two hours. Burlington's observations are listed only once.

SFCRPT KGRB 4 CO=BM BZ

This entry lists all surface observations for Green Bay, Wisconsin; Bermuda and Belize from the past four hours.

SFCRPT KZZV 13 TIME=4 TYPE=S

This entry lists only special surface observations for Zanesville, Ohio. SFCRPT lists observations starting with 4:00 UTC of the current day and ending with 16:00 UTC of the previous day.

SFCRPT NZCH 24 TIME=23 DAY=96300

This entry lists all surface observations for Christchurch, New Zealand, on day 96300.

SG

Displays a graphics frame.

Format

SG *frame*

Parameter

frame graphics frame number to display (default=1)

Remarks

Use the SG command only on McIDAS-X sessions with independent graphics. See the TERM command for more information.

See Also

Use the F command to list the current image and graphic frame numbers.

Examples

SG

This entry displays graphics frame 1.

SG 4

This entry displays graphics frame 4.

SHOWVG

Displays virtual graphics.

Format

SHOWVG *virt bframe eframe [keywords]*

Parameters

virt	virtual graphics number; the range is 1 to 9999
bframe	beginning graphics frame number (default=current)
eframe	ending graphics frame number (default=*bframe*)

Keywords

ELE=	offsets the graphic by the specified number of elements (default=0)
LINe=	offsets the graphic by the specified number of lines (default=0)

Remarks

The SHOWVG command restores a virtual graphic saved on the workstation. You create virtual graphics by using the global keyword VIRT with any program that creates graphical output. See *Appendix A, Global Keywords*, for more information.

Use the command DMAP VIRT to list all virtual graphics files saved on the workstation. Virtual graphics files are named VIRT*nnnn* where *nnnn* is the virtual graphics number.

Use the keywords to reposition the graphic. The specified values are constants added to the line and element coordinates when displaying the graphic.

Examples

SHOWVG 1000

This entry displays virtual graphic 1000 on the current frame.

SHOWVG 1000 1 2

This entry displays virtual graphic 1000 on frames 1 and 2.

SHOWVG 1000 1 2 LINE=80 ELE=100

This entry is the same as the one above except it displays the virtual graphic 80 lines lower and 100 elements to the right of the original position.

SKE

Enters a McIDAS command in the system command scheduler.

Format

SKE *day time repeat interval [keywords] "command*

Parameters

day	day to initiate the command; valid date formats include YYYYDDD, YYYY/MM/DD, and DD/MON/YYYY where YYYY, DDD, MM, and DD are integers and MON is three or more letters, for example, JAN or JANUARY (default=current)
time	time to initiate the command, HH:MM:SS; use colons when specifying the time (default=00:00:00)
repeat	number of times to run the command; if you enter 999999, the word MANY replaces it in the header and the command is run until you suspend or delete it from the scheduler using the SKU command (default=1)
interval	time interval between commands, DDDHH:MM:SS; use colons when specifying the time; for one day, enter 24:00:00 or 100:00:00 (default=1 hour)
"command	command to run; the command length may be up to 160 characters (including spaces)

Keywords

FILE=	scheduler file name; see the Remarks (default=SKEDFILE)
ID=	number to identify the command in the scheduler; the range is 1 to 9999 (default=system assigned number)
NAME=	user name assigned to the scheduler entry (default=name specified at logon)
PROJ=	project number charged when you run the command (default=project number specified at logon)
TOL=	late tolerance, HH:MM:SS; if the workstation is down or too busy for the command to run, the late tolerance determines how much time can pass before the command is skipped; if 10 to 24 hours are specified, the word BIG appears in the header; three seconds (:00:03) is the minimum acceptable time (default=1 hour)

Remarks

The first line of output from this command indicates if the scheduler is on or off. Scheduled commands will run only if the scheduler is on. Use the SKU command to turn the scheduler on or off.

You can run multiple scheduler processes. Use the FILE keyword to identify the file name for the scheduler process. See the SKED command for more information.

To run a command stored in the workstation string table, precede the string name with two pound (##) signs. That is, enter "##string in place of "command. For example:

TE MAP "MAP MID

SKE 97300 12 "##MAP

Examples

SKE 97240 13:30 2 24:00 ID=3003 "IMGLIST GOES/VIS.ALL

This entry runs the command IMGLIST GOES/VIS.ALL at 13:30 UTC on each of two consecutive days (97240, 97241). The assigned ID for

reference is 3003.

SKE #Y 09:05 999999 3 PROJ=7777 "SFCLIST KSTL OPTION=PRECIP

This entry runs command SFCLIST KSTL OPTION=PRECIP every three hours every day beginning with the current day at 09:05 UTC.

SKE 98001 0 999999 1 "RUN "PTABLE MID$(TIME$,1,2), "HOUR"

This entry runs the specified RUN command every hour, on the hour, beginning with day 98001 at 00:00 UTC. The RUN command sets the value of the string HOUR to the current hour for use in commands like PTDISP and PTLIST.

SKED

Starts a scheduler process.

Format

SKED *filename [keyword]*

Parameter

filename scheduler file name (default=SKEDFILE)

Keyword

MAXent= maximum number of scheduler entries that can be run simultaneously (default=4)

Remarks

SKED begins a scheduler process that runs until you exit the McIDAS-X session. Use the ? (Question Mark) command to list all running processes.

At start up, the default **.mcidasrc** file starts a scheduler process using the default scheduler file, SKEDFILE. If you run multiple McIDAS-X sessions under the same account or using the same **MCPATH** setting, each session should run a scheduler process using a different file name. Otherwise, you might have scheduler entries from one session running on the other sessions.

If you start a scheduler process using a file other than SKEDFILE, you must use the keyword FILE with commands SKE, SKL and SKU since those commands also use SKEDFILE as their default scheduler file.

If you do not want the scheduler to automatically start, comment out the command **-c 'SKED SKEDFILE'** in your **.mcidasrc** file, or use the question mark (?) command to list the processes, then use the KILL command to stop the SKED process.

Examples

SKED

This entry starts a scheduler process using the default file SKEDFILE.

SKED MYSKED

This entry starts a scheduler process using the file MYSKED.

SKL

Lists command scheduler entries.

Format

SKL *bid eid [keywords] "matchstring*

Parameters

bid	beginning ID number to list; the range is 1 to 9999 (default=lists all entries)
eid	ending ID number to list (default=*bid*)
"*matchstring*	lists only the entries containing this string; maximum of 40 characters (default=no match performed)

Keywords

FILE=		scheduler file name; see the Remarks (default=SKEDFILE)
FORM=	**ALL**	uses as many lines of output as needed to list the entire command
PROJ=	*min max*	lists entries with a specified project number or range of numbers (default=1000 9999)
SORT=	**ID**	sorts entries by ID number (default)
	PROJ	sorts entries by project number
	RAW	lists in the order of occurrence in the schedule (unsorted)
	TIME	sorts entries in ascending time order

Remarks

You can run multiple scheduler processes. Use the FILE keyword to identify the file name for the scheduler process. See the SKED command for more information.

All entries are listed if the beginning and ending ID numbers are omitted. The output is defined below.

Column	Output	Definition
1	T	terminal number
2	ID	command identification number
3	X/S	S suspends the scheduled command; X disables the terminal's schedule
4	Next Executn	next day and time to run a command
5	#REM	number of remaining commands; if SKE repetitions is a 6-digit number, this will say MANY
6	Interval	day and time interval between executions; the format is DDHH:MM:SS, entered by SKE *interval* where *interval* is one hour; if *interval* is 24, this output is listed as 1000000 (1 day)
7	TOL	late tolerance time, HH:MM:SS; any entry over nine hours is listed as BIG
8	NAME:	user name
9	PROJ:	project number

10	Command:	command string to run

Examples

SKL

This entry lists all the scheduler entries for your terminal.

SKL 420 430 PROJ=7000

This entry lists the scheduler entries between 420 and 430 with project number 7000.

SKU

Command scheduler utility.

Formats

SKU *option bid eid [keywords]*
SKU MSG *device* **FILE=**
SKU CHA *bid eid [keywords] "command*
SKU DOIT *bid eid* **FILE=**
SKU ON FILE=
SKU OFF FILE=

Parameters

option	**DEL**	deletes the entry from the scheduler
	REL	releases the entry so the command runs at the next scheduled time
	SUS	suspends the entry so the command will not run, but does not delete the entry from the scheduler
MSG	device specified for the scheduler text output	
CHA	changes the entry	
DOIT	runs a schedule entry without decrementing the counters	
ON	turns the scheduler on	
OFF	turns the scheduler off	
bid	beginning ID number	
eid	ending ID number (default=*bid*)	
device	**C**	sends output to the text frame
	N	does not list the output
"command	new command to replace the current command; the new command may contain up to 160 characters including spaces	

Keywords

FILE=	scheduler file name; see the Remarks (default=SKEDFILE)	
NAME=	*old new*	changes the old user ID to the new user ID
PROJ=	*old new*	replaces the old project number with the new project number

---

Remarks

You can run multiple scheduler processes. Use the FILE keyword to identify the file name for the scheduler process. See the SKED command for more information.

To change a scheduler entry, you must be logged on to the project number listed in the entry.

Use the DOIT parameter to test scheduler entries.

Examples

SKU DEL 6

This entry deletes the scheduler entry with ID number 6.

SKU SUS 1001 1009

This entry suspends scheduler entries 1001 through 1009.

SKU CHA 1001 1009 PROJ=7000 5630

This entry changes the project number for scheduler entries 1001 through 1009 from 7000 to 5630.

SKU DOIT 5

This entry runs the command in scheduler entry 5. It doesn't change the run time or decrease the counter.

STNLIST

Lists weather stations.

Format

STNLIST *id1 . . idN [keywords]*

Parameters

id1 . . idN 3- to 5-character IDs or 5-digit ID numbers of stations to list

---

Search Keywords

CO=		list stations in this country; use the CCODE command to list valid 2-letter country codes
LAT=	*slat nlat*	list stations in this latitude range, where *slat* and *nlat* are the southern and northern latitude limits, respectively
LON=	*elon wlon*	list stations in this longitude range, where *elon* and *wlon* are the eastern and western longitude limits, respectively
MATch=	*'string'*	list stations with an exact match of the specified string, including any spaces
ST=		list stations in this state; use its 2-letter state P.O. abbreviation
USERid=		list stations matching the USERID field in the **STNDB.USER** and **STNDB.SITE** files; see the Remarks

---

Output Keywords

FORm=	**ALL**	list default information plus the full station name (default=list standard 1-line summary for each station)
	USER	list all station information in the format of the **STNDB.*** files; use this option with DEV=T to create a template for your **STNDB.USER** or **STNDB.SITE** file; see the Remarks
NUM=		number of stations matching the criteria defined with the other keywords to list; e.g., NUM=50 lists the first 50 matching stations (default=ALL, meaning list all matching stations)
TYPe=	*type1 . . typeN*	station types to list; valid entries are the following single letters or their multi-letter versions: **M** or **METAR**, **A** or **SAO**, **S** or **SYNOPTIC**, **R** or **RAOB**, **T** or **TAF** or **TERMFCST**, **W** or **WSFO**, **N** or **NEXRAD**, **C** or **CMAN**, **V** or **VOR**, **E** or **NAMMOS** or **ETAMOS**, **F** or **NGMMOS** or **FOUS14**, **G** or **GFSMOS**; see the Remarks for definitions of each of the types (default=ALL, meaning list all station types)

Interactive and Frame Navigation Keywords

MODe=	**N**	non-interactive "batch" mode; uses the specified search keywords and/or the cursor's position and size at the time the

		command is run to determine the stations to list (default)
	I	interactive mode; you can select multiple regions using the right mouse button or **Alt-G**, then press the center and right buttons or **Alt-Q** to quit
RADius=	*radius*[*unit*]	lists the stations within the radius distance from the cursor position; run CU LIST to list valid units and abbreviations; not valid with the SIZE keyword (no default for *radius*; default=KM for *unit*)
SIZe=	*line ele*	list stations within the region defined by the specified number of lines and elements centered at the cursor position; not valid with the RADIUS keyword (no default if *id1..idN*, CO, LAT, LON, MATCH, ST or USERID is specified and MODE=I is not specified; otherwise default=cursor size)

Dataset Search Keywords

DATaset=	point dataset name and position to search for stations; specify as *group/descriptor.position* or *alias.position* where *position* is a positive integer (absolute position), 0 or a negative integer (relative position), or ALL (no default for *group/descriptor* or *alias*; default=ALL for *position*)
DAY=	list only the stations in the specified DATASET that reported data at the specified DAY and TIME; valid formats include YYYYDDD, YYYY/MM/DD and DD/MON/YYYY, where YYYY, DDD, MM and DD are integers and MON is three or more letters, e.g., JAN or JANUARY (default=current day if DATASET keyword is specified; otherwise no default)
TIMe=	list only the stations in the specified DATASET that reported data at the specified DAY and TIME (default=current hour if DATASET keyword is specified; otherwise no default)

Remarks:

The frame navigation, cursor, and RADIUS and SIZE keywords are used to locate stations if MODE=I is specified or if none of the search keywords (CO, LAT, LON, MATCH, ST or USERID) or *id1..idN* are specified. MODE=I lets you select multiple geographic regions to list the stations within.

Station elevations are given in meters and all distance and radius values are great circle distances in kilometers, or the unit specified in the RADIUS keyword.

The following single-character codes are used to indicate station types in the *Data Types* columns of the STNLIST output:

Code	Station Type	Code	Station Type
3	3-Hourly Synoptic	6	6-Hourly Synoptic
A	SAO	C	CMAN Coastal Marine
E	NAM MOS **(formerly called ETA MOS)**	F	NGM MOS **(formerly called FOUS14)**
G	GFS MOS	M	METAR
N	NEXRAD Doppler Radar	R	RAOB
T	Terminal Aerodrome Forecast (TAF)	V	VOR Site
W	Weather Service Forecast Office (WSFO)		

If the DATASET keyword is not specified, STNLIST lists stations in the station database files **STNDB.CORE**, **STNDB.SITE** and **STNDB.USER**. The **.CORE** file is supplied with McIDAS-X and should not be modified. The **.USER** file overrides the **.SITE** file, which overrides the **.CORE** file. This order of precedence allows you or your site to change existing stations and/or add new stations in your **.USER** or **.SITE** file.

To create a **.USER** or **.SITE** file, first run a command similar to the following to create the file in the first writable **MCPATH** directory.

STNLIST KMSN FORM=USER DEV=T STNDB.USER A

The file contains one long line that defines the data for station KMSN. An equal sign (=) denotes the value of each of the 10 fields (IDN, ID, LATLON, ULATLON, ELE, UELE, ST, CO, NAME, and DATATYPE). Replace the values in the KMSN entry with those for your station and add lines as needed for additional stations. Entries do not have to include all ten fields. The required fields are LATLON and at least one of the following: ID, IDN, USERID. Other fields like CO, ST and NAME are useful but not required. Note that you must follow the one-entry-per-line format for all stations in your **.USER** and **.SITE** files. Line breaks are not permitted.

Three additional fields, USERID, COMMENT and VALID, may be optionally added to any station in your **.USER** and **.SITE** files. The USERID field is used to assign a unique identifier to your local stations. The COMMENT='comment1' . . 'commentN' field values are listed with the

FORM=ALL keyword. The VALID field is used to enter a date range for which the station is valid. If you specify a date range that has passed (e.g., VALID=2000001 2000111), the station is no longer considered valid so will not be listed or plotted with commands S, STNLIST or STNPLOT. The VALID field is useful for leaving station information in your local file and thus providing an audit trail of changes and storing the information in case the station needs to be re-entered at a later date.

Examples

STNLIST EIDW YSSY

This entry lists station information for Dublin, Ireland and Sydney, Australia.

STNLIST ST=TX TYPE=NEXRAD WSFO

This entry lists all NEXRAD Doppler Radar and Weather Service Forecast Offices in Texas.

STNLIST CO=ES MATCH='AEROPUERTO'

This entry lists all airport stations in Spain.

STNLIST LAT=-90 -60 FORM=ALL

This entry lists all stations, and their full station names, between 60 and 90 degrees South latitude (the Antarctic region). The LAT keyword was used instead of CO=AQ because some stations are listed under other country codes, e.g., AR, RU and US.

STNLIST

This entry lists all stations located within the region defined by the current cursor location on the frame. The stations are listed from closest to farthest from the center of the cursor.

STNLIST NUM=5

This entry lists the five stations closest to the center of the cursor.

STNLIST MODE=I RADIUS=50[KM] ST=IL TYPE=METAR

This entry uses the interactive mode to list stations in muliple regions on the frame. For each region selected by pressing the right mouse button or Alt G, the METAR stations in Illinois within 50 km of the cursor center are listed.

STNLIST DATASET=RTPTSRC/SFCHOURLY MODE=I

This entry uses the interactive mode to list stations in muliple regions on the frame. For each region selected by pressing the right mouse button or Alt G, the stations in dataset RTPTSRC/SFCHOURLY that reported data for the current hour are listed. Entries like this one are useful for quickly finding stations with surface hourly data in the vicinity of a feature observed on a satellite or radar image.

STNLIST DATASET=RTPTSRC/UPPERMAND ST=FL TIME=0 TYPE=RAOB

This entry lists the Florida stations in dataset RTPTSRC/UPPERMAND that reported RAOB data at 0 UTC today. Entries like this one are useful for quickly finding RAOB stations with data that can be used in commands like UALIST, UAPLOT and UACROSS.

STNPLOT

Plots stations on a navigated frame.

Format

STNPLOT *id1 . . idN [keywords]*

Parameters

id1 . . idN 3- to 5-character IDs or 5-digit ID numbers of stations to plot

---

Search Keywords

CO=	plot stations in this country; use the CCODE command to list valid 2-letter country codes	
LAT=	*slat nlat*	plot stations in this latitude range, where *slat* and *nlat* are the southern and northern latitude limits, respectively (def=frame latitude extents)
LON=	*elon wlon*	plot stations in this longitude range, where *elon* and *wlon* are the eastern and western longitude limits, respectively (def=frame longitude extents)
MATch=	*'string'*	plot stations with an exact match of the specified string, including any spaces
ST=	plot stations in this state; use its 2-letter state P.O. abbreviation	
TYPe=	*type1 . . typeN*	station types to plot; valid entries are the following single letters or their multi-letter versions: **M** or **METAR**, **A** or **SAO**, **S** or **SYNOPTIC**, **R** or **RAOB**, **T** or **TAF** or **TERMFCST**, **W** or **WSFO**, **N** or **NEXRAD**, **C** or **CMAN**, **V** or **VOR**, **E** or **NAMMOS** or **ETAMOS**, **F** or **NGMMOS** or **FOUS14**, **G** or **GFSMOS**; (default=ALL if *id1 . . idN* is specified, otherwise default=METAR SYNOPTIC RAOB)
USERid=	plot stations matching the USERID field in the **STNDB.USER** and **STNDB.SITE** files; see the Remarks	

Output Keywords

COLor=	*c1 . . cN*	graphics color level for each TYPE (default=3)
GRA=	graphics frame number to plot the stations (default=current)	
LSIze=	plot height, in pixels, of stations; the range is 1-50 (default=6 for FORMAT=ID,IDN or NAME; default=2 for FORMAT=POINT)	
NAV=	frame to use for navigation; specify a frame number or **C** for current frame (default=C)	
FORmat=	**ID**	plot the 3- to 5-character ID at each station location; see the Remarks (default)
	IDN	plot the 5-digit ID number at each station location

NAME	plot the station name at each station location
POINT	plot a point at each station location

Remarks

STNPLOT plots stations on a navigated frame. There are no map-drawing keywords, so you must display an image or map on the frame before using STNPLOT.

Some stations do not have both a 3- to 5-character ID and a 5-digit ID number. If FORMAT=ID is specified then the ID is plotted for those stations that have an ID, and the IDN is plotted for those that don't. Likewise, if FORMAT=IDN is specified then the IDN is plotted for those stations that have an IDN, and the ID is plotted for those that don't.

STNPLOT uses the station database files **STNDB.CORE**, **STNDB.SITE** and **STNDB.USER**. The **.CORE** file is supplied with McIDAS-X and should not be modified. The **.USER** file overrides the **.SITE** file, which overrides the **.CORE** file. This order of precedence allows you or your site to change existing stations and/or add new stations in your **.USER** or **.SITE** file.

To create a **.USER** or **.SITE** file, first run a command similar to the following to create the file in the first writable **MCPATH** directory.

STNLIST KMSN FORM=USER DEV=T STNDB.USER A

The file contains one long line that defines the data for station KMSN. An equal sign (=) denotes the value of each of the 10 fields (IDN, ID, LATLON, ULATLON, ELE, UELE, ST, CO, NAME, and DATATYPE). Replace the values in the KMSN entry with those for your station and add lines as needed for additional stations. Entries do not have to include all ten fields. The required fields are LATLON and at least one of the following: ID, IDN, USERID. Other fields like CO, ST and NAME are useful but not required. Note that you must follow the one-entry-per-line format for all stations in your **.USER** and **.SITE** files. Line breaks are not permitted.

Three additional fields, USERID, COMMENT and VALID, may be optionally added to any station in your **.USER** and **.SITE** files. The USERID field is used to assign a unique identifier to your local stations. The COMMENT='*comment1*' . . '*commentN*' field values are listed with the FORM=ALL keyword. The VALID field is used to enter a date range for which the station is valid. If you specify a date range that has passed (e.g., VALID=2000001 2000111), the station is no longer considered valid so will not be listed or plotted with commands S, STNLIST or STNPLOT. The VALID field is useful for leaving station information in your local file and thus providing an audit trail of changes and storing the information in case the station needs to be re-entered at a later date.

Examples

STNPLOT

This entry plots the 3- to 5-character ID at the location of all METAR, Synoptic, and RAOB stations on the current frame.

STNPLOT KORL KTLH

This entry plots the 4-character IDs for Orlando and Tallahassee, Florida at their locations on the current frame.

MAP USA
STNPLOT TYPE=RAOB CO=US FORMAT=IDN COLOR=5

These entries plot the 5-digit ID number at the location of all RAOB stations in the United States.

MAP FL
STNPLOT TYPE=C N COLOR=4 7

These entries plot the 5-character ID at the location of all CMAN stations and the 3-character ID at the location of all NEXRAD stations on a map of Florida. The CMAN stations are plotted in graphics color level 4 and the NEXRAD stations are plotted in graphics color level 7.

MAP DEF PRO=CONF LAT=-90 LON=0 SLAT=-60 SCALE=20
STNPLOT TYPE=S FORMAT=POINT LSIZE=3

These entries plot a point at the location of all Synoptic stations in Antarctica.

SU

Image data stretching utility.

Formats

SU INI *name type unit [keywords]*
SU MAKE *name inlo inhi britlo brithi*
SU LIST *string*
SU TABLE *name*

Parameters

INI	initializes the stretch table defining the type of calibration, calibration units and initial breakpoint settings; must be used when making a new stretch table
MAKE	enters breakpoints in the table using brightness levels
LIST	lists the saved stretch tables (default=lists all stretch tables on the workstation)
TABLE	lists the breakpoints stored in a stretch table
name	name of the stretch table; eight alphanumeric characters maximum
string	lists the stretch tables beginning with this character string
type	calibration type for the input table (default=any calibration type); to find the calibration type of an image, run IMGLIST FORM=ALL and see the value for *type* in the output
unit	calibration units for the input table; valid units are **RAW**, **BRIT**, **RAD**, **TEMP**, **ALB** or user-defined units; use command IMGPROBE or D to list available units for an image (default=**BRIT**)
inlo	low value of the breakpoint for input values; specify in units input with the *unit* parameter (no default)
inhi	high value of the breakpoint for input values; specify in units input with the *unit* parameter (no default)
britlo	low value of the breakpoint for output brightness values; the range is 0 to 255 (no default)
brithi	high value of the breakpoint for output brightness values; the range is 0 to 255 (no default)

Keywords

BAND=		band number for the input data calibration type (default=all bands)
RANGE=	*min max*	minimum and maximum values of the input data; these values define the boundary of input values that can later be stretched; see the Remarks

---

Remarks

Use command SU to make the tables specified with keyword SU in the IMGDISP command. SU sets the tables to stretch raw, radiance, temperature, albedo or brightness values (depending on calibration) to a user-defined brightness value. Then you can use command EU to color the brightness ranges corresponding to the new values.

If you specify keyword RANGE, all input values less than or equal to *min* are assigned an output value of zero; all input values greater than or equal to *max* are assigned an output value of 255. Any additional breakpoints specified with the MAKE option that fall outside of *min* and *max* are assigned 0 or 255, accordingly. If you don't specify RANGE=, the stretches' slope previous to the first breakpoint is a continuation of the line formed by the first two breakpoints. Likewise, the slope following the last breakpoint is a continuation of the line formed by the last two

breakpoints.

Examples

SU LIST

This entry lists all stretch tables saved on the workstation.

SU INI TIRORAD3 TIRO RAD BAND=3

This entry initializes table TIRORAD3 for use on the radiance values of TIROS band 3 images. Each new stretch table must be initialized.

SU MAKE TIRORAD3 0 1.2 20 220

This entry sets the breakpoint values in the above table TIRORAD3. Input 0 corresponds to output 20; input 1.2 corresponds to output 220.

SU TABLE TIRORAD3

This entry lists the breakpoints in table TIRORAD3 and information on BAND, calibration type and units.

SU INI MSATEMP1 MSAT TEMP BAND=1

This entry initializes the table MSATEMP1 to contain breakpoints in temperature for Meteosat band 1 data.

SU MAKE MSATEMP1 240 260 20 240

This entry stretches the temperature values from 240 to 260 in table MSATEMP1, over brightness values 20 to 240.

SU INI GENBRIT X BRIT RANGE=100 200

This entry initializes the table GENBRIT to be used on any calibration type with brightness units. The brightness values from 100 to 200 are stretched to brightness values 0 to 255. Brightness values entered with SU MAKE that are less than 100 or over 200 are assigned 0 or 255, respectively.

SU LIST GEN

This entry lists all the saved stretch tables that begin with GEN.

SUNLIT

Lists the sunrise and sunset times for a specified location.

Formats

SUNLIT *date station*
SUNLIT *date lat lon*

Parameters

date date of the sunrise/sunset; valid formats include: YYYYDDD, YYYY/MM/DD and DD/MON/YYYY, where YYYY, DDD, MM and DD are integers and MON is three or four letters, such as JAN or JANU

YES	yesterday
TOD	today (default)
TOM	tomorrow

station station identifier, e.g., KMSN or YSSY, optionally followed by the station type in brackets, e.g., ARX[N] or ARX[V]; see the TYPE keyword in STNLIST command for valid station types

lat latitude for sunrise and sunset

lon longitude for sunrise and sunset

Remarks

You must specify either a station identifier or latitude/longitude coordinates. The local noon time is an approximate calculated by finding the midpoint of the sunrise and sunset times.

Examples

SUNLIT X KPIT

This entry lists today's sunrise and sunset times for Pittsburgh.

SUNLIT TOM KPIT

This entry lists tomorrow's sunrise and sunset times for Pittsburgh.

SUNLIT 2003147 KMIA

This entry lists the sunrise and sunset times on 27 May 2003 for Miami.

SUNLIT 31/DEC/1999 32.9 97.1

This entry lists the sunrise and sunset times on 31 December 1999 at 32.9° North and 97.1° West.

SUNLIT 2005/05/23 -33:57 -151:11

This entry lists the sunrise and sunset times on 23 May 2005 at 33:57° South and 151:11° East.

SVF

Saves frames to a virtual frame file.

Format

SVF *bframe eframe file*

Parameters

bframe	beginning frame number to save (default=1)
eframe	ending frame number to save (default=*bframe*)
file	file name in which to save the frames; if you don't specify an extension, .PIX is automatically appended

---

Remarks

SVF saves the image, graphics, frame directory, navigation, image enhancements and the graphics tables from all frames specified in the *bframe* to *eframe* range. For example, SVF 1 10 LOOP saves those components from image frames 1 through 10 and graphics frames 1 through 10 into the file **LOOP.PIX**. Thus, when you restore the frames with the RVF command, full image and graphics functionality are retained so you can use commands like EG, ERASE, EU, GU, K and W. If you're running a session with independent graphics, you can also use commands like SF, SG, J and Y to display or loop the restored image and graphics frames independently.

If multiple frames are saved, they can be of different sizes. However, when the RVF command is used to restore them, the frames specified in RVF must be the same size as the saved frames. Command LVF with FORM=ALL lists the size of each of the saved frames so you can restore them to frames of the correct size with RVF.

Saved frames can also have multiple panels as long as an image and/or graphic is displayed in every panel. When a saved frame with multiple panels is restored with the RVF command, the frame that it's restored to acquires the panel configuration of the saved frame. For example, if a saved frame with 4 panels is restored with RVF to a single-panel frame, the frame is reconfigured to have 4 panels.

Because SVF saves all image and graphics information and saves frames of any size and panel configuration, you can use it to save all of your images and graphics into a single file, then restore them at a later time for additional work or to show for demonstration purposes.

Frames saved with SVF on little endian workstations (e.g., PC systems) can only be restored with RVF on the same workstation or other little endian workstations. Likewise, frames saved with SVF on big endian workstations (e.g., Sun SPARCstations and IBM RS/6000s) can only be restored with RVF on the same workstation or other big endian workstations.

See Also

Use the RVF command to restore these files to frames. Use the LVF command to list virtual frame files.

Examples

SVF 1 5 PICTURE

This entry saves image and graphics frames 1 through 5 into the file **PICTURE.PIX**.

SYNC

Synchronizes loops running in multiple briefing windows.

Format

SYNC *syncfile*

Parameter

syncfile file containing synchronization configuration

Remarks

Use the SYNC command to create briefing windows displaying synchronized frame loops. To run the command, you must first create a text file containing the synchronization configuration. The first line in the configuration file defines the dwell rate for each frame. Subsequent lines define the frame sequence for each briefing window, one line of frame numbers for each window. A maximum of nine briefing windows (windows 0-8) is allowed in the file.

The sample file shown below sets up two briefing windows, 0 and 1. Window 0 displays frames 1-4. Window 1 displays frames 5-8. The frames loop as fast as possible with a delay on the first frame.

```
4 1 1 1 1 2 3 4 5 6 7 8
```

If you want briefing window 1 to loop at half the speed of window 0, you could create a file like the one below. Frames 1-4 will loop twice as fast as frames 5-8.

```
1 1 1 1 1 1 1 1 1 2 3 4 1 2 3 4 5 5
6 6 7 7 8 8
```

You can include comments in the configuration file by starting a line with the pound sign (#). Comments may appear anywhere in the file but must be on a separate line. Blank lines are not allowed in the configuration file.

Because the SYNC command creates briefing windows, you should close any existing briefing windows before running the command. If any briefing windows are present, duplicate windows may be created causing faulty synchronization or other conflicts.

You may run out of memory trying to run SYNC. If this happens, try making your frames smaller, or use fewer briefing windows. If your McIDAS-X briefing windows appear with private color maps, they may not all be displayed with the correct colors when they are looping. If this happens, restart your McIDAS-X session with fewer image and graphics colors.

Example

SYNC SYNC1.TXT

This entry creates briefing windows and synchronously displays frames in them using the dwell rates and frame sequences specified in the file **SYNC1.TXT**.

SYNRPT

Lists synoptic observations.

Format

SYNRPT *id1 . . idn nhours [keywords]*

Parameters

id1 . . idn	station IDs or states to list; use the two-character state P.O. abbreviation for states
nhours	number of hours to search for observations starting from DAY and TIME (default=3)

Keywords

CO= *co1 . . con*	lists observations for all stations in these countries; use the CCODE command to list valid country codes
DATaset=	ADDE dataset name; specify as *group/descriptor* or *alias* (default=RTWXTEXT/SYNOPTIC)
DAY=	Julian date to start search; use either the 5- or 7-digit format, for example, 96300 or 1996300 (default=current)
TIMe=	UTC time to start search; valid times are 0, 3, 6, 9, 12, 15, 18 and 21 (default=current)

Remarks

You must specify at least one station ID, state, or country.

