McIDAS-XCD

Administrator's Guide

Issued October 1994

Revised May 1997

includes May 1998 update

Space Science and Engineering Center
University of Wisconsin-Madison
1225 West Dayton Street
Madison WI 53706
Telephone (608) 262-2455
FAX (608) 263-6738

Table of Contents

Introduction to McIDAS-XCD

The McIDAS-X Conventional data Decoder (McIDAS-XCD) enables workstations running McIDAS-X to directly receive and process data from the National Weather Service Family of Services. All of the operational McIDAS-XCD client commands for accessing conventional data have been removed from the McIDAS-XCD package. These commands have been replaced with ADDE (Abstract Data Distribution Environment) commands distributed with McIDAS-X. See the *McIDAS-X User's Guide* for more information.

This chapter provides an introduction to McIDAS-XCD, including:

- definitions of common terms
- an explanation of how McIDAS-XCD receives and processes conventional data
- a description of the McIDAS-XCD Status window

Terminology

The terms defined below are used throughout this manual.

Term	Definition
client	workstation that requests and receives data from a server workstation
data block	WMO header description and text data
data monitor	process that runs one or more decoders
DDS	Domestic Data Service
decoder	program that converts raw data into McIDAS data files
HRS	High Resolution Data Service
IDS	International Data Service
ingestor	program that receives data through a communications port
NFS	Network File System
PPS	Public Products data Service
server	workstation that stores and supplies data to client workstations

Data receiving and processing

McIDAS-XCD uses ingestors and data monitors to receive and process asynchronous data from the National Weather Service (NWS) Family of Services. The data arrives via satellite broadcast by either an outside vendor or a dedicated phone line directly from the circuit source.

Ingestors

An ingestor is a program that reads data entering the system through a communications port. Ingestors read asynchronous data from conventional data circuits such as DDS, IDS and PPS.

Each circuit has a text formatted configuration file that the ingestor reads to configure the communications port. This configuration file resides in **~oper/mcidas/data** and is usually named with the circuit name followed by .CFG, for example, DDS.CFG. Figure 1 on page 3 is an example of the DDS configuration file. It contains information such as baud rate and the number of data bits or stop bits. McIDAS-XCD supports both text and binary ingestors.

Text ingestors

A text ingestor receives data from one of the Family of Services data circuits (DDS, IDS, PPS) in ASCII format. Each incoming circuit has its own text ingestor that writes to a set of raw text files and index files. Index files contain the location information of data in the raw text file.

For example, Figure 2 on page 4 shows that the text ingestor INGETEXT ingests data from one of the Family of Services data circuits. Each ingested data block is placed in a circuit-specific raw text file for that day. INGETEXT also files information about the data block into an index file which is used by text applications and decoders for locating data quickly.

The naming convention for the circuit-specific raw text file is *ccyyddd0*.XCD, where *cc* is the first two characters of the circuit name, and *yyddd* is the Julian day. The naming convention of the index file is *hhyyddd*.IDX where *hh* is a 2-character WMO header and *yyddd* is the Julian day.

While only one text ingestor can write into a text file, any text ingestor can write to any index file. For example, terminal forecasts (FTs) arrive on the DDS and IDS circuits. The actual forecasts are filed in DD*yyddd*0.XCD and ID*yyddd*0.XCD, respectively. However, both ingestors write their directory information into the same index file, FT*yyddd*.IDX. This ensures that applications will work consistently on similar data formats regardless of the data's source.

Index files also store related data that arrives under more than one WMO header. For example, mandatory upper level RAOB reports come in under the headers UJ, US, UK, UL, etc. Rather than having a separate index file

for each header, a routing table is created during installation to tell the ingestors where to file the headers from each circuit. For example, the index file for all RAOB WMO headers is UJ*yyddd*.IDX. If a WMO header is not forced into a particular index file, it is filed in the ZZ*yyddd*.IDX file, which is a miscellaneous index.