If you use the DAY keyword to specify a day other than the current day, you must also use the TIME keyword.

Examples

SYNRPT 03772

This entry lists all synoptic reports for London, England, from the past three hours.

SYNRPT MS 72793 8 CO=AT

This entry lists all synoptic reports for Austria, Mississippi, and Seattle, Washington, from the past eight hours.

SYNRPT MB 9 DAY=96300 TIME=12

This entry list all synoptic reports for Manitoba, Canada, from nine hours prior to 12:00 UTC on day 96300.

SYSKEY

Sets system values for local McIDAS workstation networks.

Format

SYSKEY *name*

Parameter

name name of the text file containing the values used to create the SYSKEY.TAB file (default=**SYSKEY.DOC**)

Remarks

Caution !
Creating or changing SYSKEY.TAB impacts the entire McIDAS workstation network.
SYSKEY should be used by the network administrator only.

This command reads the values from text file **SYSKEY.DOC** into the LW file **SYSKEY.TAB**. The parameters defined for local McIDAS workstation networks are stored in **SYSKEY.TAB**. These parameters contain alphanumeric and integer values needed by workstations in the network. Words 2100 to 2499 and 5000 to 5999 are reserved for parameters unique to your network.

The **SYSKEY.DOC** file must be in text format (lines must end with a linefeed). To change **SYSKEY.DOC**, use a text editor. Alphanumeric values are limited to four characters; integer values are limited to 10 digits.

Use command SYSVAL to list values directly from **SYSKEY.TAB**. **SYSKEY.TAB** values can be placed directly into local command entries using #SYS(*word*) to replace a parameter or keyword value. For example, IMGLIST GOES/VIS.1 #SYS(2100) lists images from position 1 to the value stored in word 2100.

Examples

SYSKEY

This entry sets the system values to the defaults in **SYSKEY.DOC**.

SYSVAL

SYSKEY utility.

Formats

SYSVAL LIST *bword eword*
SYSVAL CHANGE *word value*

Parameters

LIST	lists the values stored in the McIDAS system configuration file SYSKEY.TAB
CHANGE	changes the value of a SYSKEY word
bword	beginning word number to list (default=1)
eword	ending word number to list (default=*bword*)
word	SYSKEY word number
value	new value to place in SYSKEY; see the Remarks

Remarks

Caution !
Creating or changing SYSKEY.TAB impacts the entire McIDAS workstation network. The SYSVAL CHANGE option should be used by the network administrator only.

If you use the CHANGE option and designate the first character as a letter (A-Z), the value is stored in **SYSKEY.TAB** as a character string; otherwise, it is assumed to be an integer value and is converted to binary format and stored in **SYSKEY.TAB**.

The parameters defined for local McIDAS workstation networks are stored in **SYSKEY.TAB**. These parameters contain alphanumeric and integer values needed by workstations in the network. Words 2000 to 2499 and 5000 to 5999 are reserved for parameters unique to your network. For a description of **SYSKEY.TAB**, see the file **SYSKEY.DOC**. This file should be updated by the network administrator only.

SYSKEY.TAB values can be placed directly into local command entries using #SYS(*word*) to replace a parameter or keyword value. For example, IMGLIST GOES/VIS.1 #SYS(2100) lists images from position 1 to the value stored in word 2100.

Examples

SYSVAL LIST 2000 2010

This entry lists the contents of **SYSKEY.TAB** words 2000 to 2010.

SYSVAL CHANGE 5021 RA27

This entry places the string of ASCII characters RA27 into **SYSKEY.TAB** word 5021.

TAFRPT

Lists terminal aerodrome forecasts.

Format

TAFRPT *id1 . . idn nfcst [keywords]*

Parameters

id1 . . idn	station IDs or states to list; use the two-character state P.O. abbreviation for states
nfcst	number of forecasts to list (default=1)

Keywords

CO=	*co1 . . con* lists observations for all stations in these countries; use the CCODE command to list valid country codes
DATaset=	ADDE dataset name; specify as *group/descriptor* or *alias* (default=RTWXTEXT/TERMFCST)
TYPe=	type of forecast to list:
	LONG (default)
	SHORT
	ALL

Remarks

You must specify at least one station ID, state, or country.

Examples

TAFRPT KMSP

This entry lists the latest terminal aerodrome forecast for Minneapolis, Minnesota.

TAFRPT 2 CO=EC

This entry lists the last two terminal aerodrome forecasts for all stations in Ecuador.

TAFRPT YK PAFA

This entry lists the latest terminal aerodrome forecasts for Fairbanks, Alaska, and all stations in Yukon Territory, Canada.

TD

Deletes one or more strings from the current string table.

Formats

TD *name*
TD PREFIX *prefix*
TD ALL
TD ALL GLOB

Parameters

name	a single string name to delete
prefix	deletes all strings beginning with this prefix
PREFIX	searches for the string *prefix*
ALL	deletes the system table, except for global strings
GLOB	deletes all strings, including global strings

Remarks

A string name whose first character is a question mark (?) is defined as global. Global strings remain in the table unless you delete them by entering: TD ALL GLOB.

See Also

Command TL lists strings. Command TE edits strings. Command TU is the string utility that saves the current string table.

Examples

TD 9

This entry deletes the string named 9.

TD PREFIX CH

This entry deletes all strings that begin with CH.

TD ALL

This entry deletes all strings except those preceded by a question mark.

TD ALL GLOB

This entry deletes all strings, including those preceded by a question mark.

TE

Enters a string in the string table.

Format

TE *string* *"text*

Parameters

string	alphanumeric string name, 1 to 12 characters
"text	value, command, or text defining the string; the total length may be up to 160 characters including spaces; the text may include a pound sign (#) so another string's value is immediately substituted; use ## to substitute the value when *string* is run; the double quote is mandatory

---

Remarks

When *#string* appears in a command, it is replaced by its definition from the string table. When #SYS(*word*) or #UC(*word*) appears in a command, it is replaced by the value stored in the specified **SYSKEY.TAB** or User Common word, respectively. See the SYSVAL and UCU commands for more information.

If *string*'s *text* includes another string name preceded by a single pound sign (#), its value is immediately substituted. If the included string is preceded by two pound signs (##), the value is filled when *string* is run by typing *#string*.

Strings with the names KEYF, KEYAF, KEYCF, and KEYSF followed by a number between 1 and 12 (for example, KEYF1, KEYAF2) can be run using the function keys. See *Using Strings and Function Keys* for more information.

A string name whose first character is a question mark (?) is defined as global. Global strings remain in the table unless you delete them with TD ALL GLOB.

If the *string* name is a single letter (A-Z), you can run it by typing *#string* or by pressing **Ctrl** *string*.

Note

Do not use X as a string name; McIDAS interprets it as a place holder for default values. Likewise, if a series of strings begins with X (for example, XMAP), you cannot delete them with TD PREFIX X nor run them with REPEAT X 1 TO 1.

See Also

Command TL lists string tables; command TD deletes them; command TU is the string table utility. Command ASK1 can also be used to create a string or change its value.

Examples

TE LOAD

This entry assigns blanks to the string LOAD.

TE LOAD "IMGDISP GOES/VIS STATION=KMSN

This entry places an IMGDISP command in the string LOAD. To run the command, Type: #LOAD.

TE KEYF2 "MAP H

To run the MAP H command, press the F2 key.

TE NOTE "MAP OF NORTH AMERICA IN SATELLITE PROJECTION

This entry stores the text *MAP OF NORTH AMERICA IN SATELLITE PROJECTION* in the string NOTE.

TE AAA "TERM L ON

This entry stores the command TERM L ON in the string AAA.

In the compound string example below, remember to define the string replacing the # value before defining the string that uses it.

TE STN "KABQ

The replacement string is #STN.

TE WEATHER "SFCLIST #STN

The actual string stored in WEATHER is SFCLIST KABQ.

TE WEATHER "SFCLIST ##STN

To vary the value in #STN, insert two #s in the string WEATHER. The actual string stored in WEATHER is SFCLIST #STN.

If you change string STN now, the new value is used when WEATHER is run.

TE STN "KMIA

If you enter this change, the system uses SFCLIST KMIA the next time #WEATHER is run.

TERM

Sets the state of a McIDAS toggle command or mode.

Format

TERM *toggle state*

Parameters

toggle	toggle command to set; see the Remarks	
state	**ON**	switches the toggle state to on (default)
	OFF	switches the toggle state to off

Remarks

Use TERM when your workstation is not in a known state. TERM is commonly used in string tables and macros. The valid commands and their toggles are shown below. For more information, see the individual commands in this section of the manual.

toggle	**Function**	**ON (default)**	**OFF**
J	toggles graphics frames to the loop control system	connect	disconnect
K	toggles image frames on and off	visible	blank
L	toggles frame looping on and off	start	stop
O	displays the opposite image frame	opposite	base
W	toggles graphics frames on and off	visible	blank
Y	toggles image frames to the loop control system	connect	disconnect
Z	toggles the zoom on and off	on	off
IG	toggles the independent graphics mode on and off; see the Remarks	on	off
SCALE	toggles the image display mode; see the explanations in the Remarks section	scale	preserve
DMS	toggles the format of latitude and longitude values in the main GUI window, the E, D and IMGPROBE command output, and output to the command line generated by clicking the left mouse button while pressing the Ctrl key	dd:mm:ss	dd.ddd

---

Remarks

Use the TERM IG command to toggle McIDAS-X independent graphics on and off. Independent graphics are off by default when you start a McIDAS-X session; frames display both images and graphics. There are no separate image and graphics frames. Therefore, you cannot display or loop images or graphics independently, and commands like J, Y, LB G, LS G and SG are not valid.

The command TERM IG ON creates separate image and graphics frames, so you can display or loop them independently. This mode is most beneficial for users who create loops of colocated images because they don't have to draw the same map on every frame. Instead, they can draw the map on a single graphics frame and display only that graphics frame while looping through their series of image frames. Commands J, Y, LB G, LS G and SG are all valid in this mode.

You can specify **-c 'TERM IG OFF'** or **-c 'TERM IG ON'** in your **.mcidasrc** file to have McIDAS-X automatically start in the desired mode. If a TERM IG command is not set in your **.mcidasrc** and you want to change independent graphics mode, you should run the appropriate TERM command when you first start McIDAS-X (before displaying images or graphics). Changing modes later may not work correctly. Problems are more likely to arise if you switch from ON to OFF than from OFF to ON. If problems do arise, it's best to exit and restart McIDAS-X.

Use the SCALE ON and SCALE OFF options to preserve image levels in the range 0 to 255. Each preserved image level is assigned a different

display level when enhanced with the EU command. Image level scaling is on by default when you start a McIDAS-X session.

SCALE OFF preserves all image levels greater than the number of graphics color levels. Image levels less than or equal to the number of graphics color levels are not preserved; they are converted to the image level equal to the number of graphics color levels plus one. For example, if McIDAS-X is started with 16 graphics color levels, SCALE OFF converts image levels 0 to 16 to image level 17. Therefore, you cannot use the EU command to enhance any image level less than 17. Image levels 17 to 255 are preserved so they can be individually enhanced with the EU command. The SCALE OFF option is useful if the image you want to display contains little or no information in the lower end of the 0 to 255 range of image levels. It also lets you know which image levels are preserved and which are not.

SCALE ON preserves the same number of image levels as SCALE OFF; however, SCALE ON spreads the non-preserved image levels evenly over the entire 0 to 255 range. SCALE ON is most useful if the image you want to display contains important information in the lower end of the 0 to 255 range.

See Also

You can also use the SU command to preserve a specific set of image levels.

Examples

TERM L ON

This entry starts frame looping.

TERM L OFF

This entry stops frame looping.

TFILE

Sends text output to a disk file.

Format

TFILE *mode frame "file*

Parameters

mode	**OPEN**	opens the file for saving text
	CLOSE	closes the file (default)
	DUMP	dumps the contents of the text frame to the file
	DO	dumps the contents of the text frame and then opens the file
frame	text frame number; the range is 0 to 9 (default=current)	
"file	file name to write into; the double quote is mandatory	

Remarks

The specified file name cannot be in use by another task, such as a text editor. It may be fully qualified up to 40 characters.

When you activate a text frame for text capture, the text frame number appears in the TFILE field on the right end of the status bar.

Examples

TFILE

This entry closes the file that is capturing all text output written to the current text frame.

TFILE OPEN X "SURFACE.OBS

This entry opens the file **SURFACE.OBS** and places any further output directed to the current text frame into that file.

TFIND

Searches the current text frame.

Formats

TFIND U "*expression*
TFIND UC "*expression*
TFIND D "*expression*
TFIND DC "*expression*
TFIND T

Parameters

U	searches upward in the text frame for the first match of the specified expression; the search is not case-sensitive; it matches uppercase, lowercase and mixed-case text
UC	searches upward in the text frame for the first match of the specified expression; the search is case-sensitive; it only displays text that exactly matches the expression
D	searches downward in the text frame for the first match of the specified expression; the search is not case-sensitive; it matches uppercase, lowercase or mixed-case text
DC	searches downward in the text frame for the first match of the specified expression; the search is case-sensitive; it only displays text that exactly matches the expression
T	displays the first line of the text frame
"*expression*	expression to find in the text frame; must be entirely uppercase if using the U or D option; can be uppercase, lowercase or mixed-case if using the UC or UD option (default=last specified expression)

Remarks

Each of the ten text frames (numbered 0-9) can contain thousands of lines of text. You can set the number of lines for each text frame by specifying the **-oh** flag at startup or in your **.mcidasrc** file.

The Text and Command Window displays text frames. You can only view one text frame at a time and only a few lines of text. The TFIND command displays the line that matches a specified expression at the bottom of the Text and Command Window.

The far right side of the status bar in the Text and Command Window displays the offset number if you are viewing a location other than the bottom of the text frame, which has an offset of zero. For example, if 5[14] appears below the T on the status bar, it indicates that the 14th line from the end of text frame 5 is displayed.

If you enter TFIND with no parameters, it repeats the previous TFIND command. Thus, you can run one TFIND command with the direction and expression specified, then continue searching for the same expression by entering TFIND as many times as needed. If you want to change the direction of the search without changing the expression, enter TFIND U or TFIND D. Use TFIND T to display the first line of text in the text frame. Press the Enter key to return to the bottom. You can also move up or down in the text frame using the up and down arrow keys and the Page Up and Page Down keys.

Examples

TFIND

Repeats the previous TFIND command.

TFIND U "DONE

This entry searches upward in the text frame for a line that contains the expression DONE. The expression can be any mix of uppercase and lowercase letters if the letters are in the correct order with no spaces between. If a match is found, that line is displayed at the bottom of the Text

and Command Window.

TFIND D

If run after the previous example, this entry searches downward in the text frame for a line that contains the expression DONE. Like the previous example, the expression can be any mix of uppercase and lowercase letters if the letters are in the correct order with no spaces between. If a match is found, that line is displayed at the bottom of the Text and Command Window.

TFIND UC "abc DEF

This entry searches upward in the text frame for a line that contains the expression *abc DEF*. It must be an exact match, as the UC option is case-sensitive. There must be exactly one space between the *c* and the *D* since only one space was specified in the expression. If a match is found, that line is displayed at the bottom of the Text and Command Window.

TFIND T

This entry displays the top line of the text frame at the bottom of the Text and Command Window.

TIMEOUT

Runs a McIDAS command, stopping it after a waiting period.

Format

TIMEOUT *nseconds command*

Parameters

nseconds	maximum number of seconds that the command is allowed to run before it's stopped (no default)
command	complete McIDAS command to run, including parameters and keywords; see the Remarks

---

Remarks

TIMEOUT runs the specified McIDAS command and stops it if it hasn't completed within the specified waiting period. Use TIMEOUT when obtaining data from a remote server and the communications is slow or unreliable. For example, if you schedule multiple IMGDISP commands to run while the workstation is unattended, you can use the TIMEOUT command to avoid a large number of hung IMGDISP processes if communications problems arise.

If you use TIMEOUT to run a script, macro, BATCH file, or McBASI program that runs multiple McIDAS commands, the TIMEOUT waiting period begins with the first command and does NOT restart with each subsequent command. If you want to control the waiting period for each command, include each one in a separate TIMEOUT command. For example, if you run the following BATCH command:

BATCH CONTINUE=YES "IMGCOPY.BAT

and IMGCOPY.BAT consists of the commands below, the three IMGCOPY commands are run in order, each with a waiting period of 180 seconds.

TIMEOUT 180 IMGCOPY GOES/VISIBLE.0 MYDATA/VIS.1

TIMEOUT 180 IMGCOPY GOES/INFRARED.0 MYDATA/IR.1

TIMEOUT 180 IMGCOPY GOES/WATERVAPOR.0 MYDATA/WV.1

The CONTINUE=YES keyword is used so that each IMGCOPY command is run even if the previous command times out or stops abnormally.

Examples

TIMEOUT 90 IMGDISP GVAR/CONUSV STA=KCLE ALL=1 4

This entry runs the IMGDISP command, allowing 90 seconds for it to complete. If IMGDISP takes longer than 90 seconds, it is stopped by the TIMEOUT command.

TIMEOUT 300 RUN FILE=GETDATA.MCB

This entry runs the McBASI program GETDATA.MCB, allowing 300 seconds for it to complete. If it doesn't complete within 300 seconds, the program is stopped at the command that is running.

TIRCAL

Calibrates POES (TIROS) areas.

Format

TIRCAL *area* *[keyword]*

Parameter

area POES area number

Keyword

CAL=	**FULL**	applies a complete calibration (default)
	QUICK	applies quick calibration

Remarks

Caution!
Do not run TIRCAL on TIP data.

The FULL option calibrates each group of five image lines throughout the entire area. The QUICK calibration uses the average calibration for the first five good lines and applies that calibration to the entire area; thus, it runs more quickly.

The quick calibrated data shows only slight differences visually. Do not rely on the quick calibration for scientific purposes. Temperature differences of 2° to 3° Kelvin with radiance differences have been observed at the end of GAC areas.

Examples

TIRCAL 4200

This entry applies a complete calibration to area 4200.

TIRCAL 4210 CAL=QUICK

This entry applies a quick calibration to area 4210.

TL

Lists the strings in a string table.

Formats

TL ALL
TL OUT
TL *[keyword]*
TL *prefix [keyword]*

Parameters

ALL	lists the strings that begin with 0 to 9 or A to Z; the maximum is 256
OUT	lists all strings, including global strings
prefix	lists the strings beginning with this prefix

Keyword

Num=	number of strings to list (default=20)

Remarks

Strings in the string table are stored alphabetically with numbers 0 to 9 at the beginning of the table. A string whose first character is a question mark (?) is called a global string. It is listed with TL OUT or TL ?. To list other strings that do not begin with 0 to 9 or A to Z, use TL OUT.

See Also

Use command TD to delete strings and TE to enter strings. Command TU is the string table utility.

Examples

TL

This entry lists the first 20 strings from the current string table.

TL CH

This entry lists the strings that begin with CH.

TL ALL

This entry lists all strings; the maximum is 256.

TL NUM=10

This entry lists the first 10 strings in the string table.

TL TEMP NUM=15

This entry lists up to 15 strings starting with TEMP.

TOWNS

Lists the U.S. towns closest to the cursor location.

Format

TOWNS *number*

Parameter

number number of towns to list (default=20)

Remarks

The TOWNS command lets you select multiple locations by moving the cursor and clicking the right mouse button at each desired location. To end the command,

Type: **Alt Q**

The towns are listed from closest (first) to furthest (last) from the cursor location.

The state, town, county, latitude, longitude, and elevation (if available) are listed on the text frame.

This command uses data files provided by the U. S. Geological Survey, which are **towns.nd1**, **towns.nd2** and **towns.txt**.

Examples

TOWNS

This entry lists the 20 towns closest to the cursor location when you click the right mouse button. You can select several locations by moving the cursor and clicking the right mouse button at each location.

TOWNS 50

This entry lists the 50 towns closest to the cursor location when you click the right mouse button. You can select several locations by moving the cursor and clicking the right mouse button at each location.

TU

String table utility.

Formats

TU SAVE *name [keyword]*
TU DEL *name [keyword]*
TU LIST *[keyword]*
TU REST *name [keyword]*

Parameters

SAVE	saves the string table
DEL	deletes the string table
LIST	lists the saved string tables (default)
REST	restores a string table and makes it the current string table
name	1- to 12-character name identifying the string table

Keyword

SAVEFILE=	LW file name to store the table in or restore it from (default=SAVESTR)

Remarks

The maximum number of tables you can save is 512.

A string name whose first character is a question mark (?) is global and remains after each restore.

Examples

TU

This entry lists all string tables on the workstation.

TU SAVE DEMO

This entry saves the current string table under the name DEMO.

TU REST DEMO

This entry restores string table DEMO to the current string table.

TXT2MD

Creates an MD file from an ASCII text data file.

Format

TXT2MD *sfile dfile schema [keywords]*

Parameters

sfile	source ASCII text file name and extension
dfile	destination MD file number
schema	MD file schema name, for example: IRAB, ISFC

Keywords

CONTinue=	**NO**	stops filing data if a line of text that contains an invalid time format is encountered (default)
	YES	continues filing data if a line of text that contains an invalid time format is encountered
MISS=		value representing the missing data in *sfile*
ROW=		row to write the data (default=first available row)

Remarks

Use command TXT2MD to convert files containing lists of data to MD files. Before using this command, be sure that the schema is registered on your workstation. See commands LSCHE and SCHE for more information. Also see *Registering MD File Schemas*.

The *sfile* must be an ASCII text file with an end-of-line indicator at the end of each line. The *sfile* format must be columnar. The first line contains the schema key names; subsequent lines contain the corresponding data values, as shown below. Extra spaces between the columns are ignored.

```
TIME ID LAT LON ZS PSL TD T DAY TYPE NREC MOD ST CO 12 MEI 32.33 88.75 94 1009 267
272 1993072 0 9 0 MS US 12 NMM 32.55 88.56 97 1009 268 270 1993072 0 9 0 MS US 12
JAN 32.31 90.08 101 1015 265 273 1993072 0 9 0 MS US 12 MCB 31.18 90.46 126 1015
272 272 1993072 0 9 0 MS US 12 GWO 33.50 90.08 47 1014 264 273 1993072 0 9 0 MS US
12 CBM 33.65 88.45 67 1009 268 270 1993072 0 9 0 MS US 12 TUP 34.26 88.76 110 1010
265 272 1993072 0 9 0 MS US 12 GPT 30.40 89.06 9 1011 268 274 1993072 0 9 0 MS US
12 BIX 30.41 88.91 10 1011 269 273 1993072 0 9 0 MS US
```

Each data line in *sfile* is placed into one record in *dfile*. If repeat groups are defined in the schema, each data line must also include the data to be stored in the record's repeat groups.

If *dfile* does not exist, it will be created. If it already exists, the data is appended to the first available row or the row specified with keyword ROW. If you use keyword ROW, any column in the *dfile* row matching a column header from *sfile* will be replaced. If the column headers in *sfile* do not match the column headers in *dfile*, the data is appended to new columns at the end of the row.

The values for the row header keys in *dfile*, such as DAY and TIME, are set to the values found in the first data line of *sfile*. Values of the row header keys in subsequent data lines in *sfile* are ignored.

If the *dfile* schema has a key that represents time, such as HMS and TIME in the ISFC schema, it must be specified using the HH, HH:MM, or HH:MM:SS formats in *sfile*. For example, 12:00:00 UTC must be in the format 12, 12:00, or 12:00:00.

The *dfile* cannot have more than six column header keys or 10,000 columns. Each record in *dfile*, including repeat groups in that record, cannot contain more than 64 keys.

Examples

TXT2MD UPPRDATA.TXT 15 IRAB

This entry reads the ASCII text file **UPPRDATA.TXT** and puts it into MD file 15 of schema IRAB.

TXT2MD SFCDATA.TXT 5 ISFC MISS=-999

This entry reads the ASCII text file **SFCDATA.TXT** and puts it into MD file 5 of schema ISFC. Any data value set to -999 in **SFCDATA.TXT** is treated as missing when it is read into the MD file.

TXT2MD TEMPDATA 100 TEMP ROW=5

This entry reads the ASCII text file **TEMPDATA** and puts it into row 5 of MD file 100. TEMP is an MD file schema created and registered by the user.

UACROSS

Plots a vertical cross section of upper-air data.

Format

UACROSS *loc1 . . locN [keywords]*

Parameters

loc1 . . locN two or more locations to include in the cross section; each location can be specified as a 4-character station ID (e.g., KINL), a 5-digit station ID number (e.g., 72747), or a latitude/longitude pairing enclosed in single quotes (e.g.,'48.5 93.3' or '48:30:10 93:18:40'); see the Remarks for information about how the specified locations and other keywords are used to select and plot datapoints in the cross section

---

Data Selection Keywords
(Point and Grid)

DATaset= point or grid dataset(s) to use in the cross section; use the *mand sig* format for point datasets and *grid_dataset* format for grid datasets; the default is for point datasets as described below so you **must** specify this keyword to use a grid dataset; see the Remarks for information about the differences in datapoint selection with point data and grid data

mand sig point dataset(s) containing the mandatory- and significant-level data to use in the cross section; must also specify SIG=YES if you want significant levels included with the mandatory levels; currently only IRAB/IRSG and RAOB/RSIG schema point datasets can be used (*mand* default=RTPTSRC/UPPERMAND; *sig* default=RTPTSRC/UPPERSIG if *mand* =RTPTSRC/UPPERMAND and SIG=YES, otherwise *sig* default=NONE)

grid_dataset grid dataset containing the grids to use in the cross section

DAY= date of grids or observation date in point file; valid formats include YYYYDDD, YYYY/MM/DD and DD/MON/YYYY where YYYY, DDD, MM and DD are integers and MON is 3 or more letters, e.g., JAN or JANUARY (default=current for point data; no default for grids)

TIMe= time of grids or observation time in point file (for point data, default=0 if current time is between 0 UTC and 11:59 UTC, otherwise default=12; no default for grids)

MODe=

I interactive mode that uses the navigation of the current frame and the mouse to select locations for the cross section; you can select between 2 and 13 locations; see the Remarks for information about how the selected locations and other keywords are used to find and plot datapoints in the cross section; MODE=I is valid only if locations are not specified in the *loc1 . . locN* parameters (default=I if no locations are specified in *loc1 . . locN*)

N non-interactive mode where the cross section's locations are selected with the *loc1 . . locN* parameters (default=N if locations are specified in *loc1 . . locN*)

CORridor=

YES *size*[*units*]
use corridor(s) of the specified size to select datapoints for the cross section; a corridor is a rectangular region with a line running down the middle that connects the beginning and ending locations and a width equal to 2 times the *size* specified in this keyword; use *size* and/or the EXCLUDE or INCREMENT keyword to increase or decrease the number of datapoints in the cross section; run command CU LIST to list the valid *units*; see the Remarks for additional information about corridors

NO *size* [*units*]
do not use corridor(s); thus only the 3+ locations specified in *loc1 . . locN* or located with MODE=I are included in the cross section; for CORRIDOR=NO the *size* value and *units* specify the maximum radius from each location (selected with MODE=I or specified in *loc1 . . locN*) to use to find an endpoint; the closest datapoint within the radius is selected as the endpoint; if no datapoints are within the radius, UACROSS fails (default=YES 250[KM] if two locations are selected with MODE=I or are specified in *loc1 . . locN*; default=NO 250[KM] if 3 or more locations are selected or specified)

PRAnge= *min max* pressure levels, in mb, at top and bottom of diagram, respectively (default=100 1050)

Point-Only Data Selection Keywords

EXClude= *stn1 . . stnN* stations to exclude from the cross section; up to 8 stations may be excluded; each station can be specified as a 4-character station ID (e.g., KINL) or a 5-digit station ID number (e.g., 72747); this keyword is useful for re-running commands that select more stations than desired and/or include questionable data

SIG= **NO** do not include data from significant-level dataset specified in DATASET keyword; use only its mandatory-level dataset (default)

YES include data from significant-level dataset specified in DATASET keyword

---

Grid-Only Data Selection Keywords

FDAy= forecast day of grids to use in cross section; calculated by adding a grid's fhour to its day and time, e.g., a grid with day=2000300, time=12 and fhour=72 has fday=2000303; use with FTIME to select a specific forecast day and time; you cannot use FDAY with the FHOUR keyword

FHOur= forecast hour of grids to use in cross section; you cannot use FHOUR with the FDAY or FTIME keyword

FTIme= forecast time of grids to use in cross section; calculated by adding a grid's fhour to its time, e.g., a grid with time=12 and fhour=18 has ftime=6 on the following day; use with FDAY to select a specific forecast time and day; you cannot use FTIME with the FHOUR keyword

GPRo= projection of grids to use in cross section, e.g., MERC, PS, LAMB, EQUI

INCrement= *row col* grid point sampling increment for rows and columns; use this keyword and the CORRIDOR keyword to increase or decrease the number of grid points that can be used as datapoints in a cross section corridor; see the Remarks (*row* default=5; *col* default=*row*)

SRC= source of grids to use in cross section, e.g., ETA, GFS, MDX

Parameter Selection and Display Keywords

PARam= *par1* **[***unit1***]** *. . parN* **[***unitN***]**
list of up to 10 parameters to plot and their output units (default=THA[K]); the valid parameters are DIR, MIX, P, RH, SPD, SPDE, SPDN, SPDS, SPDZ, T, TD, TE, THA, THE, THW, TV, TW, U, V, WINDB, Z; see table in Remarks for parameter descriptions, their default units, and their defaults for the CINT, COLOR, DASH, LINT, LSIZE, and SMOOTH keywords

CINt= *c1 . . cN* contour interval for each PARAMeter; specify 0 to let UACROSS compute a reasonable interval, a value greater than zero to select your own value, or the name of a string that contains up to 64 values (default=value specified for corresponding parameter in **UACROSS.*** text file - see table and paragraphs before and after it in the Remarks)

COLor= *c1 . . cN* graphics color level for each PARAMeter (default=value specified for corresponding parameter in **UACROSS.*** text file - see table and paragraphs before it in Remarks)

DASh= *d1 . . dN* contour dash mode for each PARAMeter; specify one of the following values for each PARAMeter:
ALL - dash all contours
NONE - do not dash contours
NEG - dash negative contours only
POS - dash positive contours only
(default=value specified for corresponding parameter in **UACROSS.*** text file - see table and paragraphs before it in Remarks)

LINt= *l1 . . lN* contour label interval for each PARAMeter; the value must be greater than 0; the value 1 means label every contour, 2 means label every other contour, etc. (default=value specified for corresponding parameter in **UACROSS.*** text file - see table and paragraphs before it in Remarks)

LSIze= *l1 . . lN* contour label height, in pixels, for each PARAMeter (default=value specified for corresponding parameter in **UACROSS.*** text file - see table and paragraphs before it in Remarks)

SMOoth= *s1 . . sN* contour smoothing factor for each PARAMeter; the range is 3-50; larger numbers produce smoother contours (default=value specified for corresponding parameter in **UACROSS.*** text file - see table and paragraphs before it in Remarks)

Other Output Keywords

EG= **NO** do not erase graphics frame before plotting (default=YES)

GCOlor= *bg land tlab blab datapt p map path*
graphics color level of cross section components
(default=8 8 3 X 3 8 1 3), where:

bg background borders, vertical lines and tic marks

land land surface

tlab top label identifying beginning and ending datapoints

blab bottom label identifying plotted parameters; the default color for each parameter's label is the value specified in the COLOR keyword for that parameter

datapt datapoint labels at bottom of diagram

p pressure labels on left side of diagram

map inset map, when requested with MAP=YES keyword

path cross section path and datapoint markers on map

note: you can specify 0 to omit (not draw) any component

IDLabel= **NUMbers** display 5-digit ID numbers of plotted stations at bottom of cross section; valid only with point data (default for point data)

NAMes display 4-character IDs of plotted stations at bottom of cross section; valid only with point data

LALO display latitude/longitude pairings of plotted stations at bottom of cross section; valid with point or grid data (default for grid data)

GRA= graphic frame number to display cross section (default=current)

LABel= '*text*' *color line ele size*
text, color, location & size for cross section label, where:

'*text*' label's text; 80 characters maximum; specify X without quotes for default text (default=computed)

color graphics color level of label (default=3)

line beginning TV line to write label (default=lower-left corner of frame; line number depends on frame size)

ele beginning TV element to write label (default=20)

size pixel size of characters drawn (default=8)

note: if you specify this keyword, the default top and bottom labels (which identify beginning and ending datapoints, and plotted parameters, respectively) are **not** drawn

LIStinfo= **YES** list detailed station information (ID number, ID name, station name, state, country, latitude, longitude and elevation) of each station in text frame; valid only when using point data, i.e, not valid with grids

NO do not list detailed information; list only the ID number (default)

MAP= **NO** do not draw a map showing the cross section (default)

YES draws a map in the upper-right corner of the frame showing the cross section location, datapoints, and path as determined by the SPACING keyword

YES *'command'*
draws a map using the MAP command specified in '*command*' and draws the cross section datapoints and path as determined by the SPACING keyword; use this option to override the map domain and/or frame location that MAP=YES draws; the defaults for the '*command*' are the same as in the MAP command except for the LIN and ELE keywords, which default to the upper-right corner like the default MAP=YES output
Examples:

1. MAP=YES 'MAP MID X LALO'
draws a map of the Midwest with lat/lon lines instead of the default map computed by UACROSS

2. MAP=YES 'MAP MID X LALO LIN=300 450 ELE=10 210'
draws the same map but places it in the location specified with the LIN and ELE keywords instead of the upper-right corner of the frame

SPAcing= **NORMAL** spaces datapoints in the diagram using the normalized distances along the cross section axis; i.e., the distance between datapoints on the diagram is proportional to the distance between where their lines perpendicular to the axis intersect the axis; if you also draw a map using the MAP keyword, the cross section path is drawn along the axis

ACTUAL spaces datapoints in the diagram according to their actual distances from each other, not the normalized distances along the axis; if you also draw a map using the MAP keyword, the cross section path is drawn from datapoint to datapoint, in a zig-zag pattern as defined by the datapoint locations; this option is not recommended if grid data is used and CORRIDOR=YES is specified (default=ACTUAL if three or more locations and CORRIDOR=NO are specified, otherwise default=NORMAL)

Remarks

UACROSS plots contoured vertical cross sections of meteorological parameters using point or grid data. Use the DATASET keyword to specify the point or grid dataset, and keywords like TIME, DAY and FHOUR and SRC to narrow the search to more specific data. The cross section requires upper-air data from three or more datapoints. A datapoint is an observation location in a point file, or a grid point location in a grid. The datapoints are found using the locations specified in the *loc1 . . locN* parameters or the geographic points selected interactively with the mouse when MODE=I. The six paragraphs below explain how datapoints are selected and plotted in the cross section.