Binary ingestors

The binary data ingestor, INGEBIN, ingests a binary data stream regardless of the data format. INGEBIN stores the data in a circular spool file, *ccc*.SPL, where it can be processed by a data monitor. The data that INGEBIN ingests includes HRS data sent by the NWS. This data is in the GRIB message format. See Chapter 5, *Decoding GRIB Messages* for more information.

```
#DDS.CFG              serial communications configuration script for the DDS
#                     (Domestic Data Service) circuit
#--------             PORT=should be only one each site must modify

PORT=/dev/ttyc0 #port tty name (this will vary among workstations)
_____________________________________________________________________


IBAUD=19200           #  input baud rate of the circuit
OBAUD=19200           #  output baud rate of the port
CSTOPB=1              #  number of stop bits
CSIZE=8               #  number of data bits
VTIME=0               #  block indefinitely
VMIN=40               #  block until at least 40 bytes have been read
BRKINT=YES            #  generate SIGINT on BREAK
CLOCAL=NO             #  block until the modem is answered
CREAD=YES             #  receiver enabled and characters can be read
ECHO=NO               #  don't echo characters back to device
ECHOE=NO              #  don't visually erase characters
ECHOK=NO              #  don't echo KILL
ECHONL=NO             #  don't echo new line
HUPCL=NO              #  don't hang up line when last process closes the device
ICANON=NO             #  disable canonical mode
ICRNL=NO              #  do not convert carriage returns to line feeds
IEXTEN=NO             #  disable special extended character recognition
IGNBRK=NO             #  don't ignore BREAK condition
IGNCR=NO              #  do not ignore carriage returns
IGNPAR=NO             #  do not ignore characters with parity errors
INLCR=NO              #  do not convert new line character to carriage return
INPCK=NO              #  don't enable input parity error checking
ISIG=NO               #  disable terminal-generated signals caused by special
                      #  characters
ISTRIP=YES            #  strip input to 7 bits
IXOFF=NO              #  disable start-stop input control
IXON=NO               #  do not enable start-stop output control
NOFLSH=NO             #  flush queues when SIGNT and SIGUIT are sent
OPOST=YES             #  perform output processing
PARENB=NO             #  disable parity error checking
PARMRK=NO             #  don't mark parity errors
PRODD=NO              #  parity error checking set to even (not used)
TOSTOP=NO             #  don't send SIGTTOU for backgroup output
```

Figure 1. DDS Configuration File

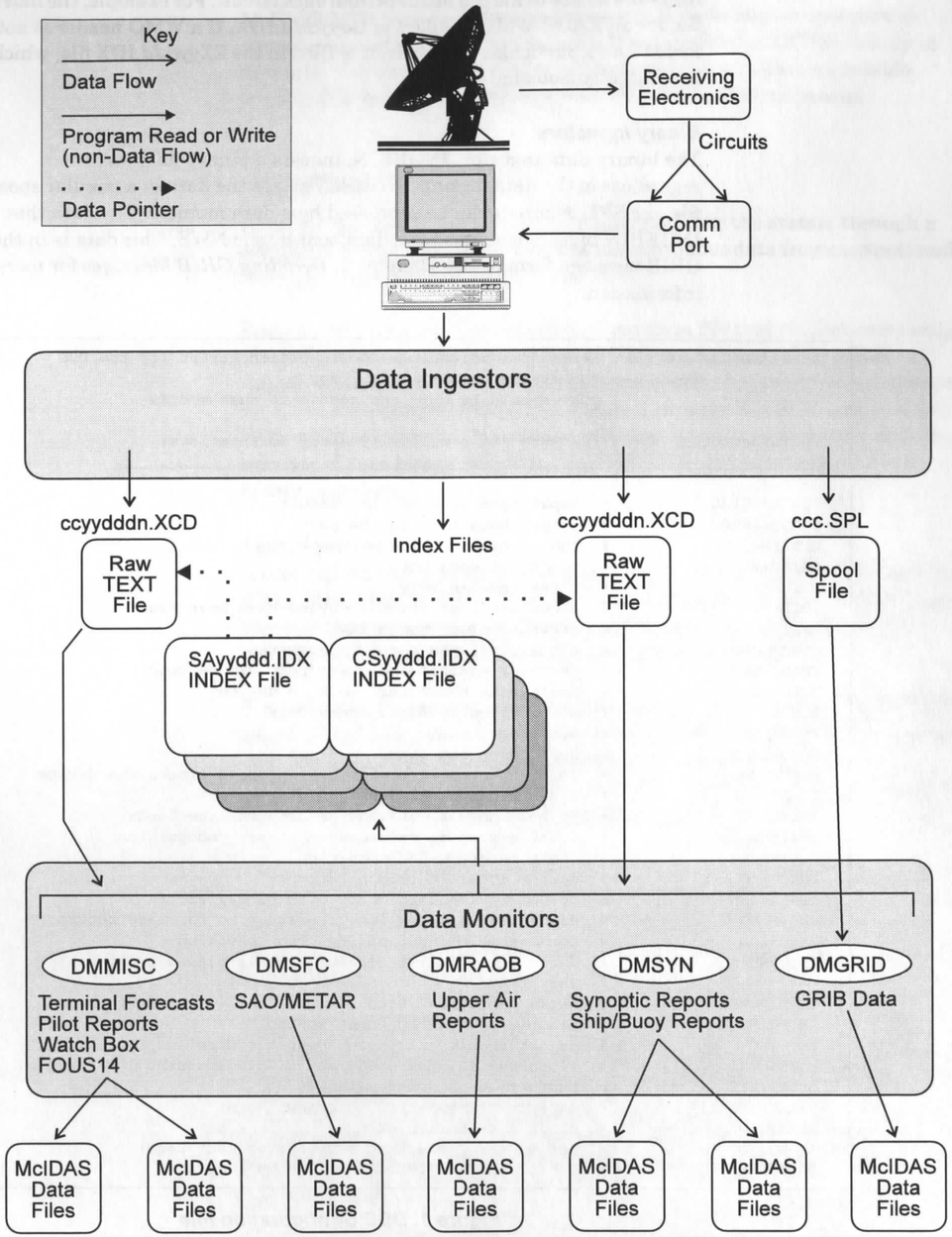

Figure 2. Conventional Data Ingesting Flow Diagram

Data monitors

A data monitor is a process that runs one or more decoders. After the data is stored in the raw text or spool files, data monitors process the raw data into McIDAS files such as surface hourly observations, synoptic reports, upper air reports, and grid files.