If you use point data and specify station IDs in *loc1 . . locN*, UACROSS uses the data only at those stations. That is, it will NOT look for data at a nearby station if the specified station didn't report data. Therefore, UACROSS will fail if any of the specified stations don't have data at the requested time.

If you use grid data, or use point data and specify latitude/longitude locations in *loc1 . . locN* or interactively choose locations with MODE=I, UACROSS selects the datapoint closest to each location if it's within the *size* value in the CORRIDOR keyword. Therefore, the lat/lons specified in *loc1 . . locN* or locations chosen with MODE=I do NOT have to exactly match those of stations in a point file or grid points in a grid.

If you specify only two locations in *loc1 . . locN* or select only two locations with MODE=I, they are used to find the beginning and ending datapoints (endpoints) of the cross section. The beginning datapoint is plotted on the left end and the ending datapoint is plotted on the right end of the diagram. Datapoints within the corridor defined by the endpoints and the *size* value in the CORRIDOR keyword are included in the cross section unless they're excluded by the program because they're too close to another datapoint or are excluded by the EXCLUDE or INCREMENT keywords.

If you specify 3 to 13 locations in *loc1 . . locN* and CORRIDOR=YES, or you select 3 to 13 locations with MODE=I and specify CORRIDOR=YES, multiple corridors are used and the first and second locations are used to find the endpoints for the first corridor, the second and third locations are used to find the endpoints for the second corridor, etc. Datapoints within each corridor defined by its endpoints and the *size* value in the CORRIDOR keyword are included in the cross section unless they're excluded by the program because they're too close to another datapoint or are excluded by the EXCLUDE or INCREMENT keywords.

If you specify 3 to 13 locations in *loc1 . . locN* and CORRIDOR=NO, or you select 3 to 13 locations with MODE=I and specify CORRIDOR=NO, corridors are not used, so only the datapoints for each location are included in the cross section. The datapoints are plotted from left to right on the cross section.

Although up to 25 datapoints can be included when CORRIDOR=YES is specified, a more practical limit is 15 or less. To decrease the number of datapoints, specify a smaller *size* value in the CORRIDOR keyword and/or larger *row* and *col* values in the INCREMENT keyword (which is valid only with grid data). For example, INC=8 6 allows datapoints to be selected only at grid points in every eighth row and every sixth column within the specified CORRIDOR. Longer and/or wider corridors often require larger INCREMENT values to keep the number of datapoints to 15 or less. The most uniform distribution of datapoints along a corridor is obtained by setting INC=1 and keeping the *size* value in CORRIDOR as small as necessary to reject neighboring grid points without excluding the anchoring endpoints. Use the SPACING keyword to specify how the sounding at each datapoint is projected onto the cross section.

The default units and keyword values for each parameter specified in the PARAM keyword are stored in the text files **UACROSS.USER**, **UACROSS.SITE**, and **UACROSS.CORE**. The **.USER** file overrides the **.SITE** file, which overrides the **.CORE** file. This order of precedence allows you to define your preferred defaults in your **.USER** file. The **.CORE** file is supplied with McIDAS-X and should not be modified. To create a **.USER** or **.SITE** file, copy the **.CORE** file from **~mcidas/data** to your **$HOME/mcidas/data** directory (or another **MCPATH** directory) and edit the values as needed.

The table below lists the contents of the **UACROSS.CORE** file, which includes the parameters that can be specified in the PARAM keyword, their default units, and their default values for the CINT, COLOR, DASH, LINT, LSIZE, and SMOOTH keywords. Default units enclosed in parenthesis cannot be changed. Otherwise the units can be changed with the PARAM keyword and/or by creating and modifying a **UACROSS.USER** or **UACROSS.SITE** file as described above. Likewise, the other values can be changed with their corresponding keywords and/or in **UACROSS.USER** or **UACROSS.SITE**.

	Default Values							
Param	**Unit**	**CINt**	**COLor**	**DASh**	**LINt**	**LSIze**	**SMOoth**	**Description**
DIR	(degrees)	90	1	ALL	1	6	20	wind direction
MIX	(G/KG)	2	4	ALL	1	6	20	mixing ratio
P	MB	50	8	NONE	2	6	20	pressure
RH	(%)	5	8	NONE	2	6	20	relative humidity
SPD	KTS	5	2	NONE	1	6	20	wind speed
SPDE	KTS	5	2	NEG	1	6	20	enhanced wind normal to cross section
SPDN	KTS	5	2	NEG	1	6	40	wind normal to cross section
SPDZ	(1/S)	5	2	NEG	1	6	20	vertical wind shear
T	C	5	5	NEG	2	6	20	temperature
TD	C	5	4	NEG	2	6	20	dewpoint temperature
TE	K	5	5	NEG	2	6	20	equivalent temperature
THA	K	5	5	NEG	2	6	20	potential temperature
THE	K	2	5	NEG	2	6	20	equivalent potential temperature
THW	K	1	4	NEG	2	6	20	wet bulb potential temp.
TV	K	5	5	NEG	2	6	20	virtual temperature
TW	K	5	4	NEG	2	6	20	wet bulb temperature
U	KTS	5	2	NEG	2	6	20	easterly wind component
V	KTS	5	3	NEG	2	6	20	northerly wind component
WINDB	(barb)	n/a	7	n/a	n/a	12	n/a	windbarb: flag=25mps(~50kts) long barb=5mps(~10kts) short barb=2.5mps(~5kts)
Z	M	2000	1	NEG	2	6	20	height from surface

The default contour interval for each parameter is taken from the appropriate **UACROSS.*** text file as described above the table. If you change the units of a parameter you may also want to change the contour interval as the defaults above are usually appropriate only for the default units. If you specify 0 for the contour interval, a value is calculated that produces a reasonable number of contours, usually between 10 and 20. In all cases the values will be adjusted slightly by the program to the nearest multiple of 1, 5, 10, 50, etc., consistent with the plot resolution.

Examples

UACROSS

This entry displays a cross section of potential temperature for locations selected interactively using the mouse. If two locations are selected, datapoints within a 500 km-wide corridor with endpoints at the two locations are selected. If three or more locations are selected, only the datapoint at or closest to (within 250 km) each location is selected.

UACROSS CORRIDOR=YES 150[MI] MAP=YES

This entry displays a cross section of potential temperature for locations selected interactively using the mouse. A 300 mile-wide corridor is computed for each pair of consecutively chosen locations (i.e., first and second, second and third, etc.). All datapoints within each corridor are plotted in the diagram unless they're excluded by the program because they're too close to another datapoint. A map is drawn in the upper-right corner showing the cross section location, datapoints and path.

UACROSS KMIA KBUF MAP=YES IDLABEL=NAMES PARAM=MIX THA SPD

This entry displays a cross section of mixing ratio, potential temperature and wind speed using the attributes defined for each parameter in the **UACROSS.USER**, **.SITE** or **.CORE** file. The cross section runs from Miami to Buffalo and includes datapoints within a 500 km-wide corridor between the two cities. The plotted stations are labeled with their 4-character IDs instead of their 5-digit ID numbers (e.g., KJAX instead of 72206). A map is drawn in the upper-right corner showing the cross section location, datapoints and path.

UACROSS KSLE KBOI 72672 KLBF PARAM=TD[F] SPDN[MPH] COLOR=2 3

This entry displays a cross section of dewpoint temperature and wind speed normal to the cross section. The units and graphics color levels specified in the command (F and 2 for TD; MPH and 3 for SPDN) override the settings in the **UACROSS.*** files. Because there are more than two locations specified and the default for the CORRIDOR keyword in this case is NO, only the four specified stations (Salem, Boise, Riverton, and North Platte) are included in the cross section.

UACROSS '38:40 09:10' 08221 07645 12843 UKKK MAP=YES CORRIDOR=YES LABEL='UACROSS across Europe' X 5 5 12

This entry displays a cross section of potential temperature for the five specified locations and the datapoints within the 500 km-wide corridors between each of pair of locations. As shown in this example, locations can be specified as latitude/longitude pairings enclosed in single quotes, 5-digit ID numbers or 4-character IDs. A map is drawn in the upper-right corner showing the cross section location, datapoints and path. The text *UACROSS across Europe* is plotted in graphics color level 3 and pixel size 12 near the top-left corner of the frame.

UACROSS KMIA KIAD MAP=YES DAY=#Y TIME=0 FHOUR=12 DATASET=RTGRIDS/ETA INCREMENT=2 2 CORRIDOR=YES 100[KM]

This entry displays a cross section of potential temperature between Miami and Washington, D.C. using the 12-hour forecast grids from today's 0 UTC run in the dataset RTGRIDS/ETA. The INCREMENT and CORRIDOR keywords work together to keep the number of grid points selected as datapoints to a reasonable limit of 15 or less. In this example, those keywords ensure that only grid points from every other row and column that are within 100 km of the axis from Miami to Washington D.C. are selected as datapoints. A map is drawn in the upper-right corner showing the cross section location, datapoints and path.

UALIST

Lists upper-air data.

Format

UALIST *station time day [keywords]*

Parameters

station	4-character station ID or 5-digit station ID number, e.g., KGRB or 72645
time	observation time (default=0 if current time is between 0 UTC and 11:59 UTC, otherwise default=12)
day	observation date; valid formats include YYYYDDD, YYYY/MM/DD and DD/MON/YYYY where YYYY, DDD, MM and DD are integers and MON is three or more letters, e.g., JAN or JANUARY (default=current)

Keywords

ANAlysis=	**YES**	include sounding/parcel analysis if OPT=ALL, ANL or MAN is specified; see the Remarks (default)
	NO	suppress sounding/parcel analysis in all cases
COuntry=	*c1 . . cN*	list data for soundings in these countries; specify up to six countries using their two-letter codes; use the CCODE command to list valid codes
DATaset=	*mand sig*	source ADDE point dataset names for mandatory- and significant-level data; specify as *group/descriptor* or *alias*; you must specify NONE in the *sig* parameter, or the OPT=MAN or the SIG=NO keywords to use only mandatory data, i.e., ignore the significant data (*mand* default=RTPTSRC/UPPERMAND; *sig* default=RTPTSRC/UPPERSIG if *mand* =RTPTSRC/UPPERMAND, otherwise, *sig* default=NONE)
DEPth=	depth in mb from the surface to include in the parcel analysis; the range is 0 - 400 where 0 means use only surface data (default=100)	
LAT=	*slat nlat*	list the data for soundings in this latitude range, where *slat* and *nlat* are the southern and northern latitude limits, respectively
LON=	*elon wlon*	list the data for soundings in this longitude range, where *elon* and *wlon* are the eastern and western longitude limits, respectively
OPTion=	**ALL**	list all mandatory and significant levels with computed potential temperature and mixing ratio at each level; temperatures are interpolated at each significant wind level, and winds are interpolated at each significant temperature level; include a sounding/parcel analysis below the sounding data unless ANA=NO is specified (default)
	ANL	list sounding/parcel analysis only; the data at each mandatory and significant level is not listed
	MAN	list only the mandatory levels with computed potential temperature and mixing ratio at each level; include a sounding/parcel analysis below the sounding data unless ANA=NO is specified
	RAW	list all mandatory and significant levels with computed potential temperature and mixing ratio at each mandatory and significant temperature level; do not include interpolated temperatures at significant wind levels or interpolated winds at significant temperature levels; do not include a sounding/parcel analysis
SIG=	**YES**	include data from the significant-level dataset specified in the DATASET keyword (default)
	NO	do not include data from the significant-level dataset specified in the DATASET keyword; use only its mandatory-level dataset

STate=	*s1 . . sN*	list data for soundings in these states; specify up to six states using their two-letter state Post Office abbreviations
UNIt=	*t spd z p*	units in which to list sounding parameters, where: *t*=temperatures; *spd*=wind speeds; *z*=heights; *p*=pressures; run CU LIST to list valid units and abbreviations (default=C KTS M MB)

Remarks

A sounding/parcel analysis is listed with OPT=ALL, ANL and MAN. The analysis is performed using mandatory- and significant-level data for OPT=ALL and OPT=ANL. The analysis is performed using only mandatory-level data for OPT=MAN. Each parameter in the analysis includes a description, abbreviation, value and units. The UAPLOT and HODO commands use these abbreviations for the values along the bottom of their diagrams. See *Appendix E, Stability Parameters* for definitions of the parameters.

To obtain the sounding/parcel analysis values, the reported heights are recomputed for hydrostatic consistency, though the adjustments are generally less than 10 meters. Also, the computed parcel temperature of a 12 UTC sounding is warmed 2 K to account for daytime heating. No adjustment is made to the computed parcel temperature of a 0 UTC sounding.

If UALIST is used with IRAB and IRSG schema MD files, the LAT, LON, STATE and COUNTRY keywords will find only mandatory (not significant) level data because the IRSG schema does not include those four parameters.

Examples

UALIST KEYW 0

This entry lists the mandatory and significant levels for Key West, Florida at 0 UTC of the current day. Wind direction and speed are interpolated at each significant temperature level, and temperature and dew point are interpolated at each significant wind level. Theta and mixing ratio are calculated at each level using the observed or interpolated values. A parcel/sounding analysis is listed below the sounding data. Note that you can also specify the station using Key West's 5-digit station ID number, 72201.

UALIST KEYW 0 OPT=RAW

This entry lists the mandatory and significant levels for Key West, Florida at 0 UTC of the current day. No interpolations are performed, so significant temperature levels do not include wind direction and speed, and significant wind levels do not include temperature, dew point, theta and mixing ratio. A sounding/parcel analysis is not listed.

UALIST KCHS 12 OPT=ANL DEPTH=50 UNIT=K

This entry lists a sounding/parcel analysis for Charleston, South Carolina at 12 UTC of the current day. Mandatory and significant levels are not listed. The sounding/parcel analysis values are computed using a boundary layer depth of 50 mb instead of the default value of 100 mb. The temperature values are listed in Kelvin instead of the default unit, Celsius.

UALIST X 12 1998310 OPT=MAN ANA=NO ST=FL

This entry lists the mandatory levels for all stations in the state of Florida at 12 UTC on day 1998310. Theta and mixing ratio are calculated at each level using the observed values. Significant levels and a parcel/sounding analysis are not listed.

UAPLOT

Plots a sounding on a thermodynamic diagram.

Format

UAPLOT *station time day [keywords]*

Parameters

station	4-character station ID or 5-digit station ID number, e.g., KGRB or 72645
time	observation time (default=0 if current time is between 0 UTC and 11:59 UTC, otherwise default=12)
day	observation date; valid formats include YYYYDDD,YYYY/MM/DD and DD/MON/YYYY where YYYY, DDD, MM and DD are integers and MON is three or more letters, e.g., JAN or JANUARY (default=current)

Keywords

ANAlysis= *text gra* controls display of sounding/parcel analysis values in the text frame and graphics frame; specify YES in *text* to list all the values in the text frame; specify YES in *gra* to plot selected values in the graphics frame below the diagram and the EL, LFC, and LCL markers on the diagram; specify NO in either or both to suppress the values (default=NO YES)

CENter= **YES** set the temperature range to place the profile near the center of the diagram; you cannot use CENTER with the TRANGE keyword

NO use the temperature range in the TRANGE keyword (default)

COLor= *t td win mix sat bg par ax*
graphics color level for each component (default=3 3 3 4 1 8 1 8 if OLAY=NO [the default]; default=2 2 2 4 1 8 1 8 if OLAY=YES)

t	temperature profile, sounding/parcel analysis values, plot title, and EL, LFC, LCL markers
td	dew point profile
win	wind barbs, or wind staffs and speeds (see the WIND keyword)
mix	constant mixing ratio lines
sat	saturation adiabats
bg	background grid (isotherms, dry adiabats, and wind pole)
par	sounding/parcel analysis labels and wind unit label
ax	pressure and temperature axes and labels

note: you can specify 0 to omit (not draw) any component

DATaset= *mand sig* source ADDE point dataset names for mandatory- and significant-level data; specify as *group/descriptor* or *alias*; you must specify NONE in the *sig* parameter or SIG=NO keyword to use only mandatory data, i.e., ignore the significant data (*mand* default=RTPTSRC/UPPERMAND; *sig* default=RTPTSRC/UPPERSIG if *mand* =RTPTSRC/UPPERMAND, otherwise *sig* default=NONE)

DEPth= depth in mb from the surface to include in the parcel analysis; the range is 0 - 400 where 0 means use only surface data (default=100)

EDIt= **YES** invoke the sounding editor after the diagram is plotted; valid only for FORM=SKEWT; the sounding editor lets you interactively change temperatures, dew point temperatures, wind speeds and directions; follow the instructions output by the command to change these values

NO do not envoke the sounding editor (default)

FORm= **SKEwt** display data on a skew T diagram, skewed 45 degrees in T vs. log P (default)

STUve display data on a Stuve diagram, linear in T vs. P**0.286 (R/CP)

LINear display data on a linear diagram, linear in T vs. P

EMAgram display data on an Emagram, linear in T vs. log P

GRA= graphic frame number to display diagram (default=current)

LEVels= **ALL** plot EL, LCL and LFC (Equilibrium Level, Lifted Condensation Level, and Level of Free Convection, respectively) markers on the diagram

NONE do not plot EL, LCL and LFC markers

lev1 lev2 plot only the specified marker(s); specify one or two of these levels: EL and/or LCL and/or LFC

OLAy= **YES** overlays the existing plot; the new wind data is drawn to the right of the OLAY=NO wind data, and the new plot title and sounding/parcel analysis values are written below the OLAY=NO title and values; use OLAY=YES to compare two soundings with different times or locations on a single plot

NO does not overlay the existing plot; erases the graphics then plots the entire diagram and sounding (default)

OUTlonly= **YES** draws a diagram without any data

NO draws a diagram and plots the upper-air data for the specified station (default)

PRAnge= *min max* pressure levels in mb at the top and bottom of the diagram, respectively (default=100 1050)

SIG= **YES** include data from the significant-level dataset specified in the DATASET keyword (default)

NO do not include the data from the significant-level dataset specified in the DATASET keyword; use only its mandatory-level dataset

TRAnge= *min max* temperatures, in degrees Celsius, at the top-left and bottom-right of the diagram, respectively; you cannot use TRANGE with the CENTER keyword (default=calculated, based on the pressure range in the PRANGE keyword)

UNIt= **MB** plot the height in millibars along the left axis (default)

FT plot the height in feet along the left axis

M plot the height in meters along the left axis

WINd= *type units scale spacing*

settings for wind data (default=BARB KTS 1.0 12), where:

type wind symbol, either BARB or STAFF where:

BARB wind barbs, where:
flag=50kts (~25mps),
long barb=10kts (~5mps),
short barb=5kts (~2.5mps)

STAFF wind staffs with the numeric wind speed plotted next to each staff's origin

units units of numeric wind speed plotted next to origin of each wind staff; run CU LIST to list valid units and abbreviations; changing units does not affect wind barbs

scale scaling factor used to change the relitive size of the wind barbs or staffs from the default; e.g., 2.0 doubles the size and 0.5 halves the size

spacing minimum space in pixels between the origins of the wind barbs along the vertical wind pole, or minimum space in pixels between wind speed values plotted next to each staff's origin

---

Remarks:

UAPLOT automatically erases the graphics before each plot unless OLAY=YES is specified.

Use the ANA keyword to list all available sounding/parcel analysis values in the text frame and plot selected values in the graphics frame below the diagram. To obtain the sounding/parcel analysis values, the reported heights are recomputed for hydrostatic consistency, though the adjustments are generally less than 10 meters. Also, the computed parcel temperature of a 12 UTC sounding is warmed 2 K to account for daytime heating. No adjustment is made to the computed parcel temperature of a 0 UTC sounding.

The following abbreviations are used for the sounding/parcel analysis values below the diagram: KI=K Index, LI=Lifted Index, SwI=Sweat Index, TTI=Total Totals Index, CVT=Convective Temperature in C, FMAX=Forecast Maximum Temperature in C, ThE=Equivalent Potential Temperature in K, EL=Equilibrium Level in mb, MIX=Mixing Ratio in g/kg, PW=Precipitable Water in mm, CAPE=Convective Available Potential Energy in J/kg, and HELI=Helicity in m**2/s**2. See *Appendix E, Stability Parameters* for definitions of these parameters.

Examples:

UAPLOT OUTL=YES

This entry plots a blank skew T diagram without data from a station.

UAPLOT KOUN 0
UAPLOT KCHS 0 OLAY=YES

The first entry plots a skew T diagram of mandatory- and significant-level data for Norman, Oklahoma at 0 UTC of the current day. The second entry overlays the existing plot with mandatory- and significant-level data for Charleston, South Carolina. The temperature profile, dew point profile, wind data, and EL (Equilibrium Level), LFC (Level of Free Convection), and LCL (Lifted Condensation Level) markers are drawn in graphics color level 3 for Norman, and graphics level 2 for Charleston.

UAPLOT KEYW 12 1998310 FORM=STUVE ANA=YES DEPTH=50

This entry plots a Stuve diagram of mandatory- and significant-level data for Key West, Florida at 12 UTC on day 1998310. The ANA=YES keyword outputs all parcel/sounding analysis values in the text frame and selected values below the Stuve diagram in the graphics frame. The sounding/parcel analysis values are computed using a boundary layer depth of 50 mb instead of the default value, 100 mb.

UAPLOT KOUN 0 PRANGE=500 850 CENTER=YES WIND=STAFF

This entry plots a skew T diagram of mandatory- and significant-level data for Norman, Oklahoma at 0 UTC of the current day. The diagram's pressure range is 500 to 850 mb instead of the default range, 100 to 1050 mb. The CENTER=YES keyword sets the diagram's temperature range so the temperature and dew point profiles are near the center. Winds are plotted as staffs with the numeric wind speed next to the origin of each staff.

UAPLOT KMFR 12 06/JUNE/1998 DATASET=LOCAL/MAND.1017

This entry plots a skew T diagram of mandatory-level data for Medford, Oregon at 12 UTC on 6 June 1998. The data is obtained from the file in position 1017 of the dataset LOCAL/MAND.

UCU

Inspects the User Common (UC) memory block.

Formats

UCU PEEK *word*
UCU POKE *word value*

Parameters

PEEK	lists the current value in User Common
POKE	places a value in User Common
word	specific word number in User Common
value	new value to place in User Common

Remarks

User Common consists of workstation information that resides in memory. Programs access User Common to obtain current workstation configuration and state data. Each User Common word stores a separate piece of information, such as cursor line and element position, current image and graphics frame, current logon initials and project number.

User Common values can be placed directly into local command entries by using #UC(*word*) to replace a parameter or keyword value. For example, LB 1 #UC(13) sets the image loop bounds from 1 to the value stored in User Common word 13.

Caution !
Do not alter User Common. If you incorrectly modify a UC value, you may need to exit and restart McIDAS to reinitialize its state.

Example

UCU PEEK 1

This entry lists the contents of User Common 1 (project number).

USAORB

Displays polar orbital tracks and coverage for North America.

Formats

USAORB *navdataset frame [keywords]*
USAORB *imgdataset frame [keywords]*

Parameters

navdataset	name of the ADDE dataset containing the system navigation files for a satellite; specify as *group*/*descriptor* or *alias*
imgdataset	name and position of the ADDE dataset containing the image to use as a navigation source; specify as *group*/*descriptor*.*position* or *alias*.*position*; when specifying an image dataset, you must also use the TYPE=I keyword (*position* default=0)
frame	frame number for display (default=current)

Keywords

CColor=	graphics color level of the lines displaying the geographical coverage of the satellite sensor or the boundaries of the image specified in *imgdataset*; use positive numbers for solid lines and negative numbers for dashed lines; see the Remarks (default=0, do not plot coverage)	
DAY=	*bday eday*	range of days for display; see the Remarks (default=current, or from *imgdataset*)
HORizon=	*gil wal*	minimum angle, in degrees, above the horizon at which a satellite signal is readable
	gil	angle at Gilmore Creek, Alaska (default=5)
	wal	angle at Wallops Island, Virginia (default=5)
INT=	interval, in seconds, between the time marks on the orbital track (default=computed)	
LABel=	'*text*' *loc col*	display label, location, and color
	'*text*'	label or description of the display (default=from *navdataset* or *imgdataset*)
	loc	location for the label at the bottom of the frame; specify a number 1 through 5; see the Remarks (default=1)
	col	graphics color level for the label (default=from the CCOLOR keyword if used; otherwise from the TRCOLOR keyword)
NAV=	**C**	use existing frame navigation; use this keyword to overlay additional orbital tracks on the same frame (default=erase frame and redraw map)
SENsor=	*type*	
	POES options:	AVHRR (default), HIRS, MSU
	DMSP options:	OLS (default), SSM/I, SSM/T, or SSM/T2
TIME=	*btime etime*	time range to plot; see the Remarks (default=0 23:59:59, or from *imgdataset*)
TMColor=	graphics color level of the time marks on the orbital track (default=3)	
TRColor=	graphics color level of the orbital track; use positive numbers for solid lines and negative numbers for dashed lines (default=2)	

TYPE=	**S**	uses the system navigation files specified with *navdataset* (default)
	I	uses the navigation of the image specified with *imgdataset*; overrides the SENSOR keyword

---

Remarks

Use both the DAY and TIME keywords to specify the time period for plotting orbital tracks and coverage. For example, DAY=182 183 TIME=21 3 specifies a range from 21:00 UTC on day 182 through 03:00 UTC on day 183.

If you use the CCOLOR keyword and TYPE=I without the DAY or TIME keyword, the displayed geographical coverage is limited to the image specified in *imgdataset*. Otherwise, the entire geographical coverage of the sensor is displayed.

Labels are displayed at the bottom left corner of the frame. A default label is constructed using information from the system navigation file or the image used for navigation. The label includes the dataset name, the satellite, sensor, and day or range of days displayed. Use the LABEL keyword and the '*text*' parameter if you want to choose a different label.

Specify NAV=C to overlay additional orbital tracks on the same frame. Use the *loc* parameter with the LABEL keyword to place additional labels on different lines at the bottom of the frame. Location 1 is the top line; additional labels are below. The maximum number of lines depends on the frame size: five for a 480 by 640 frame.

See Also

Use the PACORB command to display orbital tracks and coverage for the Pacific Test Range. Use the NAVDISP command to display orbital tracks and coverage for a user-defined map.

Examples

USAORB NAV/NOAA14 DAY=96300

USAORB NAV/NOAA12 DAY=96300 NAV=C TRCOLOR=5 LABEL=X 2

The first entry draws the NOAA-14 satellite orbital tracks for day 96300 over a map of North America. The second entry draws the NOAA-12 satellite orbital tracks on the same frame without erasing the NOAA-14 tracks. The NOAA-12 tracks and display label are drawn in graphics color level 5. The label is positioned on the text line below the first entry's label.

USAORB POESNAV/NOAA14 TIME=13:00 15:00 SENSOR=MSU CCOLOR=-7

This entry draws the NOAA-14 satellite orbital tracks from the current day between 13 and 15 UTC and displays the geographical coverage of the MSU sensor in dashed color level 7. The track data is drawn over a map of North America.

USAORB WALLOPS/N12HRPT TYPE=I

This entry draws the orbital track for the polar orbiting satellite with data in most recent image in the dataset WALLOPS/N12HRPT over a map of North America.

USAORB WALLOPS/N12HRPT TYPE=I CCOLOR=10

This entry draws the orbital track and the AVHRR geographic coverage for the satellite with data in the most recent image in the dataset WALLOPS/N12HRPT over a map of North America. The geographic coverage is drawn in color level 10.

USAORB WALLOPS/N12HRPT TYPE=I DAY=97097

This entry draws the orbital tracks for the satellite with data in the most recent image in the WALLOPS/N12HRPT dataset. The tracks for the entire day 97097 are plotted.

USAORB GILMORE/N14GAC TYPE=I NAV=C LABEL='ORBITAL TRACK FOR NOAA-14 #Y' INT=300

This entry draws the orbital track for the satellite with data in the most recent image in the GILMORE/N14GAC dataset using the current frame's navigation. A custom label is displayed, and the time marks are drawn every five minutes along the orbital track.

V

Toggles the velocity cursor on and off when tracking cloud drift winds.

Remarks

The V key toggles the velocity cursor on and off. This establishes a fixed jump in the cursor for each frame in your loop bounds, which helps track cloud features. You must be in a cloud tracking session to use command V.

To turn the velocity cursor on,

Press: **Alt V**

The letter V is displayed on the text frame status line.

Set the velocity increment by aligning the cursor over a cloud target on the first frame of your loop. Advance to the second frame with the left mouse button (it functions like **Alt A**), then align the cursor over the similar cloud feature. Doing this on the first and second frames sets the increment for all frames in the loop.

If you don't use the velocity cursor, you will be manually selecting each target. Although this can be quite tedious, it is advisable when doing single pixel cloud tracking.

VASTBL

Creates calibration transforms for Mode AA VAS data.

Format

VASTBL

Remarks

Mode AA data was transmitted by GOES satellites between 22 September 1980 and 18 November 1986. This command creates a file of about 6.5 megabytes named VASTBLS that is used specifically for 2-byte Mode AA VAS data. If you use this data on your workstation, you must run this command once to create this file.

Example

VASTBL

This entry creates the file VASTBL, which is needed only if you use Mode AA VAS data on your workstation.

W

Toggles the graphics frame on and off.

Remarks

To turn the graphics overlay frame on and off,

Press: **Alt W**

or

Type: **W**

Press: **Enter**

This entry does not alter the contents of the graphics frame. It turns the graphics off to provide a better view of the image.

See Also

Command K toggles image frames on and off.

WWDISP

Displays active weather warnings, watches and advisories.

Format

WWDISP *map frame [keywords]*

Parameters

map map to display; use any predefined map in the MAP command, for example, USA, MID, KS (default=no map is drawn, use frame navigation)

frame frame number to display the output (default=current)

Keywords

COLor= *tor svr toroutl svroutl win winoutl snoadv bliz blizoutl fldwarn fldwatch*
graphics color levels for output (default=2 3 5 4 14 10 11 7 8 6 9 for sessions with nine or more graphics color levels; 2 3 5 4 6 2 4 7 8 6 1 for sessions with eight or less graphics color levels)

tor	color of tornado watch outlines and warnings fill
svr	color of severe thunderstorm watch outlines and warnings fill
toroutl	color of county outlines drawn for tornado warnings
svroutl	color of county outlines drawn for severe thunderstorm warnings
win	color of winter storm watch outlines and warnings fill
winoutl	color of zone outlines drawn for winter storm warnings
snowadv	color of zone outlines drawn for snow advisories
bliz	color of zone fill drawn for blizzard warnings
blizoutl	color of zone outlines drawn for blizzard warnings
fldwarn	color of flood/flash flood warning outlines and fill
fldwatch	color of flood/flash flood watch outlines and fill

DAY= *bday eday* day range of active bulletins to plot (default=current *bday*)

FILl= type of shading for counties and National Weather Service zones under warnings; uses the colors specified for *svr*, *tor*, *win* and *bliz* with the COLOR keyword; use one of the following options

NONE	use no fill
SOLID	fill completely with a solid color (default)
TRAN *density*	fill partially to give a transparent effect; *density* range is 2-10; larger values give a greater transparent effect, for example, a value of 5 fills every fifth pixel (*density* default=4)

GROup= ADDE group to search; the group name must be mapped with the DATALOC command to a server with weather text data (default=RTWXTEXT)

NAV= frame to use for navigation; specify a frame number or C for current frame; cannot be used with the *map* parameter (default=C)

PLOt=	**BOX**	plot severe thunderstorm and tornado watches as box outlines
	COUNTY	plot severe thunderstorm and tornado watches as county outlines
	NBOX *height*	plot severe thunderstorm and tornado watches as box outlines with the watch number in center of each box using the specified pixel height (default=NBOX 7)
TIMe=	*btime etime*	time range of active bulletins to plot (default=current *btime*)
TYPe=	*t1 . . tn* bulletin types to plot; valid types are listed below; you cannot use TYPE with the WATCH keyword	
	BLIZZARD	blizzard warnings
	FLOODWARN	flood/flash flood warnings
	FLOODWATCH	flood/flash flood watches
	SEVWARN	severe thunderstorm warnings
	SEVWATCH	severe thunderstorm watches
	SNOWADV	snow advisories
	TORWARN	tornado warnings
	TORWATCH	tornado watches
	WINADV	winter weather advisories
	WINWARN	winter storm warnings
	WINWATCH	winter storm watches
	ALL (default)	
WATch=	*num1 . . numn*	severe thunderstorm and tornado watch numbers to plot; watches are plotted only if they are active during the time period specified with the DAY and TIME keywords; you cannot use WATCH with the TYPE keyword (default=all active watches during time period specified with DAY and TIME)
WIDth=	*width*	graphics line width for county outlines and watch boxes (default=current GD command setting)
WList=	**YES**	display the watch list in the upper-left corner of the frame; the list includes the watch number, the day of month, and the time it expires (default)
	NO	do not display the watch list

Remarks

If both watches and warnings are active, the watches are plotted before the warnings, regardless of which were issued first. The map is drawn after all the watches and warnings are plotted.

Severe thunderstorm and tornado watches are drawn as either a box or county outlines. If you use PLOT=NBOX (the default), the watch number is plotted in the center of each box. The number begins with a T, indicating tornado, or an S, indicating severe thunderstorm. Winter storm warnings are always plotted as zone outlines.

Watches are outlined in the *tor*, *svr* and *win* colors specified with the COLOR keyword. Counties and zones under a warning are filled with the *tor*, *svr*, *win* and *bliz* colors using the shading specified with the FILL keyword, and are outlined with the *toroutl*, *svroutl*, *winoutl* and *blizoutl* colors specified in the COLOR keyword. Flood/flash flood watches and warnings, and snow advisories are filled with the *fldwatch*, *fldwarn* and *snoadv* colors in a hatch pattern unless FILL=NONE is specified.

If you specify a range with the DAY and TIME keywords, any bulletins active during the range are plotted. For example, DAY=185 187 TIME=21 3 specifies a window from 21:00 UTC on day 185 through 03:00 UTC on day 187. If you specify a single day and time, bulletins active at that day

and time are plotted. The default is the current day and time.

Canceled severe thunderstorm and tornado watches are considered expired and are not displayed. Canceled warnings and winter storm watches are displayed until the original expiration time has passed.

The National Weather Service redefined winter weather bulletins beginning 09 September 2008 (NWS SCN08-40) such that Snow Advisories (TYPE=SNOWADV) are no longer issued; Winter Weather Advisories (TYPE=WINADV) were expanded to include snow, blowing snow, and sleet; and Winter Storm Warnings (TYPE=WINWARN) were expanded to include heavy snow and sleet. The TYPE=SNOWADV option was left in this command so it's valid when using archive data.

Examples

WWDISP

This entry plots all active severe thunderstorm watches and warnings, tornado watches and warnings, winter storm watches and warnings, flood/flash flood watches and warnings, blizzard warnings, and snow advisories on the current frame. No map is drawn; the current frame's navigation is used.

WWDISP USA 4 TYPE=TORWATCH TORWARN

This entry plots all active tornado watches and warnings on a map of the United States on frame 4.

WWDISP TX COLOR=5 4 6 6 TYPE=SEVWATCH SEVWARN TORWATCH TORWARN

This entry plots all active severe thunderstorm and tornado watches and warnings on a map of Texas on the current frame. The tornado watches and warnings are plotted in graphics color level 5. The severe thunderstorm watches and warnings are plotted in graphics color level 4. The outlines of counties with warnings are displayed in graphics color level 6.

WWDISP MID WATCH=490 492 PLOT=COUNTY

This entry plots severe thunderstorm or tornado watches 490, 491 and 492 as county outlines on a map of the Midwest. The watches are plotted only if they are currently active. If they are no longer active, you must use the DAY and TIME keywords to specify the day and time when they were active.

WWDISP USA DAY=2003115 2003119 TIME=0 23:59

This entry plots all severe thunderstorm watches and warnings, tornado watches and warnings, winter storm watches and warnings, flood/flash flood watches and warnings, blizzard warnings, and snow advisories that were active on days 2003115 through 2003119 on a map of the United States.

IMGDISP GOES8/VIS BAND=1 STA=KBUF REFRESH='MAP VH'
WWDISP NAV=C DAY=1998361 TIME=18 23:59 TYPE=WINWARN BLIZZARD FILL=TRAN 6

The first entry displays the most recent image in dataset GOES8/VIS on the current frame with Buffalo, New York, at the center, and includes a high-resolution map. The second entry uses the image's navigation to plot all winter storm and blizzard warnings active between 18 and 23:59 UTC on day 1998361. Every sixth pixel in the zones under a warning is filled, producing a transparent effect that allows most of the satellite image to be seen.

WWLIST

Lists active weather warnings, watches and advisories.

Format

WWLIST *state* '*county*' *[keywords]*

Parameters

state	state to list; two-character P.O. abbreviation (default=all)
'*county*'	county name or National Weather Service state zone number to list; single quotes are mandatory (default=all)

Keywords

DAY=	*bday eday*	day range of active bulletins to list (default=current *bday*)
GROup=	ADDE group name to search; the group name must be mapped with the DATALOC command to a server with weather text data (default=RTWXTEXT)	
OUTput=	**RAW**	list output as the original text product
	TAB	list output in a condensed, tabular format (default)
TIMe=	*btime etime*	time range of active bulletins to list (default=current *btime*)
TYPe=	*t1 . . tn*	bulletin types to list; valid types are listed below; you cannot use TYPE with the WATCH keyword
	BLIZZARD	blizzard warnings
	FLOODWARN	flood/flash flood warnings
	FLOODWATCH	flood/flash flood watches
	SEVWARN	severe thunderstorm warnings
	SEVWATCH	severe thunderstorm watches
	SNOWADV	snow advisories
	TORWARN	tornado warnings
	TORWATCH	tornado watches
	WINADV	winter weather advisories
	WINWARN	winter storm warnings
	WINWATCH	winter storm watches
	ALL (default)	
WATch=	*num1 . . numn*	severe thunderstorm and tornado watch numbers to list; watches are listed only if they are active during the time period specified with the DAY and TIME keywords; you cannot use WATCH with the TYPE keyword (default=all active watches during time period specified with DAY and TIME)

Remarks

Omit punctuation when specifying county names with the '*county*' parameter. For example, to list the watches and warnings for St. Cloud county, specify 'ST CLOUD', not 'ST. CLOUD'.

If both watches and warnings are active for the specified location, WWLIST lists the watches before the warnings, regardless of which were issued first. If multiple active bulletins of the same type are found, they are listed in order from oldest to most recent.

If you specify a range with the DAY and TIME keywords, any bulletins active during the range are listed. For example, DAY=185 187 TIME=21 3 specifies a window from 21:00 UTC on day 185 through 03:00 UTC on day 187. If you specify a single day and time, bulletins active at that day and time are listed. The default is the current day and time.

The National Weather Service redefined winter weather bulletins beginning 09 September 2008 (NWS SCN08-40) such that Snow Advisories (TYPE=SNOWADV) are no longer issued; Winter Weather Advisories (TYPE=WINADV) were expanded to include snow, blowing snow, and sleet; and Winter Storm Warnings (TYPE=WINWARN) were expanded to include heavy snow and sleet. The TYPE=SNOWADV option was left in this command so it's valid when using archive data.

Examples

WWLIST

This entry lists all active severe thunderstorm watches and warnings, tornado watches and warnings, winter storm watches and warnings, flood/flash flood watches and warnings, blizzard warnings, and snow advisories for the United States.

WWLIST TX TYPE=TORWATCH TORWARN

This entry lists all active tornado watches and warnings for Texas.

WWLIST FL 'ST LUCIE' OUTPUT=RAW TYPE=TORWARN

This entry lists all active tornado warnings for St. Lucie county in Florida. The data is listed in the text format of original bulletins.

WWLIST OK DAY=2003116 TIME=0 23:59

This entry lists all severe thunderstorm watches and warnings, tornado watches and warnings, winter storm watches and warnings, flood/flash flood watches and warnings, blizzard warnings, and snow advisories that were active in Oklahoma on day 2003116.

WWLIST WATCH=490 491 492

This entry lists severe thunderstorm or tornado watches 490, 491 and 492 as county. The watches are listed only if they are currently active. If they are no longer active, you must use the DAY and TIME keywords to specify the day and time when they were active.

WXTLIST

Lists weather text products.

Formats

WXTLIST *predprod [keywords]*
WXTLIST *[keywords]*
WXTLIST DIR

Parameters

predprod	lists a predefined text product; see the Remarks (no default)
DIR	displays a list of the available predefined products

Keywords

APRo=	*head1 . . headn*	lists text products with the specified AFOS/AWIPS product headers; must be three characters; you cannot use APRO with the WMO keyword; see the Remarks
ASTn=	*stn1 . . stnn*	lists text products with the specified AFOS/AWIPS stations; must be two or three characters, for example, SFO, CA
DAY=	*bday eday*	range of days to search; see the Remarks for DAY, TIME and DTIME interactions (*bday* default=current; *eday* default=*bday* if a range is specified using the TIME keyword; if a range is not specified in either DAY or TIME, then *bday*, *btime* and DTIME determine the day and time range)
DTIme=	maximum number of hours back in time to search; the range is 1-720; see the Remarks for DAY, TIME and DTIME interactions (default=96 when used with the APRO, ASTN, WMO and WSTN keywords; otherwise default=1)	
EXClude=	'*string*'	do not list text products that contain an exact match of the specified string, including any spaces
	string1 . . stringn	do not list text products containing all the specified character strings; the strings can occur anywhere in the product
FORm=	**TEXT**	lists only the text products (default)
	HEADER	lists only the header information
	ALL	lists both the header information and text products
GROup=	ADDE group to search; the group name must be mapped with the DATALOC command to a server with weather text data (default=RTWXTEXT)	
MATch=	'*string*'	lists text products that contain an exact match of the specified string, including any spaces
	string1 . . stringn	lists text products containing all the specified character strings; the strings can occur anywhere in the product
NUM=	number of text products matching the search criteria to list (default=1)	
SOUrce=	*src1 . . srcn*	lists text products with the specified source circuit, for example, IDS, DDS (default=all circuits)
TIME=	*btime etime*	range of times to search; see the Remarks for DAY, TIME and DTIME interactions (*btime* default=current if the DAY requested is the current, *btime* default=23:59:59 for any other day; *etime* default=*btime* if a range is specified in DAY keyword; if a range is not specifed in either DAY or TIME, then *bday*, *btime* and DTIME determine the day and time range)

WMO=	*head1 . . headn*	lists text products matching the specified WMO product headers; must be at least two characters; you cannot use WMO with the APRO keyword; see the Remarks
WSTn=	*stn1 . . stnn*	lists text products with the specified WMO stations; must be four characters, for example, KLAX, KSFO

Remarks

Weather text data must reside on a McIDAS-X remote server; it cannot reside on the local server.

The first line of text products received by McIDAS contains WMO information about the data being received. An example of this WMO information line is shown below:

Example: **FPUS5 KMKE 171245**

Format: *product origin date*

product	the product header; includes a two-character WMO product code (**FP**), a two-character geographic identification code (**US**), and an optional product ID number between 0 and 99 (**5**)
origin	the station initiating the report (**KMKE**)
date	day of the month (**17**), hour UTC (**12**), and minute (**45**)

The WMO keyword matches the product header. The WSTN keyword matches the station initiating the report. The DTIME and TIME keywords use the date header to determine matches within their time range.

You must specify at least two characters for the WMO keyword; the simplest search includes only the WMO product code. You can also use wildcard characters and ranges. For example, to list text products that begin with FPUS and have a product number between 1 and 5, specify WMO=FPUS1-5. To list all WMO products that begin with FP, have any country code, and a product number 5, specify WMO=FP??5. To list all FPUS products that contain a 5 as their last digit, specify WMO=FPUS?5. This entry matches on FPUS5, FPUS15, FPUS25, etc.

Many text products sent by the National Weather Service contain an additional line of information. This AFOS/AWIPS information also indicates product and origin. Examples of the AFOS/AWIPS information lines are shown below.

Examples: **ZFPWI**
LFPMKE

Format: *pppsss*

ppp	the product header; three-character AFOS/AWIPS product code (ZFP or LFP)
sss	the station initiating the report; two or three characters depending on the AFOS product (WI or MKE)

The APRO keyword matches the product header. The ASTN keyword matches the station initiating the report.

Use the APRO, ASTN, WMO and WSTN keywords to significantly decrease your search time. The values specified within each of these keywords are connected with a logical OR. For example, APRO=ZFP SFP matches any product with the AFOS/AWIPS product code ZFP or SFP. The values specified within the MATCH keyword are connected with a logical AND. For example, MATCH= TORNADO WARNING matches any product with the words TORNADO and WARNING.

If you do not specify the TIME and DAY keywords or you specify a single day and/or time with them, then DTIME determines how many hours back in time to search. For example, TIME=12 DAY=340 APR=ZFP uses the default DTIME of 96 and therefore searches from 12 UTC on day 336 through 12 UTC on day 340. Likewise, TIME=12 APRO=ZFP searches backward 96 hours from 12 UTC today. Use the NUM keyword to list more than the first (most recent) match.

If you specify a range of times and/or days with the TIME and/or DAY keywords, then you cannot use the DTIME keyword. For example, DAY=340 TIME=12 18 APRO=ZFP ignores the DTIME default of 96 hours and searches only between 12 and 18 UTC on day 340. Likewise, DAY=305 306 TIME=12 APRO=ZFP also ignores the DTIME default and searches from 12 UTC on day 305 through 12 UTC on day 306. And, DAY=300 306 TIME=21 3 APRO=ZFP searches from 21 UTC on day 300 through 03 UTC on day 306.

Output lines that contain the string(s) specified in the MATCH keyword are highlighted in bright white by default. The highlight color can be changed by specifying **-tcolfg15 '***color* **'** in your **.mcidasrc** file, where *color* is is the highlight color.

The predefined text products are defined in the files **WXUSER.PRD**, **WXSITE.PRD** and **WXCORE.PRD**. When you request a predefined product, WXTLIST first looks for it in these files on your workstation. If the product is not found on your workstation, the request is sent to the server. If the server cannot find the requested product, you see an error message.

To set up your own predefined products, create the file **WXUSER.PRD** in any directory accessible from your McIDAS session. Use this file to create names for specific search conditions you use frequently or to add search conditions to existing products.

The format of a line in a personal product file and a sample file are shown below:

product_name WXTLIST_keywords

```
# ------------ Personal Product File ------------ MSN_NOWCAST APRO=NOW ASTN=MKE
MATCH=MADISON DANE_ZONE PROD=ZONE_FCST ASTN=WI MATCH=DANE BADNEWS
PROD=TORNADO_WARNING
```

To add search conditions to a product that is already defined on the server, use the special keyword PROD. Specify PROD as an existing predefined product. You can also use this keyword to give personal alias names to other predefined products.

Use the pound sign (#) in the first column to indicate comment lines. To continue a command on the next line, end the line with a backslash character (\).

Search conditions defined in your personal product file override the keywords you enter in the WXTLIST command. For example, suppose your personal product file contains the following entries:

```
MILWAUKEE WSTN=KMKE ZONE APRO=ZFP
```

If you enter WXTLIST ZONE APRO=LFP, LFP is ignored because the product ZONE is predefined in your personal product file with the search condition APRO=ZFP.

If you enter WXTLIST MILWAUKEE APRO=LFP, WXTLIST lists products matching both WSTN=KMKE and APRO=LFP.

Examples

WXTLIST

This entry lists the last text product received.

WXTLIST WMO=FP NUM=2

This entry lists the two most recent text products with the FP product code in the WMO product header.

WXTLIST WMO=FOUS

This entry lists the most recent text product with the FOUS product/country code in the WMO product header.

WXTLIST WMO=FOUS14

This entry lists the most recent text product with a FOUS14 WMO product header.

WXTLIST WMO=FOX?60-75 NUM=3

This entry lists the three most recent text products with a WMO product header beginning with FOX, containing one additional character, and having a product ID range of 60 to 75.

WXTLIST WMO=FPUS5 FPUS3 WSTN=KMKE NUM=2

This entry lists the two most recent text products sent from Milwaukee (KMKE) with FPUS5 or FPUS3 WMO product headers.

WXTLIST WSTN=KMKE NUM=50 DTIME=2

This entry lists up to 50 of the most recent products issued out of Milwaukee in the last two hours, regardless of the WMO product header.

WXTLIST MATCH=MADISON

This entry lists the most recent text product containing the string MADISON. Because the APRO, ASTN, WMO and WSTN keywords are not used in this command, the default for DTIME is one hour. Only records from the last hour are searched.

WXTLIST WMO=FOUS14 MATCH=MSN

This entry lists the most recent FOUS14 product that also contains the string MSN. Because this product is issued from Washington DC, MSN does not appear as WMO or AFOS station information. To find the Madison forecast, you must search for the string MSN.

WXTLIST APRO=ZFP SFP ASTN=MN WI NUM=10

This entry lists the 10 most recent text products that are either AFOS products ZFP or SFP and are issued for AFOS stations MN or WI.

WXTLIST APRO=ZFP MATCH='DENSE FOG'

This entry lists the most recent ZFP product that contains the string DENSE FOG. This must be an exact match: the word DENSE, followed by a space, followed by the word FOG.

WXTLIST APRO=ZFP MATCH=DENSE FOG

This entry lists the most recent ZFP product that contains the strings DENSE and FOG. The two words can occur anywhere in the text product.

WXTLIST TORNADO_WARNING

This entry lists the most recent TORNADO_WARNING product.

WXTLIST ZONE_FCST ASTN=WI

This entry lists the most recent ZONE_FCST product issued for AFOS station WI. Because the station name included with the zone forecast for Wisconsin is WI, it must be included in the ASTN keyword. If you specify MKE, the station issuing the report, no match is found.

X

Tracks cloud drift wind targets with one keystroke.

Remarks

You can use the X key in place of the right mouse button when tracking cloud drift winds. It makes one wind measurement for each pair of frames in the loop bounds without stepping through the frames.

For example, in a loop of three images, you would press the right mouse button four times to make two wind measurements. The X key will make the same measurements by reading the cursor displacement on each frame without advancing through the frames; it's handled through the software so nothing mechanical is seen. This technique is useful if you find consistent flow in a large cluster of clouds.

To use this function,

Press: **Alt X**

Y

Toggles image frames to and from the loop control system.

Remarks

To toggle image frames to and from the loop control system,

Press: **Alt Y**

or

Type: **Y**

Press: **Enter**

Use the Y command only on McIDAS-X sessions with independent graphics. See the TERM command for more information.

The loop control system consists of the A, B, J, L, Y, LB, and LS commands.

You can connect or disconnect the image frames from the loop control keys independently of the graphics frames.

Z

Toggles the McIDAS-X zoom function on and off.

Remarks

Zooming (blowup) provides a more detailed view of a frame's features.

To run the Z command:

Press: **Alt Z**

or

Type: **Z**

Press: **Enter**

Remarks

Use the Z command to toggle between zoomed and normal view states. Use the ZM command to set the zoom factor between 1 to 6. After setting the zoom factor, use the Z command to toggle to the zoomed state.

To roam in the zoomed state, press and hold the middle mouse button. Press the right mouse button to reposition the frame so the upper-left corner of the frame appears in the upper-left corner of the Image Window. If you specified the **-autoResize** flag at start up or in your **.mcidasrc** file, the Image Window is automatically resized to fit the zoomed frame.

ZA

Writes text and/or symbols on the graphics frame.

Format

ZA *color height justify [keywords] "text*

Parameters

color graphics color level; 0 for erase (default=1)

height pixel height to draw text and symbols (default=10 for text; cursor height for TCYC keyword symbol)

justify justification of the text and symbols; specify one of the options below; see the Remarks

LA	left justified above the center of the cursor
LC	left justified at the center of the cursor
LB	left justified below the center of the cursor (default)
CA	center justified above the center of the cursor
CC	center justified at the center of the cursor
CB	center justified below the center of the cursor
RA	right justified above the center of the cursor
RC	right justified at the center of the cursor
RB	right justified below the center of the cursor
M	mouse justified; see the Remarks
N	same as the CC option, but allows numerous plots of the text or symbols by pressing mouse buttons; see the Remarks
V	vertically aligned; the upper-left corner of the first letter or symbol is placed at the cursor center; subsequent text and symbols are written below

"text text to write on the graphics frame; you may use uppercase and/or lowercase letters; quotes are mandatory and may be substituted with curly brackets (**{ }**); see the Remarks for instructions on how to plot weather symbols and arrows; the following placeholders can be specified for text substitution of information from the frame, dataset or displayed image; the output format is indicated in italics, e.g., *nnn*

(FRAME)	frame number, *nnn*
(DATASET)	ADDE dataset name, 24 characters maximum
(POS)	ADDE dataset position, *nnnn*
(BAND)	band number, *nn*
(WL)	wavelength, *nn.nn unit*
(WN)	wave number, *nn.nn unit*
(FREQ)	frequency, *nn.nn unit*

(ID) NEXRAD station ID, 3 characters

(MEMO) memo field, 32 characters maximum

(SS) sensor source number, *nnn*

(STYPE) image type, 12 characters maximum

(ERES) element resolution, *nn.nn*

(LRES) line resolution, *nn.nn*

(RES) displayed resolution of the image, *nn.nn*

(EGEORES) geographical element resolution in km, *n.nn*; see the Remarks

(LGEORES) geographical line resolution in km, *n.nn*; see the Remarks

(GEORES) geographical resolution in km, (*n.nn* X *n.nn* KM); see the Remarks

(DAY) day, *dd mon yy*

(MDY) day, *mon dd yy*

(JDAY) Julian day, *ccyyddd*

(TIME) image time, *hhmmss*

(HH) image time, *hh*

(HHMM) image time, *hh:mm*

(HHMMSS) image time, *hh:mm:ss*

Keywords

Ang= angle to write text; do not use with the FONT command or global keyword, or keywords ENCL and TCYC

Encl=

C draws a circle around the text in the N mode

E draws an ellipse around the text in the N mode

FULlframe=

YES use the entire frame for output, regardless of the panel configuration; do not use FUL=YES with the global keyword PAN

NO limit output to the current panel or the panel specified with the global keyword PAN (default)

Gra= graphics frame number (default=current)

IMA= image frame number to use for determining values of placeholders in the "*text* parameter (default=current)

POS= *lin ele* TV line and element position to draw text, symbols, or arrows (default=current cursor position)

TCYc= *color hemi*

plots a hurricane or tropical storm symbol centered on the cursor or the position specified in the POS keyword; the *justify* paramater is ignored

color graphics color level of the symbol's center

X leave the center blank

hemi determines the direction of spirals

S southern hemisphere (default)

N northern hemisphere

Wid= graphics line width, in pixels; the maximum is 64 (default=current)

Remarks

Use the global keyword FONT to change the font for a single command entry. See *Appendix A, Global Keywords*. Use command FONT to change fonts for all graphics commands.

To write lowercase letters, toggle the Caps Lock key or press the **Shift** key when typing. Press it again before entering the next command, as McIDAS commands are normally entered in uppercase letters.

In the *justify* parameter:

- left justified (LA, LC and LB options) means the text begins at the cursor position and extends to the right
- right justified (RA, RC and RB options) means the text ends at the cursor position and extends to the left
- center justified (CA, CC, CB and N options) means the text is horizontally centered at the cursor position so it extends equal distances to the left and right

The (GEORES), (LGEORES) and (EGEORES) placeholders can be used only if GEORES=YES was specified in the IMGDISP command that displayed the image. The resolution values, listed as *n.nn* in the Parameters section above, are in kilometers and are displayed in the format *0.nn*, *n.nn*, *nn.n* or *nnn* (depending on the magnitude of the resolution). The two resolution values displayed with the (GEORES) placeholder are the same as (LGEORES) and (EGEORES), in that order.

When you specify the M option for the *justify* parameter, the position of the mouse and the clicking of its buttons control the location of the cursor text. Clicking the middle mouse button repositions the start of the text at the cursor location. Clicking the right mouse button repositions the end of the text at the cursor location. The height is scaled as needed. Using the FONT command or global keyword with the M option positions the text horizontally at various sizes. You can't change the angle of the text.

If you specify the N option for the *justify* parameter, clicking the middle mouse button centers the text at the cursor location. Clicking the right mouse button erases the last drawn text.

To plot weather symbols and arrows, enter a dollar sign ($) before any of the alphanumeric characters shown below.

T - thunderstorm		C - clear		0 - arrow	
W - rain shower		Q - partly cloudy		1 - arrow	
X - snow shower		B - mostly cloudy		2 - arrow	
R - rain		O - overcast		3 - arrow	
S - snow		D - drizzle		4 - arrow	
Z - freezing rain		H - haze		5 - arrow	
		F - fog		6 - arrow	
		$$ - dollar sign	$	7 - arrow	

Tropical storm and hurricane symbols are plotted at the center of the cursor, regardless of the justification specified.

The practical maximum number of characters for keyword ENCL is three.

Examples

ZA "Wisconsin

This entry writes the word *Wisconsin* in graphics color level 1 on the current graphics frame. The letters are 10 pixels high and are left justified below the center of the cursor.

ZA 3 15 LA "McIDAS

This entry writes the word *McIDAS* in graphics color level 3 on the current graphics frame. The letters are 15 pixels high and are left justified above the center of the cursor.

ZA 2 20 CC GRA=2 {streamlines}

This entry writes the word *streamlines* in lowercase letters in graphics color level 2 on graphics frame 2. The letters are 20 pixels high and centered on the cursor. Curly brackets are used instead of the double quote.

ZA 4 7 V FONT=HELBO "Nevada

This entry writes the word *Nevada* vertically on the current graphics frame in graphics color level 4 using the bold, oblique Helvetica font. The upper-left corner of the N is at the cursor center. The letters are seven pixels high.

ZA 4 20 LC ANG=45 "Green Mountains

This entry writes *Green Mountains* at a 45° angle in graphics color level 4. The letters are 20 pixels high and left-justified in relation to the cursor.

ZA 5 12 LB FONT=COUBR "GOES-12 Band (BAND) ((WL)) from (DAY) at (HHMM) UTC

This entry writes the specified text in graphics color level 5 using a bold Courier font. The characters are 12 pixels high and are left justified below the center of the cursor. The placeholders (BAND), (WL), (DAY) and (HHMM) are replaced with the corresponding values from the displayed image.

ZLM

Draws graphics using the mouse.

Formats

ZLM BOX *color height width [keywords]*
ZLM CIRCLE *color height width [keywords]*
ZLM DRAW *color* **SMOOTH=**
ZLM FILL *color height width [keywords]*
ZLM FREE *color*

Parameters

BOX	draws a box
CIRCLE	draws a circle or ellipse
DRAW	draws a smooth, continuous line as you move the mouse
FILL	draws and fills a box
FREE	draws a straight line between points selected with the mouse (default)
color	graphics color level (default=3)
	0 erases graphics when used with the FILL option
height	height of the box, circle or ellipse, in pixels (default=cursor height)
width	width of the box, circle or ellipse, in pixels (default=cursor width)

Keywords

DASh=	**YES**	draw dashed lines for BOX, CIRCLE, DRAW or FREE option
	NO	draw solid lines (default)
GRA=	graphics frame number to display output; valid only with MODE=N and BOX, CIRCLE, or FILL option (default=current)	
MODe=	**I**	interactive mode (default)
	N	non-interactive "batch" mode; draws shape centered at line and element position specified in POS keyword on the graphics frame specified with the GRA keyword; valid only with BOX, CIRCLE, or FILL option
POS=	*line ele*	TV line and element position to center the shape on; valid only with MODE=N and BOX, CIRCLE, or FILL option (default=current cursor position)
SMOoth=	number of samples to take for the running average; use with the DRAW option; the range is 1-50; larger numbers produce smoother lines (default=3)	

Examples

ZLM

This entry permits straight-line freehand drawing in graphics color level 3. The response is:

```
Move mouse to position the cursor on the image frame. Press the MIDDLE mouse button
to select point for option. Press the RIGHT mouse button to end segment. Press
MIDDLE AND RIGHT mouse buttons or type Alt+Q to end ZLM.
```

ZLM BOX

This entry displays the response below on the screen.

```
Move mouse to position the cursor on the image frame. Press the RIGHT mouse button
to draw the box. Press MIDDLE AND RIGHT mouse buttons or type Alt+Q to end ZLM.
```

ZLM DRAW 4 SMOOTH=10

This entry displays the response below on the screen.

```
Move mouse to position cursor on the image frame. Press MIDDLE mouse button and
hold, while moving the cursor wherever on the screen you want to draw. Press RIGHT
mouse button to end segment. Press MIDDLE AND RIGHT mouse buttons or type Alt+Q to
end ZLM.
```

Following the above directions produces a smooth line drawing in graphics color level 4 with a running average of 10.

ZLM BOX 7 25 630 MODE=N POS=450 320 GRA=5

This entry draws a 25- by 630-pixel box in graphics color level 7 centered at TV coordinate 450,320 on graphics frame 5.

ZM

Sets the zoom factor to use with the Z command.

Format

ZM *zoom*

Parameter

zoom zoom factor for the currently displayed image; the range is 2 to 6 (default=list current zoom factor)

Remarks

If you specified the **-autoresize** flag at startup or in your **.mcidasrc** file, the Image Window is automatically resized to fit the zoomed frame.

See Also

Alt Z toggles the zoom function on and off; see the Z command.

Examples

ZM

This entry lists the current zoom factor.

ZM 2

This entry sets the zoom factor to 2.