The following table lists the supported McIDAS-XCD data monitors.

Monitor	Index	Data type	Decoded Data location	Configuration file
DMGRID		GRIB	Grid files	GRIBDEC.CFG
DMMISC	FO	FOUS14	MD file FO14	FO14DEC.CFG
	FT	Terminal Forecast	Rapid-Access System	TERMDEC.CFG
	SD	MDR	Grid files	MDRDEC.CFG
	TB	TIROS NAV	SYSNAV1	TIRDEC.CFG
	UA	PIREP/AIREP	MD file PIRP	PIRPDEC.CFG
DMRAOB	UJ	TEMP/PILOT	MD file IRAB/IRSG	IRABDEC.CFG
DMSFC	SA	SAO/METAR	MD file ISFC	ISFCDEC.CFG
DMSYN	SM	SYNOPTIC	MD file SYN	SYNDEC.CFG
	SM	SHIP/DRIBU	MD file ISHP	ISHPDEC.CFG

To make the data monitors more flexible, each is designed to use a text formatted configuration file similar to those used for circuit configuration. This configuration file contains the .IDX files to search, WMO headers to decode, decoder display number, MD file numbers to store data, etc.

Figure 3 on page 6 shows an example of the configuration script file for the FOUS14 decoder.

```
#  FO14DEC.CFG   - Configuration file for the FOUS14 decoder

#------------Cross Reference List (do not change)---------------------
: FLAGS[01] ERRORFLG: FLAGS[02]
: FLAGS[03] DISPLAYNUM
: FLAGS[04] MDF
: FLAGS[05] NROWS
: FLAGS[06] NCOLS

: CFLAGS[01] ERRORFILE
: CFLAGS[02] OLDIDFILE
: CFLAGS[03] NEWIDFILE
: CFLAGS[04] IDTABLE
: CFLAGS[05] MASTERFILE
#  -----------------End Of Cross Reference List-----------------------
#  -----------------You can modify any of the fields below-------------
# decoder description
DESCRIPTION="FOUS14 Decoder
# which indices to search for this decoder
INDEX=FO
# which specific wmo headers to activate the decoder for
WMO=FOUS
MINPRD=14
MAXPRD=14
# which specific station origins to activate the decoder for
ORIGIN=KWBC
ERRORFLG=0                  # error output flag set to 1 to activate
ERRORFILE=FO14DEC.ERR       # error file name
IDMONFLG=0                  # station id monitoring activation flag
                            # set to 1 or 3 to monitor new stations
                            # set to 2 or 3 to monitor old stations
OLDIDFILE=OLDFO14.IDM       # old station id file used for monitoring
NEWIDFILE=NEWFO14.IDM       # new station id file used for monitoring
DISPLAYNUM=5                # decoder number on status display
MDF=41                      # first real-time md file number to use for decoder
NROWS=38                    # number of rows to make for md file
NCOLS=350                   # number of columns to make for md file
IDTABLE=FO14DEC.IDT         # ID file to build when creating md file
MASTERFILE=LOCALID.DAT      # master ID table file to use to build IDTABLE
```

Figure 3. FOUS14 Configuration File

McIDAS-XCD Status window

The McIDAS-XCD Status window is displayed during your McIDAS-XCD session. It lists information about the data processed by the ingestors and data monitors such as:

- the data arriving on each circuit
- the last time data was received
- the data currently being processed and filed
- the last time data was processed

A sample decoder status display is shown below; the table on the next page defines each field in the display. The sample display indicates that the DDS ingestor last filed data at byte 275209 of DD942860.XCD, and that it last filed data in index location 3956 of index file UA942286.IDX. The IPS circuit filed data in FP94286.IDX.

The example also shows the surface decoder (SAODEC) last updated the bulletin board at 23:48:45 UTC and the most recent index location processed by SAODEC was at location 53212. It indicates that SAODEC continues processing data until at least index location 53224. After it processes 53224, it re-reads SA94286.IDX to determine if it should continue processing. If no new data is received, the decoding task, DMSFC, pauses for approximately 30 seconds and then checks if any new data has arrived. The example also indicates that the last observation filed data in MD file 6, row 67, column 200.