Chapter 4

McBASI Interpreter

This chapter contains an introduction to McBASI; it consists of:

- McBASI Interpreter Commands and Reserved Words
- Functions
- McBASI Sample Programs

The name McBASI was first coined in 1985 on McIDAS-MVS to describe a line oriented editor and online BASIC language interpreter within the mainframe McIDAS environment. In McIDAS-X, only the interpreter (the RUN command) has survived.

McBASI programs are presented as ASCII text files created by an editor that stores text as ASCII text with a new line character at the end of each line. The line of text for McBASI programs can not be longer than 80 characters. These files are stored in the **$HOME/mcidas/data** directory, or a REDIRECT or **MCPATH** directory with the extension .MCB. Commands may be in uppercase or lowercase; line numbers are only needed on program lines referenced as targets of GOTO or GOSUB statements.

McBASI Interpreter Commands and Reserved Words

Use the following McBASI interpreter commands and reserved words to create and run a McBASI program.

They are listed alphabetically. Each description contains the command's function, syntax and one or more examples.

- CLOSE
- GOSUB
- GOTO
- IF
- INPUT
- KEYIN
- LET
- OPEN
- PLOT
- POKE
- PRINT
- PTABLE
- REM
- RETURN
- RUN
- SLEEP
- STOP

CLOSE

Frees the I/O table entry for the specified *unit* (text file).

Syntax

CLOSE *unit*

Parameter

unit any valid numeric expression associated with a text file by a previous OPEN command

---

Remarks

A maximum of 10 files are active at once.

Example

CLOSE 10

For this example, 10 is the *unit* number associated with a text file by a prior OPEN command.

GOSUB

Changes the normal downward flow of control.

Syntax

GOSUB *x*

Parameter

x any valid numeric expression

Remark

This command is like GOTO, except control may be returned to the statement following the GOSUB by executing a RETURN statement.

Example

```
GOSUB 1000
      100   REM Return Here
      .
      .
      1000 PRINT B
      1010 RETURN
```

In this example, control is returned to line 100.

GOTO

Changes the downward, line-by-line flow of control.

Syntax

GOTO *x*

Parameter

x is any valid numeric expression

Examples

GOTO 300

GOTO 13+6*X

IF

Compares expressions, using relational operators, to determine whether to execute a statement.

Syntax

IF *expr relop expr statement*

Parameters

expr	any valid expression	
relop	any relational operator	
	=	means equal
	< > or > <	means not equal
	>	means greater than
	<	means less than
	>= or =>	means greater than or equal to
	<= or =<	means less than or equal to
statement	step to be executed if the relationship is true	

Remarks

If several *statements* are used, either all are done (if the relationship is true) or none are done (if it is false). To compare character strings, use only the = and < > relational operators. If *statement* is a let statement (a replacement statement) use the command LET.

Example

IF X=8 LET Y=13:LET J=9:GOTO 890

If X=8 is true, both LET *statements* and the GOTO *statement* are executed.

INPUT

Allows data to be gathered from an LW file and the RUN command line.

LW File Data Gathering

Syntax

INPUT *@u(r), A$*

Parameters

u numeric expression for the unit number; @ is required

(r) numeric expression used to specify a record number; record numbers begin with zero

A$ variable into which the record is read

Remarks

The input statement for LW files is restricted; only one character variable may be designated in the list and the record from the file will be read into this variable. This form of INPUT is useful for reading text files. Refer to the OPEN statement regarding fixed length and variable length records. The INPUT statement in this form handles both; the *(r)* record number is available for either format.

Example

INPUT @3, A$

IF MID$(A$,1,4)=EOF$ GOTO *xxx*

This example tests for an end-of-file, using the pseudovariable EOF$.

Run Command Line Data

Syntax

INPUT *a,b,c*

Parameters

a,b,c variables

Remarks

Values are taken from the RUN command line as positional parameters of a normal McIDAS command. If the named variable is numeric, it is picked up as a number. If the variable is a character string, the value is picked up as a character parameter (up to 12 characters, no blanks). Each INPUT statement of this form picks up the next positional parameter. Thus, these statements are equivalent to the single INPUT statement above:

INPUT *a*

INPUT *b*

INPUT *c*

If the RUN command line contains a quoted string (for example, "THIS IS A QUOTED STRING), it may be read by using a character variable. The first character in the variable is a double quote (") to indicate it was picked up from the quoted string. An attempt to input a parameter which is not present results in either a numeric value of zero or a character string of length zero.

KEYIN

Submits a McIDAS command to be run.

Syntax

KEYIN *list*

Parameter

list zero or more expressions; place a comma or semicolon between expressions; commas tabulate to a print position (column) that is a multiple of 16; semicolons allow no spaces to be inserted; do not end the *list* with a semicolon or comma unless you don't want the statement to execute until you've entered another KEYIN statement that does not terminate with a semicolon

---

Remark

The exit status from the KEYIN command is placed in a pseudovariable named STATUS. A nonzero value usually indicates failure of the command.

Example

```
LET A$="KMSN"
LET T= 6
KEYIN "SFCLIST ";A$;" ";T
IF STATUS < > 0 PRINT "Something went wrong"
```

This example runs the command SFCLIST KMSN 6. If the command fails, the message *Something went wrong* is displayed.

LET

Creates a replacement statement in a McBASI program.

Syntax

LET *var*=*expr*

Parameters

var variable for numeric quantities; it is a single letter or a single letter followed by a digit; numeric variables are REAL*4 quantities; for character strings, A$ through Z$ are available and may contain up to 255 characters each

expr any valid numeric or character expression; numeric expressions may contain the operators +,-,*, /,**, with normal precedence rules; for character or string expressions, *expr* may be a string variable, a string constant enclosed in quotes ("ABC"), a concatenation of two strings with the + operator, or a string function, e.g., pseudovariable TIME$

Remarks

The hexadecimal constant $*nnnn* is available, e.g., $FE equals 254. A single array is also available; one thousand words are reserved and referenced by following a numeric variable name with a parenthetical expression. The value of this expression is added to the value of the variable to obtain an index into this array space.

Example

LET A=0
LET B=A(13) References the 13th word
LET C=10
LET B=C(13) References the 23rd word (10 + 13)

Do not mix numeric and character expressions. The VAL function converts character to numeric; STR$ converts numeric to character. In most cases, the LET may be omitted.

OPEN

Logically equates a *unit* number with the name of a text file for INPUT and OUTPUT.

Syntax

OPEN *filename, unit, reclength*

Parameters

filename	valid string expression, less than 13 characters
unit	valid numeric expression
reclength	valid numeric expression; only useful when dealing with old-style or otherwise fixed lengthtext files; must be specified to randomly write(PRINT) data into a file; otherwise, writing is sequential; should be 80 when dealing with oldstyle McIDAS text files, including those created by specifying a DEV=F in McIDAS commands

---

Remarks

When an OPEN statement is executed, the record pointers for input and output are set to point to the first record of the file (record zero). If these options are omitted from the INPUT or PRINT statements, the next record is used.

Example

OPEN "YYEXPLE",10

PLOT

Draws lines on the image/graphic display.

Syntax

PLOT *line, element, color, frame, width*

Parameters

line	frame line coordinate value (no default)
element	frame element coordinate value (no default)
color	graphics color level (no default)
frame	graphics frame to write on (default=current)
width	line width (default=1)

Remark

Each PLOT command draws a line from the previous location to the specified coordinates. Specifying zero for *color* does not draw a line although it does change the location.

Examples

PLOT 100, 200, 0

This example establishes the drawing position at line 100, element 200 on the current frame.

PLOT A, B, 1, 3

This example draws a line from the previous position to the line/element coordinates specified in the variables A and B, respectively. The line is drawn in graphics color level 1 on frame 3.

PLOT A, A+3, 2

This example draws a line in graphics color level 2 from the previous position to a new line coordinate in the variable A, and an element coordinate equal to A plus 3.

PLOT 1, 1, 0

This example moves the starting coordinate to the upper-left corner of the frame.

POKE

Forces a value into a User Common location.

Syntax

POKE *value, location*

Parameters

value	numeric or string expression; if numeric, it is converted to an integer quantity unless a string is specified
location	numeric expression identifying the word in User Common to be written into

---

Remark

See also functions PEEK and PEEK$.

Caution !
Be careful when using this command. It is possible to set your computer into a state from which the only recovery is to boot.

PRINT

Displays command text on your workstation.

Syntax

PRINT *list*

Parameter

list is zero or more expressions; place a comma or semicolon between expressions; commas tabulate to a print position (column) that is a multiple of 16; semicolons allow no spaces to be inserted; do not end the *list* with a semicolon or comma unless you don't want the statement to execute until a future PRINT or KEYIN statement is executed

---

Remark

When developing programs that generate commands, use the PRINT statement for testing. It lets you to see the command text on your workstation. The PRINT statement is identical in structure to the KEYIN statement.

Examples

```
LET X=123
LET A$= "KMSN"
LET T= 12.0
PRINT "this is a test ";X;
KEYIN "SFCLIST ";
PRINT " of the ",
KEYIN A$;
PRINT " print statement"
KEYIN " ";T
```

This example produces the following print line and submits the command:
SFCLIST KMSN 12.

```
this is a test 123 of the print statement
```

```
? "A=";A
PRINT "A=";A
```

The question mark (?) may be used in place of the PRINT statement. For example, the two statements above are equivalent:

Printing to a Text File

Output may be directed to a text file in either fixed length or variable length (text) format. This file-directed form of the PRINT statement has the following syntax.

Syntax

PRINT *@u(r),list*

Parameters

u numeric expression for the unit number; @ is required

r numeric expression used to specify the record number (begins with zero); parentheses are required; may only be specified for files with

fixed length records (see the OPEN statement); variable length files are sequentially written; if the *r* clause is omitted, the next record is assumed (record following the last one written); the comma is required and followed by a list of variables, constants, etc.; if a PRINT statement ends with a semicolon or comma (meaning *do not print this yet*), the *@u(r)* clause is ignored

list is zero or more expressions; place a comma or semicolon between expressions; commas tabulate to a print position (column) that is a multiple of 16; semicolons allow no spaces to be inserted; do not end the *list* with a semicolon or comma unless you don't want the statement to execute until a future PRINT or KEYIN statement is executed

---

Example

PRINT @4,EOF$

This is a final PRINT statement that outputs a logical end-of-file. Use it for a sequential file that will be read by another McIDAS program.

PTABLE

Places the value of *stringvalue* in the string table entry named *stringname*.

Syntax

PTABLE *stringvalue*, *stringname*

Parameters

stringvalue	valid string expression containing the value to put into the McIDAS string named *stringname*
stringname	valid string expression containing the name of the McIDAS string; less than 13 characters

---

Remark

See the McBASI TABLE$ function.

Example

PTABLE MID$(TIME$,1,2),"HR"

This example sets the value of the McIDAS string HR to the current hour.

REM

Adds comments or inserts a help into your program.

Syntax

REM *comment*

Remark

Remarks (REMs) do not affect the flow of the program.

Examples

Use one of these examples to begin a help line. The normal McIDAS HELP facility can then be employed to list the help for this program.

REM ?
REM?

RETURN

Returns from a subroutine call.

Syntax

RETURN

Remark

Also see the McBASI command GOSUB.

RUN

Runs the McBASI program.

Syntax

RUN *p1 . . . pn* **FILE= CLEAR= VAR=**
'*filename p1 . . . pn* **CLEAR= VAR=**
RUN "*statement*
' "*statement*

Parameters

p1 . . n		parameter values
filename		name of the McBASI program to run; the extension defaults to .MCB
statement		McBASI statement to execute immediately; 80 characters maximum
FILE=		runs a program from a different file; invalid when using the single quote format (')
CLEAR=	**YES**	suppresses initialization of all variables
	NO	variables retain previous values
VAR=		variable's source or destination LW file
	SHARE	shares variables when one McBASI program RUNs another
	x	LW file in which to store the variables (different from the default, which is assigned by terminal number), or the LW file holding the source variables

---

Remark

The RUN statement can be abbreviated; see the examples below.

Examples

RUN *p1 . . . pn* FILE=*xxxxxx*

'*xxxxxx p1 . . . pn*

Replace *xxxxxx* with the file name (limited to 6 characters); these two RUN statements are interchangeable.

SLEEP

Suspends the running of your McBASI program for a specified time.

Syntax

SLEEP *seconds*

Parameter

seconds constant, variable, or expression containing the number of seconds (including decimal fractions) to suspend the program

---

Examples

SLEEP .5

SLEEP 10

SLEEP C

SLEEP 6.2

STOP

Stops your program.

Syntax

STOP
STOP "*Message*"

Remark

Your program also stops by "falling off the end," completing the last line of a program. If you don't provide a quoted message, a line is displayed indicating the program line or statement number where the STOP occurred.

Functions

The following functions are available. *x* and *y* refer to numeric expressions; *a$* and *b$* refer to character expressions.

Function	**Meaning**
SIN(*x*), COS(*x*), TAN(*x*)	trigonometric functions; *x* is an angle in radians;
ASIN(*y*), ACOS(*y*), ATAN(*y*)	the arc functions return values in radians
EXP(*x*)	exponential
LN(*x*)	natural logarithm
LOG(*x*)	base ten logarithm
ABS(*x*)	absolute value of *x*
INT(*x*)	integer part (truncated value) of *x*
NINT(*x*)	nearest integer value to *x*
MOD(*x,y*)	the remainder resulting from *x/y*
LEN(*a$*)	length of the string in *a$*
INDEX(*a$,b$*)	location in *a$* string of *b$* substring; if *b$* is not in *a$*, zero is returned
VAL(*a$*)	numeric value of string *a$*
SQRT(*x*)	square root of *x*
STR$(*x*)	calculates the string corresponding to the number *x* using McIDAS function CFG for the conversion
MID$(*a$,x,y*)	extracts a string of length *y* from *a$*, starting at character position *x* (first position = 1)
TIME$	a pseudostring variable of length 6 which has the current time in the format HH:MM:SS
DATE$	a pseudostring variable of length 7 which has the current date in the format CCYYDDD
DAY$	a pseudostring variable of length 6 which has the current date in the format YYMMDD
TFILA$ . . TFILE$	a pseudostring variable of length 9; a temporary LW file named during execution of a BASIC program; it is deleted when the program terminates; five temporary LW files are available per program execution EOF$ a pseudostring variable; contains the universal value Z80808080
PEEK(*x*)	returns a value from User Common (UC) location *x*; the value in UC is assumed integer and is converted to REAL*4
PEEK$(*x*)	returns a character string
BIN(*string$*)	the real number equivalent of the binary integer value in *string$*; assumes the first four bytes are an integer value (in binary)
BIN$(*x*)	returns an ASCII string equivalent to the integer value of *x*
CHR$(*x*)	returns a string of length one which is equivalent to the ASCII code value *x*
KSYS(*x*)	returns the value from the System KeyTable location *x*; the value in SYSKEY is assumed to be an integer and is converted to REAL*4
KSYS$(*x*)	returns the value from the System KeyTable location *x*; the value in SYSKEY is assumed to be a character string of length 4
TABLE$(*a$*)	returns the value of the string stored in the current string table under the name *a$*
STATUS	numeric pseudovariable set to the value of the exit status each time a KEYIN is performed; a nonzero status usually indicates failure

McBASI Sample Programs

1. In this program, parameters are read from the RUN command until the list is exhausted. DEV=CNN suppresses the normal DONE messages from SFCLIST and TAFRPT. This sends the standard output to the CRT, but does not send the error or debug output. Use RUN KMSN KMKE KCLE KBOS KDCA or RUN KMEM KORD for this example. A step-by-step practice session with this example follows.

```
 10     INPUT A$
        IF LEN(A$)< 4 GOTO 100
        KEYIN "SFCLIST ";A$;" 12
        KEYIN "TAFRPT ";A$;" 12
        GOTO 10
 100    REM FINISHED. . .
```

2. This program illustrates a method of looping. The NINT function ensures that B will at some time be exactly equal to 101. No parameters are necessary in the RUN statement.

```
        let a=0
        let b=0
 30     let a=a+b
        let b=b+1
        if nint(b)<>101 goto 30
        print a
```

3. In this example, the two character variables are read from the RUN statement. If they are identical, line 20 causes EQUAL to be displayed on the CRT and the program stops at line 100. If they are not equal, line 30 causes NOT EQUAL to be displayed, and lines 40 and 50 use the INDEX function to find out if one is a subset of the other. The semicolons between PRINT list items are omitted, as they are only required to prevent syntax ambiguities.

For example, **RUN ABC ABC** produces EQUAL on the CRT while **RUN ABC DEFABCGHI** produces NOT EQUAL plus BUT ABC IS WITHIN DEFABCGHI.

```
        INPUT A$,B$
        IF A$=B$ PRINT "EQUAL": GOTO 100
        IF A$< >B$ PRINT "NOT EQUAL"
        IF INDEX(A$,B$)< >0 PRINT " BUT
        "B$" IS WITHIN "A$
        IF INDEX(B$,A$)< >0 PRINT " BUT
        "A$" IS WITHIN "B$
 100    STOP
```

4. This program copies the current day's 0 or 12 UTC mandatory RAOB MD file data to a workstation MD file. If the current hour is between 0 and 11, the 0 UTC RAOB data is copied to the dataset position equal to the value of the current three-digit Julian day plus 1000. If the current hour is between 12 and 23, the 12 UTC RAOB data is copied to the dataset position equal to the value of the current three-digit Julian day plus 2000.

If this program is run from the command scheduler every 12 hours for an entire year, the 0 UTC data for days 1 through 365 are placed in positions 1001 through 1365, and the 12 UTC data is placed in positions 2001 through 2365.

```
        REM  RAOBDATA.MCB
        H=VAL(MID$(TIME$,1,2)):REM  Hour
        J=VAL(MID$(DATE$,3,3)):REM  Julian Day
        PRINT "  "
        REM
        REM  Determining which point file time to request
        IF H<=23 LET T=12
        IF H<=11 LET T=00
        IF T=00 GOSUB 100
        IF T=12 GOSUB 200
 100    LET F=J+1000
        PRINT "Destination Position: ";F;" Time: ";T
        REM
        GOTO 500
 200    LET F=J+2000
        PRINT "Destination Position: ";F;" Time: ";T
        GOTO 500
        REM  Constructing command to enter...
 500    KEYIN "PTCOPY RTPTSRC/UPPERMAND M/M.";F;"
        SEL='TIME ";T;";CO US'"
        PRINT "  "
        STOP "*** COMMAND SENT ***"
```

The command RUN FILE=RAOBDATA.MCB run at 19:20 UTC on Julian day 98019 generates the output below.

```
Destination Position: 2019 TIME: 12 PTCOPY RTPTSRC/UPPERMAND M/M.2019 SEL='TIME
12;CO US' *** COMMAND SENT ***
```

Chapter 5

F Key Menu System

The F Key Menu System uses menu files to predefine function keys with user commands and format display screens with menus. It is useful for creating command menus to simplify the user interface. This chapter describes how to create and test menu files.

This chapter consists of these sections:

- Creating a Menu File
- Pseudocommands
- Defining Loops
- Sample Menu File

The function key combinations described in this chapter work correctly only if the workstation is configured according to the instructions in the *Configuring a New User Account*.

Menus are displayed in a separate window. The text content, color and format are completely flexible. To select a menu item, simply press one of the function keys (F1 through F12) on the workstation keyboard. Use these 12 function keys alone or with one of the **Ctrl**, **Shift** or **Alt** keys, giving 48 combinations. You can also define the Esc key, allowing each menu to contain up to 48 items.

Each menu file can contain up to 1000 menus. Only one menu is displayed at a time. Menus may be nested, displaying a different menu (a submenu) in the window. Typically, a Main Menu refers to various submenus dealing with a cluster of related functions.

A menu file is an ordinary ASCII text file with a carriage return at the end of each line. Menu files reside in the **$HOME/mcidas/data** directory, or a REDIRECT or **MCPATH** directory. Sample menu files are provided at the end of this chapter; they are helpful for working through the following sections.

Starting and Debugging the F Key Menu System

Use the MCMENU command to start the menu system. A new window running the specified menu appears. For more information, see the MCMENU command in Chapter 3, *Commands*.

ASK1 and the ASK User Interface

The ASK1 command lets you prompt the user for information that can be used in a menu. See Chapter 3, *Commands*, for more information.

Creating a Menu File

A menu file defines both the menu text and commands associated with the function key combinations. It is an ASCII text file containing instructions for the McIDAS menu system. These instructions obey certain format rules: the instructions need specific character positions. The first character in an instruction line defines the type of instruction. These characters include the following.

Character	Instruction
M	Menu or Window defines the menu number described by subsequent instructions.
I	Initialization defines a McIDAS command sequence to run when the menu is accessed.
T	Text defines the screen row and column where a string of text begins, the color and the text itself.
F	The function key defines a McIDAS command or sequence of commands to run when a specified function key is pressed.
A	**Alt** + Function key defines a McIDAS command or sequence of commands to run when you press **Alt** and the specified function key simultaneously.
C	**Ctrl** + Function key defines a McIDAS command or sequence of commands to run when you press **Ctrl** and the specified function key.
S	**Shift** + Function key defines a McIDAS command or sequence of commands to run when you press **Shift** and the function key simultaneously.
U	User Common Pokes define values to poke into User Common whenever this menu is displayed.
+	+ means a line is continued from the previous line (for A, C, F, I, S and T characters only).

If the first character in an instruction line is not one of those listed above, the system treats the line as a comment line and ignores it. Each character is described in more detail below, including a character definition, format, parameter descriptions and example.

M

Defines the menu number.

Format

M *number*

Parameter

number number of the menu being defined; the range is 0 to 9999

Remarks

All menu file instructions following an M instruction refer to the specified menu (window) until the system encounters another M instruction or reaches the end of the template file. Always define menu zero in any menu file.

You can have up to 1000 menus in a menu file.

Example

M 7

This entry defines menu 7 using all subsequent instructions up to the next M or W instruction.

I

Defines the commands to run each time the menu is invoked.

Format

I "*commands*

Parameter

"*commands* series of McIDAS commands to run when the menu is invoked

---

Remarks

The I line is usually placed at the beginning of the Menu definition.

Examples

Below are some useful McIDAS commands.

IGU SET 1

This entry sets the grid file number to 1, which is the default.

MDU SET 14

This entry sets the default MD file number to 14.

LS 1- 4; DR 5 1 1 1

This entry sets the loop bounds to frames 1 through 4 and defines dwell rates.

I "IGU SET 1; MDU SET 14; LS 1- 4; DR 5 1 1 1

This entry runs the three command lines above each time this menu is displayed.

T

Defines the row and column where a string of text will begin, the text color and the text itself.

Format

T *row column color "text*

Parameters

row	screen row number; the range is 0 to 21
column	screen column number; the range is 0 to 79
color	color number; the range is 0 to 15 (see below)
"text	text to display starting at *row* and *column*

Remarks

Row and column numbers are zero based; the upper-left corner of the screen is row 0, column 0. The McIDAS color numbers are:

Number	Color	Number	Color
0	blue	8	dark gray
1	black	9	bright blue
2	green	10	bright green
3	cyan	11	bright cyan
4	red	12	bright red
5	magenta	13	bright magenta
6	brown	14	yellow
7	gray	15	white

Text is displayed as it appears in columns 8 to 80. Leading blanks are significant; trailing blanks aren't. Leave at least 1 character between columns 8 to 80. Blank lines are not allowed. Don't use rows 23 and 24.

Example

T 13 21 15 "Radar Loop

This entry displays the words Radar Loop on row 13, starting in column 21, with white characters.

F

Defines a command or sequence of commands to run when you press a function key.

Format

F *key* "*command*

Parameters

key	function key number; the range is 0 to 12
"*command*	McIDAS commands to run separated by semicolons, or a string in the form @*nnn* (which means switch to menu *nnn*)

Remarks

A function key definition pertains only to a given menu. Since you can redefine function key F1 in each menu, the commands that are run by pressing F1 depend on the currently displayed menu. For example, if you press F1 when menu 7 is displayed, the commands run are those defined for F1 in menu 7.

The function key zero (00) is a special case, and actually refers to the Esc key. Use it only when redefining the Main Menu number. The default for the *F0* or Esc key is zero. If it is not defined, menu zero is displayed when you press the Esc key.

Examples

F 1 "IMGDISP GOES/EAST 2 STA=KMSN

This entry runs the command IMGDISP GOES/EAST 2 STA=KMSN when you press F1.

F 0 "@15

This entry displays menu 15 when you press the Esc key. Define the F0 key in each menu, as the default is zero. Menu 0 is displayed unless F0 is explicitly defined in the currently displayed menu.

A

Defines a command or sequence of commands to run when you press a function key and the Alt key.

Format

A *key* "*command*

Parameters

key	function key number; the range is 1 to 12
"*command*	commands to run separated by semicolons

Example

A 4 "PTDISP PT/ISFC SELECT='TIME 12' PAR=T[F] MAP=MID

This entry runs the PTDISP command when you press the **F4** and **Alt** keys simultaneously.

C

Defines a command or sequence of commands to run when you press a function key and the Ctrl key.

Format

C *key* "*command*

Parameters

key	function key number; the range is 1 to 12
"*command*	commands to run separated by semicolons

Example

C 5 "@83

This entry displays menu 83 when you press the **F5** and **Ctrl** keys simultaneously.

S

Defines a command or sequence of commands to run when you press a function key and the Shift key.

Format

S *key* "*command*

Parameters

key	function key number; the range is 1 to 12
"*command*	commands to run separated by semicolons

Example

S 3 "EG

This entry runs the EG command when you simultaneously press **F3** and **Shift**.

U

Defines values to poke into User Common.

Format

U *number value*

Parameters

number	User Common word number
value	value to poke

Remark

The maximum number of U commands allowed per menu is 36.

Example

U 51 3

This entry pokes the value 3 in User Common word 51 and displays frame 3 in the image window. This entry is more efficient than the UCU POKE command entry as it is performed directly by the menu subsystem.

+

Continues text from the previous line.

Format

+ *"text*

Parameter

"text text continued from the previous line; the double quote is mandatory

---

Remarks

The continuation line (starting in column 2) is concatenated to the previous line, following the last nonblank character of that line.

Be careful with McIDAS commands and parameters that require blank spaces. See the examples below.

Examples

F 3 "LB 1 5; IMGDISP EAST/CONUS 1
+ "STA=CLE

This entry generates the commands LB 1 5 and IMGDISP EAST/CONUS 1STA=CLE, which is an error, since 1STA=CLE is not a legal value. Instead, enter:

F 3 "LB 1 5; IMGDISP EAST/CONUS 1
+ " STA=CLE

This entry places a space between " and STA=CLE. Consequently, it generates LB 1 5 and IMGDISP EAST/CONUS 1 STA=CLE as desired.

Pseudocommands

The F Key Menu System allows for some rudimentary flow control by permitting pseudocommands in the menu file. The pseudocommand syntax allows multiple commands on a line separated by semicolons (;). There are two restrictions:

- @ must always be the last pseudocommand on the line
- * pseudocommands must always precede other normal McIDAS commands in the line, for example:
  F7 "*S02; *R27; GDISP MET/IR STA=EGLL; @09

The F Key Menu System pseudocommands are listed below.

@*nnnn*

This pseudocommand switches to menu *nnnn*. See the Sample Menu File, the instructions for Defining Loops, and the F and C character instructions for more examples of using **@***nnnn*.

Snn*

This pseudocommand saves the current loop state (bounds, rates, etc.) under loop number *nn*. See the the instructions for Defining Loops for more examples of using ***S***nn*.

Rnn*

This pseudocommand recalls the loop state *nn*. See the the instructions for Defining Loops for more examples of using ***R***nn*.

***EXIT**

This pseudocommand prompts McIDAS to perform an EXIT command and removes all McIDAS resident programs from memory. Use this command on rare occasions only.

@RET

This pseudocommand invokes the previous menu. For example, if menu 1 switches to menu 4 using the @4 command, and menu 4 switches to menu 23 using @23, then performing an @RET pseudocommand in menu 23 invokes menu 4 just as if an @4 was done. Consequently, performing an @RET in menu 4 invokes menu 1, not menu 23.

***QUIT**

This pseudocommand exits the menu and closes its window.

Defining Loops

Use the F Key Menu System to define a sequence of frames and their dwell rates as a loop, as shown in the example below. Then use a function key to invoke the loop. Use command DR to set the dwell rates; use command LS or LB to define the loop bounds. Have available an F Key Menu containing the internal instruction *S*nn* for one of the keys. This saves the state of the workstation under the label *nn*. You can do this for up to 99 loops, although there are only 48 possible combinations of F Keys. To quickly recall the loop, define an F Key with the internal command *R*nn*. The only restriction is *R or *S must be the first command on a line.

```
M 0 U 208 0 T 2 20 7 "McIDAS Loop control T 4 10 7 "F1-Go to Loop #1 T 6 10 7 "F2-
Go to Loop #2 T 8 10 7 "F3-Go to Loop #3 T 20 10 7 "Press 0 on numeric keypad to
return to Command Mode F 1 "*R1;@1 F 2 "*R2;@2 F 3 "*R3;@3 M 1 T 2 20 7 "Current
Loop is #1 T 4 10 7 "F1-SAVE current context as Loop #1 T 6 10 7 "F2-Go to Loop #2
T 8 10 7 "F3-Go to Loop #3 T 20 10 7 "Press 0 on numeric keypad to return to
Command Mode F 0 "@1 F 1 "*S1 F 2 "*R2;@2 F 3 "*R3;@3 M 2 T 2 20 7 "Current Loop is
#2 T 4 10 7 "F1-Go to Loop #1 T 6 10 7 "F2-SAVE current context as Loop #2 T 8 10 7
"F3-Go to Loop #3 T 20 10 7 "Press 0 on numeric keypad to return to Command Mode F
0 "@2 F 1 "*R1;@1 F 2 "*S2 F 3 "*R3;@3 M 3 T 2 20 7 "Current Loop is #3 T 4 10 7
"F1-Go to Loop #1 T 6 10 7 "F2-Go to Loop #2 T 8 10 7 "F3-SAVE current context as
Loop #3 T 20 10 7 "Press 0 on numeric keypad to return to Command Mode F 0 "@3 F 1
"*R1;@1 F 2 "*R2;@2 F 3 "*S3
```

Sample Menu File

Below is a sample menu file. Commands are on the left; their functions are on the right. Blank lines are not part of the file.

Sample Menu File

Commands	Function
`M 0`	Defines menu zero.
`F 0 "@1`	The Esc key is aimed at menu 1.
`T 00 00 07 "Esc to enter menu system`	Defines the display for line zero, column zero, using graphics color level 7.
`----------`	Comment; used here to indicate a new menu.
`M 1`	Defines menu 1.
`I "SF 2;LB 2 2`	Displays frame 2, then sets the loop bounds to 2-2.
`T 2 10 14 "F1`	Displays *F1* on line 2, starting in column 10 in graphics color level 14.
`T 2 14 7 "Receive latest satellite image`	Displays this message after the *F1* above.
`T 4 10 14 "F2 T 4 14 7 "Receive current meteorogram T 6 10 14 "F3 T 6 14 7 "See satellite image T 8 10 14 "F4 T 8 14 7 "See meteorogram T 12 10 14 "F5 T 12 14 7 "Return to McIDAS-X`	Displays the rest of the menu choices.
`F 0 "@1`	Redefines the Esc key to go to menu 1.
`F 1 "IMGDISP GOES/EAST 1 STA=KMSN`	Command for F1.
`F 2 "SFCMG KMSN`	Command for F2.
`F 3 "@2 F 4 "@3 F 5 "@4`	F3 takes you to menu 2, etc.
`----------`	Comment; used here to indicate a new menu.
`M 2`	Defines menu 2.
`I "SF 1;LB 1 1`	Displays frame 1, then sets the loop bounds to 1-1.
`T 10 10 7 "Esc to return to menu`	Displays *Esc* message.
`F 0 "@1`	Redefines the Esc key to go to menu 1.
`----------`	Comment; used here to indicate a new menu.
`M 3`	Defines menu 3.
`I "SF 2;LB 2 2`	Displays frame 2, then sets the loop bounds to 2-2.
`T 10 10 7 "Esc to return to menu`	Displays *Esc* message.
`F 0 "@1`	Redefines the Esc key to go to menu 1.
`----------`	Comment; used here to indicate a new menu.
`M 4`	Defines menu 4: return to the normal interface.
`T 10 10 7 "Esc to exit menu system`	Displays *Esc* message.
`F 0 "*QUIT`	Redefines the Esc key to exit the menu.

Appendix A

Global Keywords

You can use the global keywords below with any command.

- DEV - routes text output to a device
- FONT - specifies the font for graphics
- MCCompress - specifies the data compression method
- PAN - specifies the frame panel to use for output or input
- TCOL - specifies text output color
- TWIN - specifies the text frame for output
- VIRT - creates virtual graphics files

DEV

This keyword specifies the destination device of text output generated by a command. The format of the DEV keyword is shown below.

DEV=*device filename file_option*
DEV=*device 'print_command'*

device 1-, 2- or 3-letter string indicating destination for each of the three output types (standard, error, and debug); the first letter directs standard (non-error) output, the second letter directs error output, the third letter directs debug output; do not specify both F and T together, for example, DEV=FTT; the letters below are valid options (default=CCN)

- **C** sends output to the text frame
- **D** sends output to the desktop; valid only on XP workstations; see the last example below
- **F** sends output to an LW file
- **N** suppresses output
- **P** sends output to the default printer
- **T** sends output to a text file

filename the name of the LW or text file for saving output; specify only when *device* includes F or T

file_option

- **A** appends the output to the specified file
- **R** replaces the output in the specified file (default)

'print_command' Unix print command; valid only when *device* includes a P; if *'print_command'* is not specified, then the value of the **`MCPRINT`** environment variable is used; if 'print_command' is not specified and **`MCPRINT`** is not set, then the default **`lpr`** command settings are used

See the FRMSAVE command for information about printing image and graphics output.

Below are six examples of using the DEV keyword.

IMGLIST MY/DATA.ALL DEV=NNN

This entry displays no output text.

IMGLIST MY/DATA.ALL DEV=CCC

This entry directs all text output (standard, error and debug) to the text frame.

IMGLIST MY/DATA.ALL DEV=FFF IMAGES

This entry directs all text output (standard, error, and debug) to the LW file IMAGES.

IMGLIST MY/DATA.ALL DEV=NC

This entry directs error messages to the text frame. Standard and debug messages are suppressed.

IMGLIST MY/DATA.ALL DEV=P 'lpr -Plp10ps'

This entry directs standard output text to the printer named lp10ps.

IMGLIST MY/DATA.ALL DEV=D

This entry is valid only in McIDAS-X on XP workstations. The entry directs standard output text to the desktop. It appears on the desktop as a new icon with the name *IMGLIST Output.txt*. You can manipulate the new icon like other desktop icons. For example, you can double-click it to open the file, drag it to the Printer icon to print its contents, or drag it to the Recycle Bin to remove it from the desktop.

FONT

This keyword specifies the font type for plotting McIDAS graphics. Use the command FONT LIST to list the valid font types. If the requested font size is not available, the closest available font size is used. See the Remarks in the FONT command for details. Below is an example.

ZA 3 20 FONT=HELBO "Using HELBO font

This entry runs the ZA command using a bold, oblique, Helvetica font.

MCCompress

This keyword specifies the data compression method the ADDE remote server should use when sending the requested data to your workstation. The default (if you don't specify this keyword) is to use the method set in the **MCCOMPRESS** environment variable. See *Using Compressed Data Transfers* for more information.

MCC=	**COMPRESS**	requests a compressed data transfer using the Unix **compress** command
	GZIP	requests a compressed data transfer using the Unix **gzip** command
	NONE	requests a data transfer that is not compressed

Below is an example.

IMGDISP GOES/EAST 5 BAND=4 STA=KATL MCC=GZIP

This entry requests that the remote server use **gzip** to compress the specified image data, then send it to the client, which uses **gunzip** to uncompress it then displays it in frame 5.

PAN

This keyword specifies number of the frame panel to display image or graphics output, or to provide navigation and image information. The valid range is determined by how many panels the frame has (set with the PANEL command). The default is the panel where the cursor is when the command is run. Below are four examples.

IMGDISP GOES/VIS PAN=2

This entry displays the image GOES/VIS on panel 2 of the current frame.

MAP USA PAN=4

This entry displays a map of the USA on panel 4 of the current frame.

PC L KMSN PAN=1

This entry moves the cursor into panel 1 of the current frame and positions it over the station KMSN.

IMGPROBE PAN=9

This entry lists digital values for the point in panel 9. The point listed is in the same panel location (as opposed to geographical location) as the cursor in the current panel. For example, if the cursor is positioned at the center of panel 1, this command lists values for the point at the center of panel 9 no matter what the geographical coverage.

TCOL

This keyword defines the text output color. Use the COLORS command to list the 128 valid text and background color combinations. Below is example.

IMGLIST LOCAL/IMAGES.1 100 TCOL=46

This entry runs the IMGLIST command, displaying its output in yellow text on a green background.

TWIN

This keyword specifies the destination text frame to route the command output to. The valid range is 0 through 9. The default is the current text frame. Below is an example.

GU TAB GRAPHIC TWIN=1

This entry runs the GU command, routing the output to text frame number 1.

VIRT

This keyword specifies a virtual graphics number. It applies only to commands that generate graphics. To list your virtual graphics files, use DMAP VIRT.

VIRT=	*number*	**A**	appends the graphics to the specified virtual graphics number, the range is 1 to 9999
	number	**R**	replaces the graphics in the specified virtual graphics number, the range is 1 to 9999 (default)

Graphics are simultaneously drawn on a graphics frame and saved to a designated file. The file name is of the form VIRT*xxxx* where *xxxx* is the number specified in the keyword. Below are three examples.

MAP USA VIRT=1

This entry draws a map of the USA on the graphics frame and writes the output to file VIRT0001, which can later be restored using the SHOWVG command.

MAP USA X LALO VIRT=1 A

This entry draws a map of the USA on the graphics frame, appending it to the output in file VIRT0001. It can later be restored using the SHOWVG command.

MAP MID VIRT=1 R

This entry replaces the USA map above with a map of the Midwest United States.

Appendix B

Julian Day Calendar

The chart below shows the day numbers corresponding to the numbered Greenwich days.

Day of Month	Jan	Feb	March	April	May	June	July	Aug	Sept	Oct	Nov	Dec
1	001	032	060 061	091 092	121 122	152 153	182 183	213 214	244 245	274 275	305 306	335 336
2	002	033	061 062	092 093	122 123	153 154	183 184	214 215	245 246	275 276	306 307	336 337
3	003	034	062 063	093 094	123 124	154 155	184 185	215 216	246 247	276 277	307 308	337 338
4	004	035	063 064	094 095	124 125	155 156	185 186	216 217	247 248	277 278	308 309	338 339
5	005	036	064 065	095 096	125 126	156 157	186 187	217 218	248 249	278 279	309 310	339 340
6	006	037	065 066	096 097	126 127	157 158	187 188	218 219	249 250	279 280	310 311	340 341
7	007	038	066 067	097 098	127 128	158 159	188 189	219 220	250 251	280 281	311 312	341 342
8	008	039	067 068	098 099	128 129	159 160	189 190	220 221	251 252	281 282	312 313	342 343
9	009	040	068 069	099 100	129 130	160 161	190 191	221 222	252 253	282 283	313 314	343 344
10	010	041	069 070	100 101	130 131	161 162	191 192	222 223	253 254	283 284	314 315	344 345
11	011	042	070 071	101 102	131 132	162 163	192 193	223 224	254 255	284 285	315 316	345 346
12	012	043	071 072	102 103	132 133	163 164	193 194	224 225	255 256	285 286	316 317	346 347
13	013	044	072 073	103 104	133 134	164 165	194 195	225 226	256 257	286 287	317 318	347 348
14	014	045	073 074	104 105	134 135	165 166	195 196	226 227	257 258	287 288	318 319	348 349
15	015	046	074 075	105 106	135 136	166 167	196 197	227 228	258 259	288 289	319 320	349 350
16	016	047	075 076	106 107	136 137	167 168	197 198	228 229	259 260	289 290	320 321	350 351
17	017	048	076 077	107 108	137 138	168 169	198 199	229 230	260 261	290 291	321 322	351 352
18	018	049	077 078	108 109	138 139	169 170	199 200	230 231	261 262	291 292	322 323	352 353
19	019	050	078 079	109 110	139 140	170 171	200 201	231 232	262 263	292 293	323 324	353 354
20	020	051	079 080	110 111	140 141	171 172	201 202	232 233	263 264	293 294	324 325	354 355
21	021	052	080 081	111 112	141 142	172 173	202 203	233 234	264 265	294 295	325 326	355 356
22	022	053	081 082	112 113	142 143	173 174	203 204	234 235	265 266	295 296	326 327	356 357
23	023	054	082 083	113 114	143 144	174 175	204 205	235 236	266 267	296 297	327 328	357 358
24	024	055	083 084	114 115	144 145	175 176	205 206	236 237	267 268	297 298	328 329	358 359
25	025	056	084 085	115 116	145 146	176 177	206 207	237 238	268 269	298 299	329 330	359 360
26	026	057	085 086	116 117	146 147	177 178	207 208	238 239	269 270	299 300	330 331	360 361
27	027	058	086 087	117 118	147 148	178 179	208 209	239 240	270 271	300 301	331 332	361 362
28	028	059	087 088	118 119	148 149	179 180	209 210	240 241	271 272	301 302	332 333	362 363
29	029	060	088 089	119 120	149 150	180 181	210 211	241 242	272 273	302 303	333 334	363 364
30	030		089 090	120 121	150 151	181 182	211 212	242 243	273 274	303 304	334 335	364 365
31	031		090 091		151 152		212 213	243 244		304 305		365 366

Appendix C

Sensor Source Numbers

Number	Sensor Source
0	Derived data
1	Test patterns
2	Graphics
3	Miscellaneous
4	PDUS METEOSAT Visible
5	PDUS METEOSAT Infrared
6	PDUS METEOSAT Water Vapor
7	Radar
8	Miscellaneous aircraft data
9	Raw METEOSAT
10	Composite image
12	GMS Visible
13	GMS Infrared
14	ATS 6 Visible
15	ATS 6 Infrared
16	SMS-1 Visible
17	SMS-1 Infrared
18	SMS-2 Visible
19	SMS-2 Infrared
20	GOES-1 Visible
21	GOES-1 Infrared
22	GOES-2 Visible
23	GOES-2 Infrared
24	GOES-3 Visible
25	GOES-3 Infrared
26	GOES-4 Visible (VAS)
27	GOES-4 Infrared and Water Vapor (VAS)
28	GOES-5 Visible (VAS)
29	GOES-5 Infrared and Water Vapor (VAS)
30	GOES-6 Visible
31	GOES-6 Infrared
32	GOES-7 Visible, Block 1 Auxiliary Data
33	GOES-7 Infrared
34	FY-2B
35	FY-2C
36	FY-2D
37	FY-2E
38	FY-2F
39	FY-2G
40	FY-2H
41	TIROS-N (POES)
42	NOAA-6
43	NOAA-7

44	NOAA-8
45	NOAA-9
46	Venus
47	Voyager 1
48	Voyager 2
49	Galileo
50	Hubble Space Telescope
51	Meteosat-8 (MSG-1)
52	Meteosat-9 (MSG-2)
53	Meteosat-10 (MSG-3)
54	Meteosat-3
55	Meteosat-4
56	Meteosat-5
57	Meteosat-6
58	Meteosat-7
60	NOAA-10
61	NOAA-11
62	NOAA-12
63	NOAA-13
64	NOAA-14
65	NOAA-15
66	NOAA-16
67	NOAA-17
68	NOAA-18
69	NOAA-19
70	GOES-8 Imager
71	GOES-8 Sounder
72	GOES-9 Imager
73	GOES-9 Sounder
74	GOES-10 Imager
75	GOES-10 Sounder
76	GOES-11 Imager
77	GOES-11 Sounder
78	GOES-12 Imager
79	GOES-12 Sounder
80	ERBE
82	GMS-4
83	GMS-5
84	MTSAT-1R
85	MTSAT-2
87	DMSP F-8
88	DMSP F-9
89	DMSP F-10
90	DMSP F-11
91	DMSP F-12
92	DMSP F-13
93	DMSP F-14
94	DMSP F-15
95	FY-1B
96	FY-1C

97	FY-1D
101	TERRA-L1B
102	TERRA-CLD
103	TERRA-GEO
104	TERRA-AER
106	TERRA-TOP
107	TERRA-ATM
108	TERRA-GUS
109	TERRA-RET
111	AQUA-L1B
112	AQUA-CLD
113	AQUA-GEO
114	AQUA-AER
116	AQUA-TOP
117	AQUA-ATM
118	AQUA-GUS
119	AQUA-RET
128	TERRA-SST
129	TERRA-LST
138	AQUA-SST
139	AQUA-LST
160	TERRA-NDVI
161	TERRA-CREF
170	AQUA-NDVI
171	AQUA-CREF
180	GOES-13 Imager
181	GOES-13 Sounder
182	GOES-14 Imager
183	GOES-14 Sounder
184	GOES-15 Imager
185	GOES-15 Sounder
186	GOES-16 Imager
195	DMSP F-16
196	DMSP F-17
200	AIRS-L1B
210	AMSR-E L1B
211	AMSR-E RAIN
216	AMSU-A LWP
230	Kalpana-1
240	Metop-A
241	Metop-B
242	Metop-C
250	COMS
300	NPP-VIIRS
400	South Pole Composite
401	North Pole Composite

Appendix D

Bands for Satellite Imagery

The bands for satellite data, which are sometimes called channels, are described in this appendix.

The band numbers in the *Band* columns in this appendix are numbers that are assigned to each central wavelength data output from a specific satellite instrument. We attempt to keep the same numbering scheme in McIDAS, but it is not always possible. Satellites having more than one instrument may have a separate assignment table for each instrument. The information in the other columns (*Central Wavelength, Primary Uses*, etc.) are from official publications and websites devoted to the satellites.

Resolution refers to the maximum resolution of the instrument, which is not always the same as the resolution of the stored data in the area. Definitions for the Terms and Acronyms are located at the back of this appendix.

Operational status and future launch dates are current as of this manual's Revised date. However, launch dates sometimes change so you should check websites of the launch agencies and groups (e.g., http://www.osd.noaa.gov/Spacecraft%20Systems/Spacecraft_Sys/Sat_Systems_Launch_Schedule.html) for up-to-date launch schedules. Another good resource is the WMO Space Programme's Satellite Status page, http://www.wmo.int/pages/prog/sat/satellitestatus.php.

Appendix directory

The following satellite data sources are described:

Satellite Data Source	Orbit Type
GOES Variable Imager	Geostationary
GOES Variable Sounder	Geostationary
GOES AAA	Geostationary
Meteosat	Geostationary
Meteosat Second Generation	Geostationary
GMS	Geostationary
MTSAT	Geostationary
FY-2	Geostationary
Kalpana	Geostationary
COMS	Geostationary
POES	Polar
Metop	Polar
EOS	Polar
FY-1	Polar

-

GOES Variable Imager

Satellites GOES-8 through GOES-11

Orbit Type Geostationary

Instrument Imager

Resolution see table below

Band	Central Wavelength (μm)	Resolution (km)	Primary Uses
1	0.65	1	Visible cloud and surface features
2	3.9	4	Infrared low-level cloud/fog, fire detection
3	6.8	8	Infrared mid-level water vapor
4	10.7	4	Infrared surface/cloud-top temperature
5	12.0	4	Infrared surface/cloud temperature, low-level water vapor

Satellites GOES-12 through GOES-15 (12 is operational at ~60°W; 13 is operational at ~75°W; 14 is in standby at ~89.5°W; 15 is operational at ~135°W)

Orbit Type Geostationary

Instrument Imager

Resolution see table below

Band	Central Wavelength (μm)	Resolution (km)	Primary Uses
1	0.62 - 0.65 *	1	Visible cloud and surface features
2	3.9	4	Infrared low-level cloud/fog, fire detection
3	6.5	4	Infrared mid-level water vapor
4	10.7	4	Infrared surface/cloud-top temperature
6	13.3	4 - 8 **	Infrared CO_2; cloud heights
* The band 1 central wavelength is 0.65 μm for GOES-12, 0.63 μm for GOES-13, and 0.62 μm for GOES-14 and 15.			
** The band 6 resolution is 8 km for GOES-12 and 13, and 4 km for GOES-14 and 15.			

GOES Variable Sounder

Satellites GOES-8 through GOES-15 (12 is operational at ~60°W; 13 is operational at ~75°W; 14 is in standby at ~89.5°W; 15 is operational at ~135°W)

Orbit Type Geostationary

Instrument Sounder

Resolution 10 km

Band	Central Wavelength (μm)	Absorbing Gases	Wave Number (cm-1)	Sounding Level (mb)	Primary Uses
1	14.7	CO_2	680	90	Stratosphere temperature
2	14.4	CO_2	694	300	Sounding; tropopause temperature
3	14.1	CO_2	709	550	Sounding; upper-level temperature
4	13.6	CO_2; H_2O	735	850	Sounding; mid-level temperature
5	13.4	CO_2; H_2O	746	900	Sounding; low-level temperature
6	12.7	H_2O	787	Surface	Sounding; total precipitable water
7	12.0	H_2O	833	Surface	Surface temperature and moisture
8	11.0	H_2O	909	Surface	Surface temperature
9	9.7	O_3	1031	25	Total ozone
10	7.4	H_2O	1351	725	Low-level moisture
11	7.0	H_2O	1428	550	Sounding; mid-level moisture
12	6.5	H_2O	1538	350	Sounding; upper-level moisture
13	4.6	N_2O	2173	1000	Sounding; low-level temperature
14	4.5	N_2O	2222	950	Sounding; mid-level temperature
15	4.4	CO_2; N_2O	2272	650	Sounding; upper-level temperature
16	4.1	CO_2	2439	Surface	Sounding; boundary layer
17	4.0	Window	2500	Surface	Temperature
18	3.7	Window	2702	Surface	Temperature
19	0.70	Window	14286	Surface	Clouds
20					Star scan

GOES AAA

Satellites GOES-6 and GOES-7 (no longer in use)

Orbit Type Geostationary

Instrument VAS

Resolution 1 km VIS, 4/8 km IR

Band	Central Wavelength (μm)	Major Absorbing Gases	Sounding Level (mb)	Primary Uses
1	14.7 (IR)	CO_2	70	Temperature sounding
2	14.5 (IR)	CO_2	125	Temperature sounding
3	14.3 (IR)	CO_2	200	Temperature sounding
4	14.0 (IR)	CO_2	500	Temperature sounding, cloud heights
5	13.3 (IR)	CO_2	920	Temperature sounding, cloud heights
6	4.5 (IR)	CO_2, N_2O	850	Temperature sounding
7	12.7 (IR)	H_2O	1000	Water vapor detection
8	11.2 (IR)	Window	Surface	Nighttime cloud detection, surface temperature
9	7.3 (IR)	H_2O	600	Water vapor detection
10	6.7 (IR)	H_2O	400	Water vapor detection
11	4.4 (IR)	CO_2, N_2O	300	Temperature sounding
12	3.9 (IR)	Window	Surface	Surface skin temperature, cloud detection
13*	-	-	-	-
NA	0.65 (VIS)	Window	Surface	Daytime cloud detection
* Band 13 in a McIDAS Area is the same as band 8 but at lower resolution.				

Meteosat

Satellites Meteosat-1 through Meteosat-7 (7 is operational at ~57°E)

Orbit Type Geostationary

Instrument METEOSAT Radiometer

Resolution 2.5 km VIS, 5 km IR

Band	Central Wavelength (μm)	Primary Uses
1	0.75	Visible cloud and surface features
8	11.5	Infrared surface/cloud-top temperature
10	6.9	Infrared mid-level water vapor

Meteosat Second Generation

Satellites Meteosat-8 and Meteosat-9 (8 is in backup mode; 9 is operational at ~0°E)

Orbit Type Geostationary

Instrument SEVIRI

Resolution 1 km Broadband Visible, 3 km all others

Band	Central Wavelength (μm)	Primary Uses
1	0.6	Visible cloud and surface features
2	0.8	Visible aerosols over water, vegetation
3	1.6	Near-Infrared Surface, cloud phase
4	3.9	Infrared low-level cloud/fog, fire detection
5	6.2	Infrared upper-level water vapor
6	7.3	Infrared mid-level water vapor
7	8.7	Infrared total water, cloud phase, dust
8	9.7	Ozone
9	10.8	Infrared surface/cloud-top temperature
10	12.0	Infrared surface/cloud temperature, low-level water vapor
11	13.4	Infrared CO_2; cloud heights
12	0.7	Broadband visible (0.4 - 1.1 μm)

GMS

Satellites GMS-4 with sensor source numbers 12 and 13 (no longer in use)

Orbit Type Geostationary

Instrument VISSR

Resolution 1.25 km VIS, 5.0 km IR

Band	Central Wavelength (μm)	Primary Uses
Raw Ingest:		
1	11.5	Infrared surface/cloud-top temperature
4	Documentation	Not displayable; ingest documentation
14	0.63	Visible cloud and surface features
Converted:		
1	0.63	Visible cloud and surface features
8	11.5	Infrared surface/cloud-top temperature

Satellites GMS-4 through GMS-5 with sensor source numbers 82 through 86 (no longer in use)

Orbit Type Geostationary

Instrument VISSR

Resolution 1.25 km VIS, 5.0 km IR

Band	Central Wavelength (μm)	Primary Uses
1	0.63	Visible cloud and surface features
2	11.0	Infrared surface/cloud-top temperature
3	12.0	Infrared surface/cloud temperature, low-level water vapor
4	6.8	Infrared mid-level water vapor
5	Documentation	Not displayable; ingest documentation

MTSAT

Satellites MTSAT-1R and MTSAT-2 (1R is backup at ~140°E; 2 is operational at ~145°E)

Orbit Type Geostationary

Data Type HRIT (current) or HiRID (terminated March 2008)

Instrument Imager

Resolution HRIT: 1.0 km VIS, 4.0 km IR;
HiRID: 1.25 km VIS, 5.0 km IR

Band	Central Wavelength (μm)	Primary Uses
1	0.73	Visible cloud and surface features
2	10.8	Infrared surface/cloud-top temperature
3	12.0	Infrared surface/cloud temperature, low-level water vapor
4	6.75	Infrared mid-level water vapor
5	3.75	Infrared low-level cloud/fog, fire detection

FY-2

Satellites FY-2A and FY-2B (2A and 2B are non-operational)

Orbit Type Geostationary

Instrument VISSR

Resolution 1.25 km VIS, 5.0 km IR

Band	Central Wavelength (μm)	Primary Uses
1	0.8	Visible cloud and surface features
2	11.5	Infrared surface/cloud-top temperature
3	6.8	Infrared mid-level water vapor

Satellites FY-2C through FY-2E (2C is non-operational; 2D is operational at ~86°E; 2E is operational at ~105°E)

Orbit Type Geostationary

Instrument S-VISSR

Resolution 1.25 km VIS, 5.0 km IR

Band	Central Wavelength (μm)	Primary Uses
1	0.73	Visible cloud and surface features
2	10.8	Infrared surface/cloud-top temperature
3	12.0	Infrared surface/cloud temperature, low-level water vapor
4	6.8	Infrared mid-level water vapor
5	3.8	Infrared low-level cloud/fog, fire detection

Kalpana

Satellites Kalpana-1 (positioned at ~74°E)

Orbit Type Geostationary

Instrument VHRR

Resolution 2.0 km VIS, 8.0 km IR

Band	Central Wavelength (μm)	Primary Uses
1	0.65	Visible
2	11.5	Infrared
3	6.4	Water vapor

COMS

Satellites COMS-1 (positioned at ~128°E)

Orbit Type Geostationary

Instrument Meteorological Imager (MI)

Resolution 1.0 km VIS, 4.0 km IR

Band	Central Wavelength (μm)	Primary Uses
1	0.675	Visible cloud and surface features
2	3.75	Infrared low cloud/fog, fire detection
3	6.75	Infrared mid-level water vapor
4	10.8	Infrared surface/cloud-top temperature
5	12.0	Infrared surface/cloud-top temperature, low-level water vapor

POES

Satellites NOAA-6 through NOAA-19 (see http://www.oso.noaa.gov/poesstatus/ for current status)

Orbit Type Polar

Data Types HRPT, GAC and LAC

Instruments AVHRR (NOAA-6 through 14) and AVHRR/3 (NOAA-15 through 19)

Resolution 1 km (VIS and IR)

Band	Central Wavelength (μm)	Primary Uses
1	0.63	Visible cloud and surface features
2	0.86	Visible aerosols over water, vegetation
3*	3.74	Infrared low-level cloud/fog, fire detection
4	10.8	Infrared surface/cloud-top temperature
5	12.0	Infrared surface/cloud temperature, low-level water vapor; band 5 on NOAA-10 is not available
6*	1.61	Near-infrared surface, cloud phase (NOAA-15 and later)
* For NOAA-15 and later, channels 3A and 3B are labeled in McIDAS as bands 6 and 3, respectively.		

Satellites NOAA-6 through NOAA-19 (see http://www.oso.noaa.gov/poesstatus/ for current status)

Orbit Type Polar

Data Type TIP

Instruments HIRS (NOAA-6 through 14), HIRS/3 (NOAA-15 through 17) and HIRS/4 (NOAA-18 and later)

Resolution 20 km

Band	Central Wavelength (μm)	Major Absorbing Gases	Sounding Level (mb)	Primary Uses
1	15.0 (IR)	CO_2	30	Temperature sounding
2	14.7 (IR)	CO_2	60	Temperature sounding
3	14.5 (IR)	CO_2	100	Temperature sounding
4	14.2 (IR)	CO_2	400	Temperature sounding
5	14.0 (IR)	CO_2	600	Temperature sounding
6	13.6 (IR)	CO_2, H_2O	800	Temperature sounding
7	13.4 (IR)	CO_2, H_2O	900	Temperature sounding
8	11.1 (IR)	Window	Surface	Surface temperature
9	9.71 (IR)	O_3	25	Ozone measurement
10	8.2 (IR)	H_2O	900	Water vapor detection (NOAA-6 through 10 and 12)
10	12.5 (IR)	H_2O	Surface	Water vapor detection (NOAA-11, and 14 and later)
11	7.33 (IR)	H_2O	700	Water vapor detection

12	6.8 (IR)	H_2O	500	Water vapor detection (NOAA-6 through 14)
12	6.52 (IR)	H_2O	500	Water vapor detection (NOAA-15 and later)
13	4.57 (IR)	N_2O	1000	Temperature sounding
14	4.52 (IR)	N_2O	950	Temperature sounding
15	4.47 (IR)	CO_2, N_2O	700	Temperature sounding
16	4.45 (IR)	CO_2, N_2O	400	Temperature sounding
17	4.13 (IR)	CO_2	Surface	Temperature sounding
18	4.00 (IR)	Window	Surface	Surface temperature
19	3.76 (IR)	Window	Surface	Surface temperature
20	0.69 (VIS)	Window	Cloud	Daytime cloud detection

Satellites NOAA-6 through NOAA-14 (see http://www.oso.noaa.gov/poesstatus/ for current status)

Orbit Type Polar

Data Type TIP

Instrument MSU

Resolution 150 km

Band	Frequency (GHz)	Major Absorbing Gases	Primary Uses
1	50.3	O_2	Temperature sounding
2	53.8	O_2	Temperature sounding
3	55.0	O_2	Temperature sounding
4	58.0	O_2	Temperature sounding

Satellites NOAA-15 through NOAA-19 (see http://www.oso.noaa.gov/poesstatus/ for current status)

Orbit Type Polar

Data Type TIP

Instrument AMSU-A

Resolution 50 km

Band	Frequency (GHz)	Primary Uses
1	23.8	Surface temperature
2	31.4	Surface temperature
3	50.3	Surface temperature
4	52.8	Troposphere temperature
5	53.6	Troposphere temperature
6	54.4	Troposphere temperature

7	54.9	Troposphere temperature
8	55.5	Tropopause temperature
9	57.3	Tropopause temperature
10	57.3	Temperature sounding: 2-band combo 57.3 +/- .2
11	57.3	Temperature sounding: 4 bands (57.3 +/- .3) +/- .048
12	57.3	Temperature sounding: 4 bands (57.3 +/- .3) +/- .022
13	57.3	Temperature sounding: 4 bands (57.3 +/- .3) +/- .010
14	57.3	Temperature sounding: 4 bands (57.3 +/- .3) +/- .004
15	89.0	Surface characteristics

Satellites NOAA-15 through NOAA-19 (see http://www.oso.noaa.gov/poesstatus/ for current status)

Orbit Type Polar

Data Type TIP

Instrument AMSU-B

Resolution 16 km

Band	Frequency (GHz)	Primary Uses
16	89.0	Temperature sounding: 2 band combo 89.0 +/- .9
17	150.0	Temperature sounding: 2 band combo 150.0 +/- .9
18	183.3	Temperature sounding: 2 band combo 183.3 +/- 1.0
19	183.3	Temperature sounding: 2 band combo 183.3 +/- 3.0
20	183.3	Temperature sounding: 2 band combo 183.3 +/- 7.0

Metop

Satellites Metop-A and Metop-B (see http://www.eumetsat.int/ for current status)

Orbit Type Polar

Instrument AVHRR/3

Resolution 1 km (VIS and IR)

Band	Central Wavelength (μm)	Primary Uses
1	0.63	Visible cloud and surface features
2	0.86	Visible aerosols over water, vegetation
3	3.74	Infrared low-level cloud/fog, fire detection
4	10.8	Infrared surface/cloud-top temperature
5	12.0	Infrared surface/cloud temperature, low-level water vapor
6	1.61	Near-infrared surface, cloud phase

Satellites Metop-A and Metop-B (see http://www.eumetsat.int/ for current status)

Orbit Type Polar

Instrument HIRS/4

Resolution 20 km

Band	Central Wavelength (μm)	Major Absorbing Gases	Sounding Level (mb)	Primary Uses
1	15.0 (IR)	CO_2	30	Temperature sounding
2	14.7 (IR)	CO_2	60	Temperature sounding
3	14.5 (IR)	CO_2	100	Temperature sounding
4	14.2 (IR)	CO_2	400	Temperature sounding
5	14.0 (IR)	CO_2	600	Temperature sounding
6	13.6 (IR)	CO_2, H_2O	800	Temperature sounding
7	13.4 (IR)	CO_2, H_2O	900	Temperature sounding
8	11.1 (IR)	Window	Surface	Surface temperature
9	9.71 (IR)	O_3	25	Ozone measurement
10	12.5 (IR)	H_2O	Surface	Water vapor detection
11	7.33 (IR)	H_2O	700	Water vapor detection
12	6.52 (IR)	H_2O	500	Water vapor detection
13	4.57 (IR)	N_2O	1000	Temperature sounding
14	4.52 (IR)	N_2O	950	Temperature sounding
15	4.47 (IR)	CO_2, N_2O	700	Temperature sounding

16	4.45 (IR)	CO_2, N_2O	400	Temperature sounding
17	4.13 (IR)	CO_2	Surface	Temperature sounding
18	4.00 (IR)	Window	Surface	Surface temperature
19	3.76 (IR)	Window	Surface	Surface temperature
20	0.69 (VIS)	Window	Cloud	Daytime cloud detection

Satellites Metop-A and Metop-B (see http://www.eumetsat.int/ for current status)

Orbit Type Polar

Instrument AMSU

Resolution 50 km

Band	Frequency (GHz)	Primary Uses
1	23.8	Surface temperature
2	31.4	Surface temperature
3	50.3	Surface temperature
4	52.8	Troposphere temperature
5	53.6	Troposphere temperature
6	54.4	Troposphere temperature
7	54.9	Troposphere temperature
8	55.5	Tropopause temperature
9	57.3	Tropopause temperature
10	57.3	Temperature sounding: 2-band combo 57.3 +/- .2
11	57.3	Temperature sounding: 4 bands (57.3 +/- .3) +/- .048
12	57.3	Temperature sounding: 4 bands (57.3 +/- .3) +/- .022
13	57.3	Temperature sounding: 4 bands (57.3 +/- .3) +/- .010
14	57.3	Temperature sounding: 4 bands (57.3 +/- .3) +/- .004
15	89.0	Surface characteristics

Satellites Metop-A and Metop-B (see http://www.eumetsat.int/ for current status)

Orbit Type Polar

Instrument MHS

Resolution 16 km

Band	Frequency (GHz)	Primary Uses
1	89.0	Temperature sounding: 2 band combo 89.0 +/- .9
2	157.0	Temperature sounding: 2 band combo 157.0 +/- .9
3	183.3	Temperature sounding: 2 band combo 183.3 +/- 1.0

4	183.3	Temperature sounding: 2 band combo 183.3 +/- 3.0
5	190.3	Temperature sounding: 2 band combo 190.3 +/- 7.0

EOS

Satellites Terra and Aqua

Orbit Type Polar

Instrument MODIS

Resolution 0.25 km bands 1-2, 0.50 km bands 3-7, 1.0 km bands 8-38

Band	Central Wavelength (μm)	Primary Uses
1	0.6465	Land/Cloud Boundaries
2	0.8567	Land/Cloud Boundaries
3	0.4656	Land/Cloud Properties
4	0.5537	Land/Cloud Properties
5	1.2419	Land/Cloud Properties
6	1.6291	Land/Cloud Properties
7	2.1143	Land/Cloud Properties
8	0.4118	Ocean Color/Plankton/Biogeochemistry
9	0.4421	Ocean Color/Plankton/Biogeochemistry
10	0.4869	Ocean Color/Plankton/Biogeochemistry
11	0.5297	Ocean Color/Plankton/Biogeochemistry
12	0.5468	Ocean Color/Plankton/Biogeochemistry
13*	0.6656	Low polarization Ocean Color/Plankton/Biogeochemistry
14*	0.6767	Low polarization Ocean Color/Plankton/Biogeochemistry
15	0.7464	Ocean Color/Plankton/Biogeochemistry
16	0.8662	Ocean Color/Plankton/Biogeochemistry
17	0.9041	Atmospheric Water Vapor
18	0.9353	Atmospheric Water Vapor
19	0.9361	Atmospheric Water Vapor
20	3.7882	Surface/Cloud Temperature
21	3.9921	Surface/Cloud Temperature
22	3.9719	Surface/Cloud Temperature
23	4.0567	Surface/Cloud Temperature
24	4.4732	Atmospheric Temperature
25	4.5454	Atmospheric Temperature
26	1.3820	Cirrus Cloud Water Vapor
27	6.7654	Cirrus Cloud Water Vapor
28	7.3367	Cirrus Cloud Water Vapor
29	8.5288	Cirrus Cloud Water Vapor
30	9.7344	Ozone
31	11.0186	Surface/Cloud Temperature
32	12.0325	Surface/Cloud Temperature
33	13.3651	Cloud Top Altitude
34	13.6835	Cloud Top Altitude
35	13.9133	Cloud Top Altitude
36	14.1957	Cloud Top Altitude

37*	0.6656	High polarization Ocean Color/Plankton/Biogeochemistry
38*	0.6767	High polarization Ocean Color/Plankton/Biogeochemistry
* Channels 13B and 14B are labeled in McIDAS as bands 37 and 38, respectively		

Satellite Aqua

Orbit Type Polar

Instrument AIRS

Resolution 13.5 km

AIRS data contains 2378 bands (channels). They are not listed here because of space considerations. The website http://disc.gsfc.nasa.gov/AIRS/documentation.shtml and files **AIRSBAND.DOC** and **SATBAND** in the **~mcidas/data** directory contain complete listings of the bands.