The RAOB decoder, RABDEC, indicates that all the data from the appropriate index file (UJ94286.IDX) was processed because the index pointers (3108) are identical.

```
Decoder Status Display    94286  234851
##  CIRCUIT   INGESTOR  TIME    BYTE    INDEX   FILENAME      ORIGIN   WMO   PRODUCT
1   DDS       INGETEXT  234757  275209  3956    UA94286.IDX   KWBC     UAAK  3
2   IDS       INGETEXT  234849  148090  8476    FP94286.IDX   KAWN     FPUS  12
3   HRS       INGEBIN   234819  123456          HRS.SPL

                                                GRIDF  GRID
##  DECODER   TIME    BEGPTR  LASPTR  MD      ROW    COL   TEXT          INDEX
1   SAODEC    234845  53212   53224     6     67     200                 SA94286.IDX
2   RABDEC    234830   3108    3108    26     14       3                 UJ94286.IDX
3   SYNDEC    234836  12451   12452    56      8    5011                 SM94286.IDX
4   WBXDEC    234700    512     512                      Watch #23  WW94286.IDX
10  GRIB      234000      0       0   701     19         HZBC 70 KWBC 40000
```

Figure 4. Sample Decoder Status Display

The table below defines each field in the decoder status display.

Field	Definition
##	ingestor or decoder number
CIRCUIT	circuit receiving the data
INGESTOR	ingestor command name
TIME	time the data was last received
BYTE	last byte number the ingestor wrote
INDEX	last directory location the ingestor filed; not used by INGEBIN
FILENAME	index file name last written to; for INGEBIN this field displays the spool name
ORIGIN	origin of the last block filed; the value is extracted from the WMO header; not used for INGEBIN
WMO	product header of the last block filed; not used by INGEBIN
PRODUCT	WMO product number of the last block filed; not used by INGEBIN
DECODER	decoder name
TIME	time data was last processed
BEGPTR	current index location being decoded
LASPTR	last index location decoder processes before checking for more data
GRIDF/MD	last GRID or MD file the decoder wrote to
GRID/ROW	last GRID number or ROW number written to
COL	last MD column number written to
TEXT	text description of the decoder process

McIDAS-XCD Software Installation

The McIDAS-XCD software installation process makes the directories **~oper/mcidas/xcd7.4**, **~oper/mcidas/xcd7.4/src** and **~oper/mcidas/xcd7.4/data**, places the source and data files in them, and builds the software. When the build is complete, the source, helps, data, and binaries are copied to the directories **~oper/mcidas/src**, **~oper/mcidas/help**, **~oper/mcidas/data**, and **~oper/mcidas/bin**, respectively.

Check the system requirements before installing the McIDAS-XCD software package on your McIDAS-X workstation. Then use the instructions that follow to install McIDAS-XCD.

System requirements

- The McIDAS-XCD software package runs on IBM RISC System/6000, SGI, Sun SPARC and HP/Apollo 9000 series 700 workstations running McIDAS-X version 7.4. McIDAS-X must be installed in the **mcidas** account according to the specifications in Chapter 1, *Installation and Configuration*, in the *McIDAS User's Guide*. Be sure to include the directory **~mcidas/bin** in the environment variable **PATH**.
- The Unix workstation running the McIDAS-XCD software must have the group name **mcdata** which contains the user **oper**.
- The Unix workstation running the McIDAS-XCD software must have the user account **oper**. This account must be configured with the appropriate directories, links, and paths to run McIDAS-X. For more information, see the section titled *Configuring A New User Account* in Chapter 1 of the *McIDAS User's Guide*. If you already have an **oper** account on your workstation and do not want to run the -XCD package under this account, contact the McIDAS Help Desk (608) 262-2455.
- The workstation requires one asynchronous port for each circuit ingesting data. If the workstation does not have enough asynchronous ports, you must obtain third party hardware that allows for more. SSEC recommends the ST1008+ from Central Data. For more information, contact SSEC.

In addition, the workstation must have enough disk space to run the McIDAS-XCD software package. The table below lists the system space requirements per day for each circuit and data type decoded with the -XCD package.

Circuit/data type	Daily space requirements
DDS circuit	75 MB
IDS circuit	24 MB
PPS circuit	20 MB
Surface hourly MD file (ISFC)	25 MB
RAOB MD file (IRAB/IRSG)	7 MB
Synoptic MD file (SYN)	7 MB
Ship/buoy MD file (ISHP)	4 MB
FOUS14 MD file (FO14)	2 MB
PIREP/AIREP/ACARS (PIRP)	5 MB
Approximate total	169 MB (plus an extra 250 MB for other -XCD files)

If you process all the grids in GRIB data, the GRIB decoder requires an additional 500 MB per day.

For example, assume your site receives the three circuits and decodes all the data sources above. To store six days of MD data online, your minimum space requirement will be 1014 + 250 = 1264 MB. If you also receive and store two days of GRIB data, your minimum space requirements will be 2264 MB.