FY-1

Satellites FY-1C and FY-1D

Orbit Type Polar

Data Type CHRPT

Instrument MVISR

Resolution 1.0 km (VIS and IR)

Band	Central Wavelength (μm)	Primary Uses
1	0.63	Visible cloud and surface features
2	0.865	Visible aerosols over water, vegetation
3	3.74	Heat source, nighttime cloud detection
4	10.8	Infrared surface/cloud-top temperature
5	12.0	Infrared surface/cloud temperature, low-level water vapor
6	1.61	Near-infrared surface, cloud phase
7	0.455	Ocean color
8	0.505	Ocean color
9	0.55	Ocean color
10	0.932	Water vapor

Terms

Term	Definition
absorbing gases	primary gases that absorb and emit radiation at the central wavelength
band	number given to the wavelength channel of the satellite data
B-sector	data covering Europe, northern Africa, and the eastern Atlantic
central wavelength	wavelength falling in the middle of the band over which the satellite collects data
primary uses	examples of how a band is used in physical applications
sounding level	average level where most of the radiant energy sensed by the satellite originates, assuming a cloud-free atmosphere
window	channel that is transparent to atmospheric radiation

Acronyms

Acronym	Definition
AIRS	Atmospheric Infrared Sounder
AMSU	Advanced Microwave Sounding Unit
AVHRR	Advanced Very High Resolution Radiometer
CHRPT	Color High Resolution Picture Transmission
EOS	Earth Observing System
FY	Fengyun (Chinese satellite series)
GAC	Global Area Coverage
GMS	Geostationary Meteorological Satellite
GOES	Geostationary Operational Environmental Satellite
HiRID	High Resolution Image Data
HIRS	High resolution Infrared Radiation Sounder
HRIT	High Rate Information Transmission
HRPT	High Resolution Picture Transmission
IR	InfraRed
LAC	Local Area Coverage
MetOp	Meteorological Operational (EUMETSAT Polar System satellite)
MHS	Microwave Humidity Sounder
MODIS	MODerate-resolution Imaging Spectroradiometer
MSG	Meteosat Second Generation
MSI	MultiSpectral Imagery
MSU	Microwave Sounding Unit
MTSAT	Multi-functional Transport SATellite
MVISR	Multichannel Visible and InfraRed Scan Radiometer
NIR	Near InfraRed
NOAA	National Oceanic and Atmospheric Administration
POES	Polar Operational Environmental Satellite
S-VISSR	Stretched Visible and Infrared Spin Scan Radiometer
SEVIRI	Spinning Enhanced Visible and InfraRed Imager
TIP	TIROS Information Processor
VAS	VISSR Atmospheric Sounder
VHRR	Very High Resolution Radiometer
VIS	VISible
VISSR	Visible and InfraRed Spin Scan Radiometer
WV	Water Vapor

Appendix E

Stability Parameters

This appendix defines some of the sounding, parcel, and stability parameters output by the HODO, UALIST, and UAPLOT commands. The *surface parcel* that appears in several descriptions uses the mean temperature and mixing ratio from the boundary layer defined by the DEPTH keyword in the commands. The *environment* is defined as the sounding's measured conditions (temperature, dew point, wind, etc.) at the corresponding level. Parcels that are lifted adiabatically are lifted dry adiabatically until saturation, and moist adiabatically thereafter.

The parameters are listed alphabetically below with the abbreviation used by the commands in parenthesis.

The stability parameters defined here are:

- Convective Available Potential Energy (CAPE)
- Convective Inhibition (CIN)
- Convective Temperature (CVT)
- Dew Point Temperature (TD)
- Equilibrium Level (EL)
- Equivalent Potential Temperature (ThE)
- Forecast Maximum Temperature (FMAX)
- Helicity (HELI)
- K-Index (KI)
- Level of Free Convection (LFC)
- Lifted Condensation Level (LCL)
- Lifted Index (LI)
- Mixing Ratio (MIX)
- Potential Temperature (Th)
- Precipitable Water (PW)
- Severe WEAther Threat Index (SwI)
- Showalter Index (ShI)
- Total Totals Index (TTI)

Convective Available Potential Energy (CAPE)

Convective Available Potential Energy (aka, Positive Area) is the amount of energy available to a parcel as it freely rises between the Level of Free Convection (LFC) and the Equilibrium Level (EL). The CAPE is non-zero only if an LFC exists. The default units for CAPE are J/kg.

Convective Inhibition (CIN)

Convective Inhibition (aka, Negative Area) is the amount of energy that must be supplied to a parcel for it to rise to the Level of Free Convection (LFC). Convective Inhibition is non-zero only if an LFC exists. The default units for CIN are J/kg.

Convective Temperature (CVT)

The Convective Temperature is the temperature to which a surface parcel must be heated to allow it to freely convect. The CVT is found by first computing a surface parcel's equivalent potential temperature (theta-E). This is followed by adiabatically lifting the parcel to the point where its theta-E equals the environment's theta-E. From this point (called the Convective Condensation Level), the parcel is brought down to the surface pressure dry adiabatically. The temperature at the surface pressure is the CVT. The default units for CVT are degrees Celsius.

Dew Point Temperature (TD)

The Dew Point Temperature is the temperature to which a parcel must be cooled (at constant pressure) for saturation to occur. The default units for TD are degrees Celsius.

Equilibrium Level (EL)

The Equilibrium Level (aka, Equilibrium Pressure) is the pressure at which a rising saturated parcel encounters negative buoyancy. Above that pressure, the parcel is cooler than its surroundings. The default units for EL are mb.

Equivalent Potential Temperature (ThE)

The Equivalent Potential Temperature is the Potential Temperature a parcel would have if all its moisture were condensed out, liberating the latent heat. The default units for ThE are degrees Kelvin.

Forecast Maximum Temperature (FMAX)

The Forecast Maximum Temperature is found by displacing a parcel at the top of the boundary layer to the surface pressure dry adiabatically. It's best used with morning soundings. The default units for FMAX are degrees Celsius.

Helicity (HELI)

Helicity is a measure of the storm-relative streamwise vorticity advection. It gives an indication of the low-level shear in the atmosphere. Large values of helicity are correlated with supercellular thunderstorm development. The default units for HELI are m**2/s**2.

K-Index (KI)

The K-Index is a measure of thunderstorm potential based on lapse rate and the vertical extent of the moisture content in the lower atmosphere.
KI= (850 mb temp - 500 mb temp) + 850 mb dew point - 700 mb dew point depression

Level of Free Convection (LFC)

The Level of Free Convection is the level at which the temperature of a surface parcel that's lifted adiabatically first becomes warmer than the temperature of the environment. The default units for LFC are mb.

Lifted Condensation Level (LCL)

The Lifted Condensation Level is the level at which a surface parcel lifted dry adiabatically becomes saturated. The default units for LCL are mb.

Lifted Index (LI)

The Lifted Index is a measure of thunderstorm potential. It is found by lifting a surface parcel adiabatically to 500 mb. The difference between the 500 mb temperature and the lifted parcel's temperature is the LI.

Mixing Ratio (MIX)

The Mixing Ratio is a measure of the amount of water vapor in the air. The default units for MIX are g/kg (grams of water vapor per kilogram of dry air).

Potential Temperature (Th)

The Potential Temperature is the temperature a parcel would have if moved dry adiabatically to 1000 mb. The default units for Th are degrees Kelvin.

Precipitable Water (PW)

Precipitable Water is the sum of average mixing ratios across pressure layers up to and including 500 mb, i.e., w(ave) * DP, where w(ave) represents the average mixing ratio across the pressure layer (DP). The data used is from the interpolated sounding seen with the following command: UALIST *station* OPT=ALL. The default units for PW are mm.

Severe WEAther Threat Index (SwI)

The Severe WEAther Threat Index (aka, SWEAT Index) is a measure of thunderstorm potential. It is a composite index that combines 850 mb moisture, wind speed and direction, the Total Totals Index, 500 mb speed and direction, and wind shear.
SwI = 12*(Td at 850) + 20*MAX((Total Totals Index - 49), 0) + 2*(850 mb wind speed in knots) + (500 mb wind speed in knots) + SHEAR, with the following conditions:

- if the 850 mb dewpoint is <0, it is set to zero
- the SHEAR term is zero if the 850 mb wind direction is < 130 or > 250 or if the 500 mb wind direction is < 210 or > 310 or if DIR500-DIR850 < 0 or if the windspeed at 500 mb or 850 mb is less than or equal to 15 knots; otherwise, the SHEAR term is 125 * (sine (direction at 500 mb - direction at 850 mb) + 0.2)

Showalter Index (ShI)

The Showalter Index is a measure of thunderstorm potential. It is equal to the difference between the sounding's 500 mb temperature and the temperature of the sounding's 850 mb parcel lifted adiabatically to 500 mb.

Total Totals Index (TTI)

The Total Totals Index a measure of thunderstorm potential. It is equal to the sum of the Vertical Totals (VT) and Cross Totals (CT).
TTI = Vertical Totals + Cross Totals = (850 mb temp + 850 mb dew point) - 2 (500 mb temp)

Appendix F

System Files

Listed below in alphabetical order are some of the McIDAS-X system files. Some are required to run McIDAS while most are used by McIDAS commands (e.g., **OUTLUSAM** is used by command MAP). You can manually change only those files below with the statement "can be modified with a text editor" in their function description.

The system files are located in the **~mcidas/data**, **$HOME/mcidas/data**, **$HOME/.mctmp** and other **MCPATH** and **REDIRECT** directories. File names ending with **.CORE** (e.g., **MAP.CORE** and **UACROSS.CORE**) are supplied with McIDAS and should not be modified. Site and user versions of these files (if they exist) end with **.SITE** and **.USER**, respectively.

File name	Function
ADDERROR.DOC	list of ADDE error messages and the routines that produce them
AIRSBAND.DOC	text file containing information about each of the 2378 bands in AIRS data
AIRSBANDINFO	binary file created from **AIRSBAND.DOC** that the AIRS server uses to map wavelength and wave number to band number
ALLOC.WWW	allocation of frame sizes
AREA*nnnn*	area files, where *nnnn* is the area number
BASICVV1	LW file used by the RUN command to store variables
***.bdf**	bitmap distribution files that define fonts you can use
BREF.ET	image enhancement for NEXRAD Base Reflectivity products
bufrVersion*.txt	text files used by the -X BUFR server and -XCD BUFR filer to label BUFR files in a consistent manner using category/subcategory
CNTYOUTL.DAT	county outline basemap for the WWDISP command
CONTEXT.DAT	configuration settings for the McIDAS-X GUI
CONTOUR.DEF	default contour intervals for the GRDDISP and PTCON commands; can be modified with a text editor
CORE.SAT	satellite names and information used by the ADDE navigation commands
COUNTRY.TXT	list of countries and their two-letter codes used by the CCODE command
COUNTRY.TXT.OLD	list of countries and their two-letter codes used in McIDAS-X versions before 2002; this file is for reference purposes only so users who access MD files created with a version earlier than 2002 can find out which country codes are used in those files
CREF.ET	image enhancement for NEXRAD Composite Reflectivity products
DCGFSMOS	text file used by the SCHE command to register the GFS MOS MD file schema
DCGRET	LW file used by the SCHE command to register the GRET MD file schema
DCIHGT	LW file used by the SCHE command to register the IHGT MD file schema
DCIRAB	LW file used by the SCHE command to register the IRAB and IRSG MD file schemas
DCISEN	LW file used by the SCHE command to register the ISEN MD file schema
DCISFC	LW file used by the SCHE command to register the ISFC MD file schema
DCISHP	LW file used by the SCHE command to register the ISHP MD file schema
DCNAMMOS	text file used by the SCHE command to register the NAM MOS MD file schema
DCNGMMOS	text file used by the SCHE command to register the NGM MOS MD file schema
DCPIRP	LW file used by the SCHE command to register the PIRP MD file schema
DCRAOB	LW file used by the SCHE command to register the RAOB and RSIG MD file schemas
DCRAWI	LW file used by the SCHE command to register the RAWI MD file schema
DCSVCA	LW file used by the SCHE command to register the SVCA MD file schema
DCSYN	LW file used by the SCHE command to register the SYN MD file schema
DCUPPR	text file used by the SCHE command to register the UPPR MD file schema
DEFAULT.GRX	default graphics palette for the GU command
DNKEYS	list of MD file keys, field widths and units used by the system commands (ASCII format); LW file **FLDWDTH** is generated from this list

***.ET**	image enhancement files created with the EU command
***.FFD**	font types used by the FONT command and global keyword
FLDWDTH	LW file list of MD file keys and field widths for the MDL command
FRAMED	frame panel information
FRAMENH.001	frame enhancement
Frame*n.p*	frame directory where *n* is the frame number and *p* is the panel number; examples are 1.1, 3.2, 2.4, 10.2
gbtbpds001.*	text files used to translate GRIB file information to McIDAS grid header data; file **gbtbpds001.0v1** contains originating center information; files **gbtbpds001.2v*** contain parameter names and units; file **gbtbpds001.4v1** contains time definitions; files **gbtbpds001.a*** contain model information; files **gbtbpds001.b*** contain map/domain information
GEO.CORE	parameter and keyword settings for the GEO command
GEODATA.CORE	satellite dataset, channel, resolution and coverage settings for the GEO command
***.GIF**	GIF-format files created by the FRMSAVE command
GMSCAL	GMS calibration file
GR*aannnn*	grid files where *aa* is either part of the grid file number or the letters ID; *nnnn* is the grid file number
GRAPHIC.GRX	default graphics palette for the GU command
***.GRX**	graphics palette files created by the GU command
GRAPHICS.KEY	shared memory Block Key for graphics
grib2*.txt	text files used by the -X GRIB server and the -XCD GRIB filer to translate GRIB2 file information to McIDAS grid header data; file **grib2NCEPModels.txt** contains model information; file **grib2NavLookup.txt** contains navigation information; file **grib2Parameters.txt** contains parameter information
HIRSCRPF	HIRS calibration reference parameters for computing HIRS brightness temperatures
HIRSTAUL	HIRS transmittance coefficients
IDSVCA	surface station identifiers (binary format)
IMAGE.ET	default image enhancement for the EU command
IMGDISP.CORE	predefined products for the IMGDISP command
LEO.CORE	parameter and keyword settings for the LEO command
LEODATA.CORE	navigation dataset, satellite dataset, channel and resolution settings for the LEO command
license.txt	McIDAS software acquisition and distribution policies
LREF.ET	image enhancement for NEXRAD Layer Reflectivity products
LWPATH.NAM	current file redirection listing
MAP.CORE	coverages and projections of predefined maps used by MAP, WWDISP and all other commands that allow specification of predefined maps
***.MCB**	McBASI program text files; can be modified with a text editor
MCCMDHIST.TXT	list of commands entered through the command line or run through the GUI; each new McIDAS session is indicated with the line *Starting a new McIDAS session on <date and time>*; the number of commands stored in the file is equal to the value set in the **-ih** flag in the **.mcidasrc** file; **MCCMDHIST.TXT** is accessed through the Command History GUI under the main GUI window's *List* menu or through the SEE command or a text editor for cutting and pasting
MCRGB.TXT	list of predefined colors and RGB values for the EU and GU commands (ASCII format); can be modified with a text editor
MCTABLE.TXT	ADDE client routing table; created and modified by the DATALOC and AKA commands
MD*aannnn*	MD files where *aa* is either part of the MD file number or the letters XX; *nnnn* is the MD file number
***.MNU**	F Key Menu text files; can be modified with a text editor
MOD06.cfg	sample configuration file used to map McIDAS parameter names to the actual parameter names in a MODIS MOD06 HDF point file
MOD07.cfg	sample configuration file used to map McIDAS parameter names to the actual parameter names in a MODIS MOD07 HDF point file
MODIS.CORE	MODIS Level 2 product server (MODX) display variables
*monyy***INQ.TXT**	list of an upgrade's changed modules, grouped by SSEC inquiry number; the date of the upgrade is indicated by *mon* (first three letters of the month, e.g., NOV for November) and *yy* (last two digits of the year, e.g., 01 for 2001)
*monyy***OPN.TXT**	list of open (unresolved) SSEC inquiries at the time of the upgrade; the date of the upgrade is indicated by *mon* (first three letters of the month, e.g., NOV for November) and *yy* (last two digits of the year, e.g., 01 for 2001)
*monyy***PRG.TXT**	list of an upgrade's changed modules, in alphabetical order; the date of the upgrade is indicated by *mon* (first three letters of the month, e.g., NOV for November) and *yy* (last two digits of the year, e.g., 01 for 2001)

MSUSCRPF	MSU calibration reference parameters
***.NAM**	file redirection listings for the REDIRECT command
netcdfbuoy.nc	sample netCDF point file containing buoy data; **netcdfpoint.cfg** is a sample configuration file for this type of data
netcdfgrid.cfg	sample configuration file used to map McIDAS parameter names to the actual parameter names in a netCDF grid file
netcdfpoint.cfg	sample configuration file used to map McIDAS parameter names to the actual parameter names in a netCDF point file
OUTLANT	Antarctica basemap (high resolution)
OUTLARST	Argentinian states basemap (high resolution)
OUTLAUSS	Australia states basemap (high resolution)
OUTLAUST	Australian region basemap (high resolution)
OUTLAUSW	Australia's Lake Eyre basemap (high resolution)
OUTLEURO	European region basemap (high resolution)
OUTLHPOL	world political boundaries basemap (medium resolution)
OUTLHRES	world coastal boundaries basemap (high resolution)
OUTLSUPU	USA basemap with states and major waterways (very high resolution)
OUTLSUPW	world coastal boundaries basemap (medium resolution)
OUTLUSAL	North American basemap for text (nongraphical) maps (very low resolution)
OUTLUSAM	North American basemap with states and provinces (high resolution)
OUTLWRLD	world basemap for text (nongraphical) maps (very low resolution)
***.PIX**	virtual frame files saved with the SVF command
***.PRD**	predefined text products for WXTLIST command
PRE1.ET	image enhancement for NEXRAD 1-Hour Precipitation products
PRE3.ET	image enhancement for NEXRAD 3-Hour Precipitation products
PRET.ET	image enhancement for NEXRAD Storm-Total Precipitation products
RADAR.ET	radar enhancement for the EU command
RAOBCON.CORE	default parameter and keyword settings for the RAOBCON command
RAOBPLOT.CORE	default parameter and keyword settings for the RAOBPLOT command
RESOLV.SRV	ADDE dataset names on the local server; created and modified by the DSSERVE command
RVEL.ET	image enhancement for NEXRAD Radial Velocity products
SATANNOT	satellite annotation line information
SATBAND	image band information used by the IMGLIST command
SAVESTR	string table save file
SCHEMA	blueprints of all registered schemas
SERVERTABLE.TDAT	list of datasets available to the McIDAS-X GUI; the list is automatically generated by the GUI by running DSINFO commands for each group in the client routing table, **MCTABLE.TXT**
SFCCON.CORE	default parameter and keyword settings for the SFCCON command
SFCPLOT.CORE	default parameter and keyword settings for the SFCPLOT command
SKEDFILE	system scheduler commands file
***.ST**	stretch tables created with the SU command
STNDB.CORE	station database
STRTABLE	active string table files
SYSKEY.DOC	system key table; ASCII representation for the **SYSKEY.TAB** file; can be modified with a text editor
SYSKEY.TAB	system key table derived from **SYSKEY.DOC** (binary format)
TERMCHAR.001	McIDAS display characteristics file
TOPOHRES	topography file used by the ADT command
TOPOLRES	topography file used by the ADT command
TOPS.ET	image enhancement for NEXRAD Echo Tops products
towns.nd1	geographic data used by the TOWNS and FIND commands
towns.nd2	geographic data used by the TOWNS and FIND commands
towns.txt	geographic data used by the TOWNS and FIND commands
UACROSS.CORE	default units and keyword values for the UACROSS command

UASAVE.*fff*	sounding edit files created with UAPLOT command, where *fff* is the frame number that the diagram was displayed on
UC.DOC	list of User Common words and their functions
UNIDATA.MNU	sample F Key Menu template file (ASCII format)
USCOUNTY.MAP	National Weather Service county basemap used by the MAP and WWDISP commands
USSTATE.MAP	National Weather Service state basemap used by the MAP and WWDISP commands
USZONE.MAP	National Weather Service zone basemap used by the MAP and WWDISP commands
VASTBLS	calibration file used to display 2-byte mode AA VAS data transmitted by GOES satellites from 22 September 1980 to 18 November 1986
VERSION.TXT	ASCII text for the current version of McIDAS-X
VIL.ET	image enhancement for NEXRAD Vertically-Integrated Liquid products
VIRT*nnnn*	virtual graphics files created with the VIRT global keyword, where *nnnn* is the virtual graphic number
WXUSER.PRD	user's predefined text products used by the WXTLIST command; can be modified with a text editor; there may also be a **WXSITE.PRD** file created by the site administrator
XRD.TXT	list of commands available in McIDAS-XRD optional sidecar package; the version number of the package, e.g., McIDAS-XRD 2009.1, is specified near the beginning of the file

Appendix G

MCPATH

MCPATH is a Unix environment variable that defines directories for McIDAS-X commands to search when they look for data and help files. Commands requiring these files look in each of the directories listed in **MCPATH** to find them.

MCPATH has a default value that automatically locates the data files that required linking or redirection in previous versions of McIDAS-X. If users elect to set their own **MCPATH** value, they only need to know the directories where the files reside, not the exact file names.

This section consists of the following discussions:

- MCPATH Format
- Setting MCPATH
- Writing Data
- REDIRECT Precedence
- MCPATH Requirements
- Troubleshooting

MCPATH Format

The **MCPATH** variable specifies, in order of precedence, the directories McIDAS-X searches when looking for data and help files. A colon (:) separates each directory.

For example, if **MCPATH** is set to:

```
/experiment:/home/fred/mcidas/data:~mcidas/data:~mcidas/help
```

and McIDAS-X wants to read from a data file, it first looks for the file in **/experiment**. If the file is not there, McIDAS-X checks for the file in **/home/fred/mcidas/data**, then **~mcidas/data**, and finally **~mcidas/help**. If the file is not located in any of the directories, the system outputs a message saying the file was not found.

Setting MCPATH

You can set **MCPATH** in two ways:

- let the McIDAS-X start-up script set the value
- set **MCPATH**'s value before starting McIDAS-X

If you do not define **MCPATH**, the McIDAS-X start-up script sets **MCPATH** to the value below.

```
$HOME/mcidas/data:$HOME/mcidas/help:~mcidas/data:~mcidas/help
```

To change **MCPATH**'s value, exit McIDAS-X, set **MCPATH** to the new value, and restart McIDAS-X. Run a Unix command (substitute the equivalent if using C shell) similar to the one below to explicitly set **MCPATH**.

```
export MCPATH; MCPATH=$HOME/data:$HOME/mcidas/data:~mcidas/data:~mcidas/help
```

To set this value of **MCPATH** each time you log on to your account, add the command above to your **$HOME/.mcenv** file and add the following line to your **$HOME/.profile** (ksh) or **$HOME/.cshrc** (csh) file.

```
. $HOME/.mcenv
```

When you start McIDAS-X, the temporary directory created, for example, **$HOME/.mctmp/356052**, is appended to the **MCPATH** value, whether the McIDAS-X start-up script sets **MCPATH** or you set its value.

Writing Data

If you run a McIDAS-X command that creates a new file, the file is placed in the first **MCPATH** directory where you have write permissions. If you run a command that writes to a file residing in one or more **MCPATH** directories, the command selects the file in the first **MCPATH** directory in which the file exists.

For example, if a file named **AREA0001** exists in the third and fourth **MCPATH** directories, McIDAS commands that write to the file (IMGCOPY, for example) will do so in the third **MCPATH** directory. If you do not have write permissions for that file, you may get unexpected results; for example, the command stops due to write failure.

REDIRECT Precedence

Files that are redirected with the McIDAS-X REDIRECT command, have precedence over files defined by **MCPATH**. This lets you explicitly state the location of a particular file. For example, if **MCPATH** is set to:

```
/home/experiment:/home/fred/data:~mcidas/data:~mcidas/help
```

and each of these directories contains a file named **AREA0001**, McIDAS commands that use the file (IMGDISP, for example) will use the one in **/home/experiment** if the file is not redirected to another directory. However, if you have the redirection entry:

```
AREA00* /home/dave/aircraft/data
```

the McIDAS command uses the file **AREA0001** stored in **/home/dave/aircraft/data** instead of a **MCPATH** directory.

MCPATH Requirements

MCPATH must contain the following directories:

- **~mcidas/data**
- **~mcidas/help**
- a directory where you have write permissions, for example, **$HOME/mcidas/data**

The first two directories let you access the required data and help files that are installed with each version of McIDAS-X. The third requirement lets you write or modify files with McIDAS-X commands like IMGCOPY, EU, and FRMSAVE.

Troubleshooting

If you have problems running McIDAS-X and believe the problems may be related to **MCPATH**, complete the following steps:

1. Run the McIDAS-X command below to verify that your current **MCPATH** value meets the three requirements listed above.

   Type: **OS "echo $MCPATH**

You will not see **~mcidas/data** and **~mcidas/help** in the directory list because the Unix **echo** command expands them to their full directory names, for example, **/home/mcidas/data** and **/home/mcidas/help**. If your current **MCPATH** value does not meet the three requirements listed above, exit McIDAS-X, reset **MCPATH** to a valid value, and restart McIDAS-X.

2. Run the following command from a Unix window to see if the files **SKEDFILE**, **ALLOC.WWW**, **ALLOC.FLG**, **FRNVDIR.001** and **FRNVEXP.001** reside in the **~mcidas/data** directory. Enter the command as one line.

   Type: **ls ~mcidas/data/SKEDFILE ~mcidas/data/ALLOC.* ~mcidas/data/*.001**

These files exist in the **~mcidas/data** directory if user **mcidas** started a McIDAS-X session sometime in the past. Since you can no longer run McIDAS-X as user **mcidas**, these files should not be present in **~mcidas/data**. If they are present, ask the system administrator or person who installs McIDAS-X to delete them, since they may cause problems with other users' McIDAS-X sessions.

Appendix H

Running Commands Outside a McIDAS-X Session

You can run McIDAS-X commands without starting a McIDAS-X session by entering commands at the Unix prompt or using the **mcenv** process. When you run commands without starting McIDAS-X, a McIDAS environment is created, like the one created for a McIDAS session. This appendix describes the following.

- Creating a McIDAS-X Environment
- Using Unix Conventions
- Entering Commands at the Unix Prompt
- Using the mcenv Process
- Running Scripts and Commands Using mcenv
- Starting Shell Scripts That Run mcenv
- Running mcenv From cron
- Passing Variables into a mcenv Script
- Using String Tables

Creating a McIDAS-X Environment

To run McIDAS-X commands outside of a McIDAS-X session, a McIDAS-X environment must be created. A McIDAS-X environment is the information a McIDAS session uses to run commands. It includes User Common, the redirect table, **MCPATH** and the frame objects that McIDAS-X programs need to run commands. This information is stored in a shared memory segment and a temporary subdirectory of **$HOME/.mctmp**.

A McIDAS-X environment is created automatically when you start a McIDAS command with the **.k** extension or run the **mcenv** process from a Unix prompt. The duration of the McIDAS-X environment depends on how it was started, as listed in the table below.

Starting Procedures from the Unix Prompt	**Duration**
mcidascommand.**k**	duration of the command
mcenv with a Unix script	duration of the script or program
mcenv without a Unix script	until **mcenv** is cancelled with the Unix **exit** command

When you are running commands outside of a McIDAS-X session, it is important to consider the duration of the environment. You don't want the environment to close before your commands or scripts are finished.

Using Unix Conventions

When entering McIDAS commands at the Unix prompt or with the **mcenv** process, enter the command in lowercase, followed by **.k**. All parameters and keywords are entered in the same form and case as in the McIDAS Text and Command Window, as shown in the example below.

```
& map.k VH IMA=1 GRA=1
```

Some characters require a preceding escape character so that Unix interprets them literally. Use a backslash (\) to escape the character immediately following it. Use pairs of double (") and single quotes (') to escape groups of characters. Double quotes escape all characters except other double quotes and dollar signs ($). Single quotes escape all characters except other single quotes.

The characters below have special meaning in Unix and require an escape character when entered as part of a McIDAS-X command.