First-time installation procedures

If you are updating an existing version of McIDAS-XCD, skip this section and go to the *Installation procedures* section on the next page. If this is your first installation of McIDAS-XCD, perform the following tasks:

- adding the **mcdata** group
- assigning directory permissions

Adding the mcdata group

The workstation running the McIDAS-XCD software must have the group name **mcdata** which contains the user **oper**. Use this procedure to add the **mcdata** group, if needed.

1. Log on to the **root** account.
2. Add the following line to the **/etc/group** file. Replace *groupid* with a unique group ID number.

```
mcdata::groupid:oper
```

3. Log out of the **root** account.

Assigning directory permissions

Use the steps below to assign write privileges to the directory **~mcidas/data**.

1. Log on to the workstation as user **mcidas** and change the group for the **~mcidas/data** directory to the group **mcdata**.

   Type: **chgrp mcdata ~mcidas/data**

2. Change the privileges for the directory **~mcidas/data** so only the user **mcidas** or the members of the group **mcdata** have write permissions.

   Type: **chmod 775 ~mcidas/data**

3. Log out of the **mcidas** account.

Installation procedures

The McIDAS-XCD software installation consists of eight tasks:

- obtaining the McIDAS-XCD software package from the MUG Web Site, or the provided tape or CD
- loading the software
- configuring the McIDAS-XCD files
- configuring the communications port
- activating the GRIB decoder
- starting the McIDAS-XCD package
- configuring the **mcadde** account

Obtaining the McIDAS-XCD software

Use one of the following procedures to copy the McIDAS-XCD files to your workstation: *Obtaining McIDAS-XCD from the MUG Web site* or *Obtaining McIDAS-XCD via tape or CD*.

The McIDAS-XCD 7.4 package contains the following files.

File name	Description
`xcd7.4.tar.Z`	compressed tar file that contains all source and data files
`xcd_init`	shell script that initializes the environmental variables for the -XCD installation
`xcd_chksys`	shell script that checks for the proper setting of the environmental variables used during installation
`xcd_install`	shell script that installs the McIDAS-XCD software
`xcd_README_7.4`	lists information to review before installation

Obtaining McIDAS-XCD from the MUG Web Site

1. Use your Web browser to download the files listed on the previous page. Access the McIDAS User's Group Web Site at *http://www.ssec.wisc.edu/mug*, and follow the link for McIDAS-XCD software. Each site has its own login and password for downloading files. Contact the McIDAS Help Desk if you can't remember yours.
2. Log on to the McIDAS-XCD workstation as user **oper** and move the downloaded files to the **~oper/mcidas** directory.
3. List the files and check the ownership. If the -XCD files are owned by user **oper**, skip steps 4-6.

   Type: **`ls -l ~oper/mcidas`**
4. Change the ownership to user **oper**, if needed. You must have root permission to do this. Switch to user **root**.

   Type: **`su root`**
5. Change to the **~oper/mcidas** directory.

   Type: **`cd ~oper/mcidas`**
6. Run the command below for each of the downloaded -XCD files.

   Type: **`chown oper`** *file*

Obtaining McIDAS-XCD via tape or CD

1. Log on to the McIDAS-XCD workstation as user **oper**.
2. Change to the **~oper/mcidas** directory.

   Type: **`cd ~oper/mcidas`**
3. Insert the upgrade tape or CD in the drive and extract or copy the files.

   If you're using a tape, run a command similar to the one shown below. Specify *tapedevice* as the device name of your tape unit.

   Type: **`tar xvf /dev/`***tapedevice*

   If you're using a CD, run a command similar to the one shown below. Specify *cdrom* as the file system mountpoint of your CD unit. Note the period (**.**) at the end of the command.

   Type: **`cp /`***cdrom*`/xcd/*` **.**

Loading the McIDAS-XCD software

To begin this procedure, you should still be logged on as user **oper**. Before loading the software, be certain that the **PATH** environment variable contains the **~mcidas/bin** directory. The underscore (_) characters in the command lines below are part of the file names and must be typed.

1. Change the file permissions of the installation scripts to allow them to run.

   Type: **chmod 755 xcd_***

2. Run the shell script **xcd_init** to initialize the environmental variables **McIDAS_ROOT**, **McINST_ROOT**, and **McXCD_ROOT**. You must leave a space between the two periods (. .) when typing the command below.

   Type: **. ./xcd_init**

3. If this is the first installation of McIDAS-XCD on this workstation, run the shell script **xcd_install all** to build the McIDAS-XCD software. If you are updating McIDAS-XCD version 7.3 on this workstation, go to step 4.