- ampersand (&)
- semicolon (;)
- parentheses ( )
- pipe (|)
- less than and greater than symbols (< >)
- unpaired double quotation marks (")
- unpaired single quotation marks (')
- backslash (\)
- pound sign (#)
- asterisk (*)
- question mark (?)
- dollar sign ($)

In the example below, a backslash is used in the first line because the unpaired double quote must be escaped. In the second line, the paired single quotes do NOT need to be escaped; paired single quotes are also capable of escaping other characters, so the semicolon does not need to be escaped.

```
& skl.k \"GOES-9 & ptlist.k SURFACE/HOURLY SELECT='TIME 12 ; ID KBWI'
```

When using string names, the command must be preceded by the command **mceval.k** in addition to a pound sign (#). See *Using String Tables* for more information.

Entering Commands at the Unix Prompt

When you enter McIDAS-X commands from the Unix prompt, a McIDAS environment is started and lasts only the duration of the command. For example, a separate McIDAS environment is started for each command below.

```
$ imgdisp.k GOES/HEMI.1000 1 $ frmsave.k 1 NHEM
```

The command **imgdisp.k** loads the image in position 1000 of dataset GOES/HEMI into frame 1. When **imgdisp.k** is finished, the frame is deleted and the McIDAS environment closes. The Unix shell reads the next command, **frmsave.k**, and creates a new McIDAS environment and a new frame 1. The **frmsave.k** command saves the contents of frame 1 into the file **NHEM.GIF**. However, frame 1 is empty so **NHEM.GIF** is black except for the annotation line at the bottom.

Using the mcenv Process

Use the **mcenv** process to run programs, scripts, or McIDAS-X commands. When you use **mcenv**, you ensure that the same McIDAS environment is used for the life of the script or until you exit the **mcenv** process. You can enter single McIDAS-X commands or create a script with multiple McIDAS-X commands to manipulate frames, grids, and images without starting a McIDAS-X session. You can start the **mcenv** process from the Unix prompt, shell scripts, and the Unix scheduler **cron**.

The following sections describe:

- running scripts and commands using **mcenv**
- starting shell scripts that run **mcenv**
- running **mcenv** from **cron**
- passing variables into a **mcenv** script
- using string tables

Running Scripts and Commands Using mcenv

Use the format below for starting a **mcenv** process.

mcenv **-f** *framespec* **-e** *bytes* **-g** *number* **-i** *numberprog*

The table below provides a description of the flags. See the section *Configuring McIDAS-X Sessions with .mcidasrc* for more information about the **-f** and **-e**, **-g** and **-i** flags. The **-g** and **-i** flags are analogous to the **-graphicsColors** and **-imageColors** flags in that section.

Flag	Description
-f *framespec*	specifies a set of frames to include in the McIDAS environment (default=one, 480x640 frame)
-e *bytes*	specifies a memory pool for the frames
-g *number*	specifies the number of graphics levels (default=8)
-i *number*	specifies the number of display levels (default=48)
prog	specifies a program, script or command to run; see *Using Unix Conventions* for the correct format to enter McIDAS commands

You can nest **mcenv** processes. Commands run under a subsequent **mcenv** do not affect the McIDAS environment created by the initial **mcenv**.

The examples below show commands and scripts run from **mcenv**. The dollar signs ($) indicate the Unix prompt and the ampersands (&) indicate the McIDAS environment.

In the example below, the user started a McIDAS environment, logged on to the workstation, and copied an image from one dataset to another, then closed the McIDAS environment.

```
$ mcenv & logon.k USER 1234 & imgcopy.k EAST/VIS.1 MYDATA/VIS & exit $
```

In the example below, the user started a McIDAS environment with a 400 line by 500 element frame and ran the **mkvisgif** script. The second box shows the contents of the **mkvisgif** script.

```
$ mcenv -f 400x500 mkvisgif
```

```
#!/bin/sh # mkvisgif: this script generates GIFs logon.k USER 1234 imgcopy.k
EAST/VIS.1 MYDATA/VIS imgdisp.k MYDATA/VIS 1 frmsave.k 1 MYPICT.GIF exit 0
```

The exit status from a McIDAS command run within a script can be checked by adding the command **ret=$?** (Bourne or Korn shell) or **set ret=$status** (C shell) on the line following the McIDAS command. A nonzero exit status usually indicates failure. When that occurs, you can take appropriate action, e.g., run a different McIDAS command or exit the script.

Note for users running under C shell

When you start **mcenv**, it adds a new **.mctmp** subdirectory to your **MCPATH**. If your **SHELL** environment variable is set to**/bin/csh** (C shell), **mcenv** then runs your **.cshrc** file, if it exists. If **MCPATH** is set in **.cshrc**, it overwrites the **MCPATH** that **mcenv** just set. Edit your **.cshrc** file so that it checks to see if your **MCPATH** is already set. An example of **.cshrc** file's original **MCPATH** setting is shown below.

```
setenv MCPATH $HOME/mcidas/data:$HOME/mcidas/help:~mcidas/data:~mcidas/help
```

Make the following changes to the **MCPATH** setting.

```
if ( ! ${?MCPATH} ) then setenv MCPATH
$HOME/mcidas/data:$HOME/mcidas/help:~mcidas/data:~mcidas/help endif
```

Starting Shell Scripts That Run mcenv

Other programs and scripts can run **mcenv**. The example below shows a shell script that runs a **mcenv** script. The **mcenv** script appears in bold.

```
#!/bin/sh # Sample script MCPATH=/home/user/mcidas/data:/home/mcidas/data export
MCPATH mcenv -f 520x690 << 'EOF' logon.k USER 1234 imgcopy.k EAST/VIS MYDATA/VIS
imgdisp.k MYDATA/VIS 1 frmsave.k 1 MYPICT.GIF exit EOF exit 0
```

The **mcenv** script runs everything from the **logon.k** command to the EOF statement within the McIDAS environment set with a 520x690 frame. Note that single quotes surround the first EOF in the first line of the **mcenv** script. Single quotes prevent the shell from substituting any variables within the **mcenv** script before they are set to the correct value.

Running mcenv From cron

You can run **mcenv** from **cron**, the Unix scheduler. However, you must set the shell environment for **mcenv**. It is not started automatically when **mcenv** is run in **cron**. You can specify the environment as part of the **mcenv** script, or set it up before starting the script.

Setting the environment before starting the script

To set the shell environment before starting the script, follow the steps below.

1. Create a file that specifies the environment in which your script will run. In the example below, the file containing the **PATH** and **MCPATH** environment variables is named **$HOME/.cronprofile**.

```
#This file is named .cronprofile PATH=$HOME/bin:
PATH=$PATH:/usr/sww/bin:/usr/local/bin:$HOME/mcidas/bin
PATH=$PATH:/home/mcidas/bin:/bin:/usr/bin:/etc:/usr/ucb
PATH=$PATH:/usr/bin/X11:/common/tool/bin:. export PATH MCPATH=$HOME/mcidas/data
MCPATH=$MCPATH:/home/mcidas/data export MCPATH
```

2. Add a **crontab** entry to **cron** similar to one of the two examples below. Add the top entry if the script contains **mcenv** and the second entry if the **mcenv** is external to the script. Enter the entries below as single lines.

```
10 * * * * ( . /home/user/.cronprofile && /home/user/mcidas/data/mcenvscript )
```

```
10 * * * * ( . /home/user/.cronprofile && /home/mcidas/bin/mcenv -f 300x600
/home/user/ mcidas/data/mcenvscript)
```

Setting the environment in the script

To set the environment in the script, follow the steps below.

1. Add lines to the script that specify the environment. In the sample script below, the lines that set the environment appear in bold.

```
#!/bin/sh # Sample script #set up environment PATH=$HOME/bin:
PATH=$PATH:/usr/sww/bin:/usr/local/bin:$HOME/mcidas/bin
PATH=$PATH:/home/mcidas/bin:/bin:/usr/bin:/etc:/usr/ucb
PATH=$PATH:/usr/bin/X11:/common/tool/bin:. export PATH MCPATH=$HOME/mcidas/data
MCPATH=$MCPATH:/home/mcidas/data export MCPATH mcenv -f 520x690 << 'EOF' logon.k
USER 1234 imgcopy.k EAST/VIS MYDATA/VIS imgdisp.k MYDATA/VIS 1 frmsave.k 1
MYPICT.GIF exit EOF exit 0
```

2. Add an entry to **crontab** similar to the example below.

```
10 * * * * /home/user/mcidas/data/mcenvscript
```

3. It may be useful to redirect output from a **cron** job to a log file. Add lines similar to the example below to **crontab** to prevent output from being mailed by **cron** to the owner of the **crontab**. The entry below directs the standard error and output to a file.

```
10 * * * * /home/user/mcidas/data/mcenvscript >>/home/user/logs/mclog 2>&1
```

Passing Variables into a mcenv Script

Shell variables are not passed to **mcenv** unless they are first exported. If the script contains the **mcenv** command (as opposed to being run as argument of the **mcenv** command), any variables set before calling **mcenv** must be exported if they are to be used within the **mcenv** environment. In the example below, the variable SAT is being passed.

```
#!/bin/sh # Sample script # get first argument to script SAT=$1 export SAT # Set
MCPATH MCPATH=/home/user/mcidas/data:/home/mcidas/data export MCPATH mcenv -f
520x690 << 'EOF' logon.k USER 1234 imgcopy.k ${SAT}/VIS MYDATA/VIS imgdisp.k
MYDATA/VIS 1 frmsave.k 1 MYPICT.GIF exit EOF exit 0
```

Using String Tables

Strings previously set in McIDAS can still be accessed from **mcenv** or at a Unix prompt. The command **mceval.k** must start any line containing a McIDAS string variable and a backslash (\) must precede pound signs (#). For example, to use the string D with its value IMGDISP EAST/FDV.0 from **mcenv** or a Unix prompt, add the command **mceval.k** and a pound sign (#) as shown below.

```
$ mceval.k \#D 2 MAG=-4 DAY=\#Y
```

The command **mceval.k** does not let you use a semicolon(;) as an argument. To run consecutive commands on the same line, **mceval.k** must be specified each time, as shown below.

```
$ mceval.k \#D 2 MAG=-4 DAY=\#Y ; mceval.k IMGLIST EAST/FDV.0 DAY=\#Y
```

It may be easier to use Unix variables to build McIDAS commands rather than McIDAS strings when writing Unix scripts. To use a Unix variable, set the Unix script variable to the command or value. Use quotes to enclose the command or value. The following two boxed examples use a script to list images. The first example uses Unix variables and the second uses strings. Both scripts produce the same output.

```
#!/bin/sh # List images using Unix variables mcenv << 'EOF' logon.k USER 1234 #
Make key in # Set date to current Julian day date=`date -u +%y%j` # Build keyword
day="DAY=$date" # Build commands listFDV="imglist.k EAST/FDV" listboth="imglist.k
EAST/FDV $day ;imglist.k EAST/FDIR $day " # Run keyin eval $listFDV $day # Run same
keyin 2 times using a different form for each eval $listFDV $day \; $listFDV
DAY=${date} # Run keyin with a Unix variable inside another Unix variable # and
also utilizes a ; eval $listboth exit EOF

exit 0
```

```
#!/bin/sh # List images using strings mcenv << 'EOF' logon.k USER 1234 # Make
string # commands te.k listFDV \"IMGLIST EAST/FDV te.k listnext \"IMGLIST EAST/FDIR
DAY=\#Y # Run keyin mceval.k \#listFDV DAY=\#Y # Run same keyin 2 times mceval.k
\#listFDV DAY=\#Y ; mceval.k \#listFDV DAY=\#Y # Run keyin combination mceval.k
\#listFDV DAY=\#Y ; mceval.k \#listnext exit EOF exit 0
```

Appendix I

McIDAS-X Site Administration and Management

This appendix is intended for site administrators and operations staff responsible for acquiring and making data available to the site's users. It contains the following information:

- Configuring an ADDE Remote Server account on a McIDAS-X workstation
- Acquiring and Naming Data on a Remote Server
- Setting the ADDE Server Timeout
- Setting the Tracking Mode for Area and MTSAT Servers
- Updating Client Routing Tables
- Updating Clients to Use Compressed Data Transfers
- Activating Transaction Logging on a Remote Server
- Restricting Access to Remote Server Datasets
- Configuring MODIS Servers to use MRTSwath for Bowtie Correction

Configuring an ADDE Remote Server

If you have data on a McIDAS-X workstation that you want to make available to all of your site's users, you can create a remote server account on the workstation. Datasets in the remote server account are available to all other McIDAS-X clients. Each McIDAS-X workstation can have only one remote server account.

Before you configure the remote server account on your workstation, you must select a name for the account. Consider the following guidelines.

- Do not use the **mcidas** account as the remote server account. You cannot run McIDAS-X while logged in as **mcidas**. This account should only be used to install McIDAS-X and other SSEC-supplied software packages that run under it.
- The McIDAS-XCD and SSEC Data Ingestor (SDI) packages provide conventional and satellite data that you may want to make available to your ADDE clients. The -XCD and SDI packages require an account, usually named **oper**, for administrative purposes. Do **not** use this account as the remote server account.
- Although you can run ADDE client commands from the remote server account, you should use other accounts for this purpose. Dedicate the remote server account to acquiring and naming data for ADDE clients.

SSEC recommends adding a new account named **mcadde** for the remote server account. The remote server account must be configured with its default **umask** set to 755, its shell set to **sh** or a variant like **ksh** or **bash** in the **/etc/passwd** file (the remote server account is the only one with this requirement), plus have the appropriate directories and **PATH** to run McIDAS-X. The **PATH** requirements are discussed in *Configuring a New User Account*.

Installing a Remote Server

Once you configure the remote server account to run McIDAS-X, follow the steps in the section called *Installing the McIDAS-X ADDE Remote Server*. The remote server shell script provided with the McIDAS-X package makes the configuration changes that allow the account to act as a remote server.

The next page describes how to remove the remote server account.

If, instead, you want to proceed with making data available on your remote servers proceed to the following page, *Acquiring and Naming Data on a Remote Server*.

Removing a Remote Server

To remove the remote server account from the workstation or make a different account the remote server, you must first uninstall the remote server using the same shell script. Perform steps 1 through 3 in the section called *Installing the McIDAS-X ADDE Remote Server*.

Acquiring and Naming Data on a Remote Server

When the remote server account is configured, McIDAS-X clients can request data from it. To make the data available, you must acquire it and then create dataset names. See the corresponding sections below for details.

Acquiring Data

When a client requests data, the remote server searches for the dataset in, by default, the **$HOME/mcidas/data** and the **~mcidas/data** directories, respectively. If you have datasets in another directory, use one of the following methods to make them accessible to clients.

- Add the directory to the remote server account's **MCPATH** environment variable.
- Use the McIDAS REDIRECT command to add a path redirection to the directory.
- If the directory is on another workstation, NFS mount the directory and use **MCPATH** or REDIRECT.

If you use **MCPATH**, you must explicitly define its value in the remote server account's **$HOME/.mcenv** file and source it in the remote server account's **$HOME/.profile** file. For example, add a line similar to the following in the **$HOME/.mcenv** file:

```
export MCPATH;
MCPATH=$HOME/mcidas/data:/alldata:/u/demo/images:~mcidas/data:~mcidas/help
```

To set the **MCPATH** value each time you log in to the remote server account, add the following line to the **$HOME/.profile** file. Be sure to leave a space between the period (.) and **$HOME**.

```
. $HOME/.mcenv
```

If you use the REDIRECT command, you must run it in a McIDAS-X session started from the remote server account. The examples below make areas 100 to 199 in the directory **/alldata** and areas 8000 to 8999 in the directory **/u/demo/images** accessible to ADDE clients.

REDIRECT ADD AREA01* "/alldata

REDIRECT ADD AREA8* "/u/demo/images

You can start a McIDAS-X session, run your REDIRECT commands and then exit the session. The client requests will find the data in the REDIRECT and **MCPATH** directories even if the remote server is not running a McIDAS-X session.

Naming Data

Since clients request the data by name, you must also name the datasets with the DSSERVE command. DSSERVE assigns the group and descriptor to a dataset consisting of one or more files of the specified format.

See the DSSERVE command documentation for detailed information and examples.

Setting the ADDE Server Timeout

By default, ADDE servers timeout if the client application (that requested the data) does not read any of the data for a period of 120 seconds. You can change the server timeout period with the environment variable **ADDETIMEOUT**. To do so, complete the steps below.

1. Add the following line, where *numseconds* is the timeout period in seconds, to the remote server account's **$HOME/.mcenv** file.

```
export ADDETIMEOUT; ADDETIMEOUT=numseconds
```

2. Examine the remote server account's **$HOME/.profile** file and add the following line if it's not already present. Be sure to leave a space between the period (.) and **$HOME**.

```
. $HOME/.mcenv
```

Setting the Tracking Mode for Area and MTSAT Servers

If your remote server receives area or MTSAT format data (via SDI ingestor, **ftp** from another server, or some other method), you can choose whether to track the data. Tracking means that clients that request data while it's being received by the server will wait until all of the requested data is available before completing the request. For example, if a client runs an IMGDISP command while the area, GVAR or MTSAT data is being received on the server, the IMGDISP command will continue running until it receives all of the requested data.

The area, GVAR and MTSAT servers are set, by default, to track data being received. You can turn off the tracking option in the area and MTSAT servers (but not GVAR) so that client commands that request area data that's being received by the server will obtain only the data that had been received when the command was issued. To turn off the tracking option, complete the steps below.

1. Add the following line to the remote server account's **$HOME/.mcenv** file.

```
export MCTRACK; MCTRACK=NO
```

2. Examine the remote server account's **$HOME/.profile** file and add the following line if it's not already present. Be sure to leave a space between the period (.) and **$HOME**.

```
. $HOME/.mcenv
```

To reset the tracking option to the default (on), remove the line listed in step 1 above from the remote server account's **$HOME/.mcenv** file.

Note: In McIDAS-X version 2012.1 the client environment variable **MCTRACK=YES/NO** and IMGCOPY, IMGDISP, IMGFILT, IMGMAG, IMGOPER, IMGPLOT, IMGPROBE, IMGREMAP, IMGRGB and IMGTRACK command keyword TRACKING=YES/NO were added to allow the client to **not** track area, GVAR or MTSAT data even if the server has tracking enabled. For example, if your GVAR server (which has tracking enabled) receives an IMGDISP command with TRACKING=NO, the server will send only the data that was ingested when the command was issued. See Setting the Image Tracking Mode for more information about the client options.

Updating Client Routing Tables

When an ADDE command is run, the client checks its routing table to determine where to route the data request. If you add a remote server, you must use the DATALOC command to update the routing table on each client using the server's data.

Use one of the following methods to update the client routing tables.

- Start a McIDAS session on each McIDAS-X client account and run the DATALOC commands.
- Place the DATALOC commands in a text file that each client can access. Start a McIDAS session on each client and use the text file with the BATCH command to run the DATALOC commands.
- Create or update a site routing table that contains the routing information for your site's clients.

The site routing table is the easiest way to modify routing tables for a large number of McIDAS-X clients. It allows the site administrator or operations staff to maintain a single table that is accessible to all clients at the site. The site routing table can be created and updated in any McIDAS-X session by using the DATALOC command to link group names to the IP addresses of remote servers. You may need to copy the table to a directory that is accessible to all clients. If a client is on another workstation, the client must NFS mount the directory to access the file.

The default file name for all client routing tables is **MCTABLE.TXT**. To use a different name when creating or updating the site table, you must specify the file name in the **MCTABLE_WRITE** environment variable value of the account running the McIDAS-X session. Write-protect the file so it can't be modified or deleted by your clients.

To make the site routing table accessible to your McIDAS-X clients, each client's **MCTABLE_READ** environment variable value must be modified to include this table. See *Client and Server Look-up Tables* for detailed information about the **MCTABLE_READ** and **MCTABLE_WRITE** environment variables.

Updating Clients to Use Compressed Data Transfers

Remote servers running McIDAS-X versions 2003 or later can transfer data to McIDAS-X clients using three different data transfer methods. The methods are gzip-compressed, compressed, and uncompressed.

The client determines which method is used by setting a flag in the request header. If the client is running McIDAS-X 2003, requests for gzip-compressed data are sent to port 112, compressed to port 503, and uncompressed to port 500. If the client is running McIDAS-X 2004 or later, all requests, regardless of the data transfer method, are sent to port 112. The remote server then gathers the requested data and uses the appropriate compression method to package the data, then sends it to the client through the same port.

If the remote server is running version 2003 or 2004, ports 112, 500 and 503 are open so that requests from clients running version 2003 or older are accepted. If the remote server is running version 2005 or later, only port 112 is open (ports 500 and 503 are NOT open). Therefore, remote servers running 2005 or later will respond only to data requests from clients running version 2004 or later (and also 2003 IF the **MCCOMPRESS** environment variable is set to **gzip**). Thus, commands sent to server port 500 or 503 from clients running older versions will fail on the client (eventually timeout because of no response).

Sites with firewalls or other screening mechanisms must open the needed ports on their remote servers (port 112 on servers running 2005, and ports 112, 500 and 503 on servers running 2003 or 2004) if they want to allow their clients to access the server's data.

The gzip-compressed and compressed methods are most beneficial to clients who transfer large data files or those with slow data connections. However, they only reduce the amount of data during the transfer; they do not change the resultant file size or accounting (the byte counts remain the same). For example, the destination file size from an IMGCOPY command is the same no matter which of the three methods is used.

For a client to use the gzip-compressed or compressed methods, the user may set the environment variable **MCCOMPRESS** in their account or use the MCCOMPRESS global keyword. See *Using Compressed Data Transfers* for instructions. Other than firewall precautions, no special actions are required on the remote server. The three data compression options are automatically implemented when you run the **mcinet***version#***.sh** script as part of the normal McIDAS-X installation. See *Installing the McIDAS-X ADDE Remote Server*.

To allow your McIDAS-X users to take advantage of all three data transfer methods, first install McIDAS-X 2003 or later on ALL your remote servers. Notify the users that your remote servers are updated and that they can configure their McIDAS-X sessions to request gzip-compressed or compressed data transfers by completing the instructions in *Using Compressed Data Transfers*. Warn them that they should NOT make those changes unless their client workstation and ALL of the remote servers they access are running McIDAS-X 2003 or later. Before updating your remote servers to version 2005, you should give your users sufficient notice that they need to upgrade their client workstations to version 2004 or later (or 2003 IF their **MCCOMPRESS** environment variable is set to **gzip**) in order to access the data on your servers.

Activating Transaction Logging on a Remote Server

Remote servers can use transaction logging to monitor data requests that the server receives from McIDAS-X clients. This information is written into the file **SERVER.LOG**, which resides in the first writable directory of the environment from which the server is run.

The following items are logged.

Field	Bytes	Type	Description
1	4	unsign	Server IP address
2	4	int	Server port used
3	4	unsign	Client IP address
4	4	char	User ID
5	4	int	Project number
6	12	char	Password
7	4	char	Transaction type
8	4	int	Number of bytes received by server
9	120	char	ADDE request string
10	4	int	Number of bytes going out
11	4	int	100 x CPU seconds
12	4	int	Return code
13	72	char	Error message
14	4	int	Start date in IYD format
15	4	int	Start time of transaction in hhmmss
16	4	int	End time of transaction in hhmmss

Transaction logging is activated by setting the environment variable **ADDE_LOGGING** in the remote server account before the server starts. Complete the steps below to do this automatically each time you log in to the remote server account.

1. Add the following line to the remote server account's **$HOME/.mcenv** file.

```
export ADDE_LOGGING; ADDE_LOGGING=YES
```

2. Examine the remote server account's **$HOME/.profile** file and add the following line if it's not already present. Be sure to leave a space between the period (.) and **$HOME**.

```
. $HOME/.mcenv
```

WARNING!
The SERVER.LOG file will continue to grow in size until it is truncated or removed. High access volume servers should schedule the file to be rotated or truncated to prevent filling of the file system.

To deactivate transaction logging, remove the line listed in step 1 above from the remote server account's **$HOME/.mcenv** file.

Restricting Access to Remote Server Datasets

By default, when you configure a remote server there are no restrictions to accessing its datasets. Thus, any McIDAS-X client with a client routing table entry for your server (i.e., an entry that maps a group name that exists on your server to your server's IP address) can access its data. This section describes how to configure your remote server to restrict access to all of its datasets, or to specific datasets identified by group or group/descriptor.

Access to one or more of the server's datasets can be restricted to the following.

- authorized user IDs (entered with the McIDAS LOGON command)
- authorized four-digit project numbers (also entered with LOGON)
- authorized client IP addresses

There are three types of files used to allow/restrict access to the server's datsets: Server Files, Group Files, and Group.Descriptor Files. The files must adhere to these characteristics/requirements:

- The files must reside in a **MCPATH** or REDIRECT directory of the remote server account (normally **mcadde**).
- The files are standard ascii text format, created with any text editor.
- Each file may have any free format comment on each line, after the required information, which must begin in column 1. At least one space should separate the required information from the comment.
- Lines beginning with an asterisk (*) are considered comments, and are thus ignored.

Server Files, Group Files, and Group.Descriptor Files are described in further detail below. **Important note:** These files are enabled (i.e., perform their described actions) only if transaction logging is activated. See the previous section, *Activating Transaction Logging on a Remote Server*, for instructions.

Server Files

To be allowed access to all datasets on a server, the user must have a valid entry in one of the three files listed below, if the file exists. If any of the files required for validation is missing, that type of validation is not performed.

The three types of Server Files are listed below.

- **SERVER.IP** - contains IP addresses of trusted machines in dotted decimal format
- **SERVER.USR** - contains user initials, one per line in uppercase; each record must be at least four characters long
- **SERVER.PRJ** - contains four-digit project numbers, one per line

Group Files

To be allowed access to all datasets in a particular group on a server, the user must have a valid entry in one of the three files listed below or one of the Server Files described above. If any of the files required for validation is missing, the server will then check if the user is valid based on the Server Files.

The three types of Group Files are listed below.

- ***GROUP*.IP** - contains IP addresses of trusted machines in dotted decimal format
- ***GROUP*.USR** - contains user initials, one per line in uppercase; each record must be at least four characters long
- ***GROUP*.PRJ** - contains four-digit project numbers, one per line

For example, to allow users logged on to McIDAS as user JOHN access to all datasets in the group GOES, the file **GOES.USR** must contain a line that says "JOHN".

The Server Files are used in conjunction with the Group Files. For example, if the files **SERVER.IP** and **SERVER.PRJ** also exist, the user JOHN must be accessing the data from a valid IP address, and using a valid project number in those files.

Multiple files with duplicate extensions can also exist. For example, if the **MSG.IP** and **SERVER.IP** files exist, you can configure them to allow IP address 144.92.109.205 access only to datasets in group MSG while also allowing IP address 128.104.110.92 access to all datasets. To do so, the file **MSG.IP** must contain 144.92.109.205 and file **SERVER.IP** must contain 128.104.110.92.

Group.Descriptor Files

To be allowed access to a particular dataset (group and descriptor, e.g., GOES/CONUS) on a server, the user must have a valid entry in one of the three files listed below or one of the files described above. If any of the files required for validation is missing, the server will then check if the user is valid based on the Server Files and Group Files.

The three types of Group.Descriptor Files are listed below.

- ***GROUP.DESCRIPTOR*.IP** - contains IP addresses of trusted machines in dotted decimal format
- ***GROUP.DESCRIPTOR*.USR** - contains user initials, one per line in uppercase; each record must be at least four characters long
- ***GROUP.DESCRIPTOR*.PRJ** - contains four-digit project numbers, one per line

For example, to allow users logged on to McIDAS as user JOHN access to only dataset MSG3HR/HRV (and no other datasets on the server), the file **MSG3HR.HRV.USR** must contain a line that says "JOHN".

Configuring MODIS Servers to use MRTSwath for Bowtie Correction

A subset of the MODIS ADDE image servers in McIDAS-X and -XRD version 2011.1 and later contain the ability to correct the "bowtie effect" in MODIS imagery. The correction is activated by the user by including the keyword NAVTYPE=RECT in their IMGCOPY, IMGDISP, IMGFILT or IMGOPER command. See below for more information about the bowtie effect and how to configure your MODIS ADDE image servers to apply the correction using the MRTSwath software package.

- Introduction
- Installation
- Implementation
- Limitations

Introduction

The scanning geometry associated with the MODIS instrument introduces oversampling artifacts at the far ends of each swath of data. This is commonly referred to as the "bowtie effect". These artifacts can be eliminated from the image by remapping the native satellite projection into a standard geographic projection.

A common application used to perform this reprojection is **MRTSWath** (MODIS Reprojection Tool - Swath, package courtesy of the U.S. Geological Survey). At the heart of MRTSwath is the swath2grid executable which can be run using a simple command line interface. It is this interface that forms the harness linking the McIDAS ADDE service to MRTSwath.

Installation

A subset of the MODIS ADDE image servers in McIDAS-X and -XRD version 2011.1 and later contain modifications that allow them to use the MRTSwath extension and its swath2grid executable. There are no special installation instructions for the servers; the mechanism used to run swath2grid only requires that the MRTSwath environment be installed within the scope defined by MCPATH. The package can be obtained from the Land Processing DAAC at the following address:

> https://lpdaac.usgs.gov/lpdaac/tools/modis_reprojection_tool_swath

The MRTSwath package is self contained with support for numerous platforms / operating systems. At SSEC, we have installed and tested the package on OS X, Linux32 and Linux64 systems. During the installs we specified **/usr/local/MRTSwath** as the installation directory and accepted the path to the Java binary when prompted. We recommend following those steps so the installer sets the environment (e.g., adds **/usr/local/MRTSwath/bin** to the **PATH** and sets the **MRTSWATH_HOME** and **MRTSWATH_DATA_DIR** environment variables) to run the program successfully.

Implementation

Not all MODIS ADDE image servers contain the MRTSwath extension; see the *Limitations* section below for more details. The servers that do allow MRTSwath reprojection do so by spawning a child process from the ADDE server. Execution of the ADDE server is suspended while the reprojection tool runs, producing a binary file containing the corrected image. Once the reprojection is completed, control is passed back to the ADDE server and processing continues. The returned image is a remapped version of the original data that resides on disk. Thus, IMGDISP of MODIS data using the reprojection tool will display an image but the digital data used to create the image only resides in memory, not on disk. This means that users who want to use the IMGPROBE command with the corrected image will need to use IMGCOPY with NAVTYPE=RECT to copy it to the local disk, then display it with IMGDISP and then probe it with IMGPROBE.

McIDAS users can request bowtie corrected image segments through ADDE client applications IMGCOPY, IMGDISP, IMGFILT and IMGOPER by adding keyword NAVTYPE=RECT to their command request. Specifying NAVTYPE=RECT signals the MODIS ADDE image servers to invoke swath2grid to perform a bowtie correction and reprojection of the data from satellite to Rectilinear projection.

Limitations

A subset of the MODIS ADDE image servers have been extended to permit users to specifically request bowtie corrected image segments from the online datasets. MRTSwath's resolution restrictions limit the number of Level 2 products that can be reprojected. See below for the list of MODIS ADDE image servers that have been extended to request MRTSwath bowtie correction if the user specifies the NAVTYPE=RECT keyword in their IMGCOPY, IMGDISP, IMGFILT or IMGOPER command.

- modsaget.c - McIDAS-X Level 1b image server
- mod8aget.c - McIDAS-X Sea Surface Temperature (SST) server
- modcaget.c - McIDAS-XRD Cloud Mask image server (version 2 products from SSEC/CIMSS only)

- modiaget.c - McIDAS-XRD Normalized Difference Vegetation Index (NDVI) server
- modlaget.c - McIDAS-XRD Land Surface Temperature (LST) server

The image files used with MRTSwath must match the MODIS direct broadcast naming convention (e.g., **a1.yyddd.hhmm.1000m.hdf** and **t1.yyddd.hhmm.500m.hdf**) and the MODIS navigation files (e.g., **a1.yyddd.hhmm.geo.hdf** and **t1.yyddd.hhmm.geo.hdf**) must also be present in the same directory.

Although both McIDAS and MRTSwath provide numerous geographic projections, inconsistencies between the definitions within each system tend to introduce warping. To avoid these errors, the servers are limited to a Rectilinear transform. Because of the way the Rectilinear projection is defined, it eliminates the potential for computational errors when moving from MRTSwath to McIDAS.