   Type: **./xcd_install all**

   This script performs the following steps:

   - creates the directories **xcd7.4**, **xcd7.4/src**, and **xcd7.4/data** from the **~oper/mcidas** directory
   - uncompresses the file **xcd7.4.tar.Z**
   - compiles the source code and copies the binaries to the directory **~oper/mcidas/bin**; approximately 140 modules are compiled, so this step takes a few minutes to complete; your compiler may generate some warnings while the macro commands are compiling
   - copies data files to the **~oper/mcidas/data** and the **~mcidas/data** directories
   - copies the help files to the **~oper/mcidas/help** directory

When the script **xcd_install** is finished, you see the message below.

```
McIDAS-XCD package installation is now complete
```

Continue with step 7.

4. Exit the McIDAS session that is running the ingestors and data monitors. From the McIDAS Text and Command window,

   Type: **EXIT**

5. Run the shell script **xcd_install build** to build the McIDAS-XCD software. From an **oper** xterm,

   Type: **./xcd_install build**

   When the script **xcd_install build** is finished, you see the message below.

```
McIDAS-XCD binaries built correctly
```

6. Run the script below to install the new executable code and a subset of the necessary McIDAS-XCD data files for your workstation.

   Type: **./xcd_install cutover**

   When the script **xcd_install cutover** is finished, you see the message below.

```
McIDAS-XCD package cutover is now complete
```

7. Switch to user **mcidas** so you can install the McIDAS-XCD ADDE servers.

   Type: **su mcidas**

8. Run the script below to install the new ADDE server executable code in the **mcidas** account.

   Type: **./xcd_install addeservers**

9. Exit from user **mcidas**.

   Type: **exit**

Configuring the McIDAS-XCD files

To configure the -XCD files, start McIDAS-XCD from a McIDAS-X session. If you already have a McIDAS-X session running under the Unix login name **oper**, start with step 2. Enter commands exactly as shown; case is important.

1. Log on to the workstation as the user **oper**.

2. Determine the full Unix path of the **~mcidas/data** directory. Use this path in step 4. From an xterm,

   Type: **echo ~mcidas/data**

3. Start a McIDAS-X session.

   Type: **mcidas**

4. Create the McIDAS string MCDATA to contain the full Unix path of the **~mcidas/data** directory. From the McIDAS session,

   Type: **TE MCDATA "***/datadirectory*

   For example: **TE MCDATA "/home/mcidas/data**

5. Run the batch file **XCD.BAT**.

   Type: **BATCH "XCD.BAT**

   XCD.BAT redirects several data files, saves them in the redirection table XCD, and initializes the **GROUPS.DAT** and **COUNTRY.DAT** files.

   The message "**BATCH:DONE**" must be displayed before you can continue.

6. Run the batch file **XCDDEC.BAT**. This file restores the XCD redirection table, adds the server's redirections and saves the redirection table as XCDDEC. It also initializes some files, registers the required data schemas and builds the pointer files required for processing data.

   Type: **BATCH "XCDDEC.BAT**

   When the message "**BATCH:DONE**" appears, go to the next step.

7. List the active data circuits.

   Type: **CIRCUIT**

   The table below lists the default values of the circuits for the Family of Services data stream.

Circuit	Active	Comm. Port	Command	Configuration File
DDS	yes	/dev/ttyC0	INGETEXT	DDS.CFG
PPS	no	/dev/ttyC1	INGETEXT	PPS.CFG
IDS	no	/dev/ttyC2	INGETEXT	IDS.CFG
HRS	no	/dev/ttyC3	INGEBIN	HRS.CFG

8. If your workstation will receive and process data from the Family of Services IDS, HRS, and PPS circuits, use the following command to activate them.

   Type: **CIRCUIT SET IDS ACTIVE; CIRCUIT SET PPS ACTIVE; CIRCUIT SET HRS ACTIVE**

9. If your workstation will receive and process data from a NOAAPORT SDI ingestor, run the following command.

   Type: **BATCH NOAAPORT.BAT**

   This McIDAS batch file replaces the Family of Services circuits in the circuit configuration file with the NOAAPORT text and binary circuits, NTXT and NBIN. It also activates both circuits and configures the WMO header routings for the NTXT circuit. Both circuits must always be active. The table below lists the default values of the circuits for the NOAAPORT data stream.

Circuit	Active	File	Command	Configuration File
NTXT	yes	/tmp/jmb.fifo.1	INGETEXT	NTXT.CFG
NBIN	yes	/tmp/jmb.fifo.2	INGEBIN	NBIN.CFG

10. List the active data monitors, their associated decoders, and status.

    Type: **DECINFO**

    The table below lists the status of data monitors and decoders.

Data Monitor	Decoder	MD File	Status	Description	Configuration File
DMSFC	SAODEC	1-10	A	Surface hourly	`ISFCDEC.CFG`
DMRAOB	RABDEC	11-30	A	Upper air	`IRABDEC.CFG`
DMSYN	SYNDEC	51-60	A	Synoptic	`SYNDEC.CFG`
	SHPDEC	31-40	A	Ship/Buoy	`ISHPDEC.CFG`
DMMISC	F14DEC	41-50	A	FOUS14	`FO14DEC.CFG`
	PIRDEC	61-70	A	PIREP/AIREP	`PIRPDEC.CFG`
	TERDEC		A	Terminal Fcst	`TERMDEC.CFG`
	TIRDEC		I	TIROS NAV	`TIRDEC.CFG`
	MDRDEC		A	MDR grids	`MDRDEC.CFG`
DMGRID	GRIBDEC		I	NMC GRIDS	`GRIBDEC.CFG`

11. Use the DECINFO command to deactivate any data monitors and decoders you do not want running. Deactivating a data monitor will deactivate all decoders running. See Chapter 3, *McIDAS-XCD Administrative Commands* for more information.

Configuring data communications

If you are updating an existing version of McIDAS-XCD, skip this section and go to the *Activating the GRIB decoder* section. If this is your first installation of McIDAS-XCD, follow one of the procedures below.

If your data source is the Family of Services, follow the *Family of Services communications port configuration* procedure.

If your data source is the NWS NOAAPORT broadcast via an SDI ingestor, there are two options for circuit configuration. If your McIDAS-XCD software is on the same workstation as the NOAAPORT SDI ingestor, follow the *Local NOAAPORT circuit configuration* procedure. If your -XCD software is on a remote workstation, follow the *Remote NOAAPORT circuit configuration* procedure.

Family of Services communications port configuration

Use this procedure if your data source is the Family of Services.

1. Determine the `PORT=/dev/tty`*nn* values of the communications port on your workstation.

2. Edit the configuration files `DDS.CFG`, `PPS.CFG`, `IDS.CFG`, and `HRS.CFG` in the `~oper/mcidas/data` directory. In each file, change the `/dev/tty`*nn* value on the `PORT=` line to the value determined in step 1. The default values are listed below.

Path and file name	Installation defaults
`~oper/mcidas/data/DDS.CFG`	`PORT=/dev/ttyC0`
`~oper/mcidas/data/PPS.CFG`	`PORT=/dev/ttyC1`
`~oper/mcidas/data/IDS.CFG`	`PORT=/dev/ttyC2`
`~oper/mcidas/data/HRS.CFG`	`PORT=/dev/ttyC3`

3. Log on to the `root` account and add the following lines to the end of the file `/etc/rc.local` to prevent the owner privileges of your communications port from changing when you boot the workstation. The pound sign (#) represents the letter specific to the communications port on your workstation. For example, if your communication device names are the same as those listed in step 2, `tty#0` is set to `ttyC0` and `tty#?` is set to `ttyC?`.

```
if  [ -f /dev/tty#0 ]; then
    chown root /dev/tty#?
    chmod 776 /dev/tty#?
fi
```

4. Log out of the `root` account.

Local NOAAPORT circuit configuration

Use this procedure if your data source is the NWS NOAAPORT broadcast and your McIDAS-XCD software is on the same workstation as the SDI ingestor.

The NOAAPORT circuit configuration files should contain the correct values. In the configuration files **NTXT.CFG** and **NBIN.CFG** in the **~oper/mcidas/data** directory, verify that the **/tmp/jmb.fifo.#** value on the **FILE=** line is set to the values shown below.

Path and file name	Installation defaults	Circuit Type
~oper/mcidas/data/NTXT.CFG	/tmp/jmb.fifo.1	Text
~oper/mcidas/data/NBIN.CFG	/tmp/jmb.fifo.2	Binary

Remote NOAAPORT Circuit Configuration

Use this procedure if your data source is the NWS NOAAPORT broadcast and your McIDAS-XCD software is on a workstation other than the SDI ingestor workstation.

1. Determine the IP or name address of the host running the NOAAPORT SDI ingestor. SSEC recommends that the workstation running the NOAAPORT SDI ingestor and the workstation running -XCD reside on the same local network.

2. Edit the configuration files **NTXT.CFG** and **NBIN.CFG** in the **~oper/mcidas/data** directory. In each file, comment out the **FILE=** and **PERM=** lines by inserting a pound sign (#) at the beginning of the line.

3. Uncomment the **HOST=** and **HOST_PORT=** lines by removing the pound sign (#). These lines define the host and TCP port number used by the McIDAS-XCD ingestors. In each file, change the default value on the **HOST=** line to the IP or name address of the SDI ingestor. You should not have to modify the **HOST_PORT=** line. The default values are listed below.

Path and file name	HOST	HOST_PORT	Circuit Type
~oper/mcidas/data/NTXT.CFG	127.0.0.1	1501	Text
~oper/mcidas/data/NBIN.CFG	127.0.0.1	1502	Binary

Activating the GRIB decoder

If you will decode NCEP grids, activate the GRIB decoder and data monitor. If you are not decoding NCEP grids, skip this section. Run the two commands below from an **oper** McIDAS-X session.

Type: **DECINFO EDIT DMGRID GRIB ACTIVE CONFIG=GRIBDEC.CFG**

Type: **DECINFO SET DMGRID ACTIVE**

Starting the McIDAS-XCD package

Use the steps below to start the McIDAS-XCD software. For more information about the commands STARTXCD, QRTMDG, DELWXT, and **statdisp** which are used in this section, see Chapter 3, *McIDAS-XCD Administrative Commands*.

1. Start the McIDAS-XCD software from the McIDAS-X Text and Command Window. The STARTXCD command is only run when McIDAS-XCD is installed.

   Type: **STARTXCD**

   Never run more than one STARTXCD command at a time and do not include the command STARTXCD in your **STARTUP.SYS** file.

   The STARTXCD command runs continuously in your McIDAS-X session, starting and stopping data monitors and ingestors as needed. If a data monitor or ingestor stops, STARTXCD automatically restarts it. If you cancel STARTXCD, cancel the associated data monitors and ingestors. If you exit McIDAS-X, your decoders and ingestors will stop running.

   If you activate or deactivate a data monitor or ingestor, STARTXCD automatically starts or cancels it. If you activate or deactivate an individual decoder within a data monitor, you must deactivate and reactivate the data monitor for that decoder.

2. Enter these three commands in the McIDAS-X local time scheduler to delete old data files. The variable *nn* represents the number of days of data to keep online.

   Type: **SKE #Y 00:01:00 999999 24 "QRTMDG MD 1 70** *nn* **DEV=NNN**

   Type: **SKE #Y 00:01:00 999999 24 "DELWXT** *nn* **DEV=NNN**

   Type: **SKE #Y 00:01:00 999999 24 "QRTMDG GR 5001 5310 2 DEV=NNN**

   Command QRTMDG deletes old MD and grid files generated by the decoders; command DELWXT deletes old text files generated by the ingestors.

3. Display the McIDAS-XCD status window with the Unix command **`statdisp`**. The ampersand (**`&`**) runs **`statdisp`** in the background.

   To display the status window from the McIDAS-X Text and Command Window,

   Type: **OS "statdisp &**

   To display it from the Unix window,

   Type: **`statdisp &`**

   The McIDAS-XCD status window is displayed during your McIDAS-XCD session. It lists information about data processed by the ingestors and data monitors. It reads status information from the LW file **`~oper/mcidas/data/DECOSTAT.DAT`**.

   To modify the McIDAS-XCD status window, use different flags when starting it. For more information on the available flags and their defaults, see Chapter 3, *McIDAS-XCD Administrative Commands*.

Configuring the mcadde account

This section describes how to configure the McIDAS-XCD decoder workstation to serve -XCD data using McIDAS ADDE. McIDAS-XCD version 7.4 includes ADDE servers to provide users with data types decoded and stored in McIDAS-XCD.

1. Log on to the workstation as the user **`mcadde`**.

2. Determine the full Unix path of the **`~mcidas/data`** directory on the server. Use this path in step 4. From an xterm session,

   Type: **`echo ~mcidas/data`**

3. Start a McIDAS-X session.

   Type: **`mcidas`**

4. Create the McIDAS string MCDATA to contain the fully expanded path of the **`~mcidas/data`** directory.

   Type: **TE MCDATA "***/serverdirectory*

   For example: **TE MCDATA "/home/mcidas/data**

5. Run the batch file **`XCD.BAT`**.

   Type: **BATCH "XCD.BAT**

   `XCD.BAT` redirects several data files, saves them in the redirection table XCD, and initializes the **`GROUPS.DAT`** and **`COUNTRY.DAT`** files.

   The message "**`BATCH:DONE`**" must be displayed before you can continue.

6. Run the batch file **`XCDADDE.BAT`** to initialize the real-time ADDE datasets and complete the installation of the McIDAS-XCD software. A list of the datasets created is shown on the next page.

   Type: **BATCH "XCDADDE.BAT**

   When the message "**`BATCH:DONE`**" appears, the installation is complete.

The datasets created by the **XCDADDE.BAT** batch file are listed below.

ADDE dataset	Type	Files	Description
RTGRIDS/ALL	GRID	5001-5400	all model grids decoded by McIDAS-XCD
RTGRIDS/ETA	GRID	5011-5050	real-time ETA model grids
RTGRIDS/MRF	GRID	5101-5200	real-time MRF model grids
RTGRIDS/NGM	GRID	5051-5090	real-time NGM model grids
RTGRIDS/RUC	GRID	5200-5280	real-time RUC model grids; may not be available on all workstations
RTGRIDS/MISC	GRID	5001-5010	miscellaneous real-time grids
RTPTSRC/AIRCRAFT	POINT	61-70	real-time AIREP and PIREP data
RTPTSRC/FOUS14	POINT	41-50	real-time FOUS14 data
RTPTSRC/SFCHOURLY	POINT	1-10	real-time surface hourly data
RTPTSRC/SHIPBUOY	POINT	31-40	real-time ship and buoy surface reports
RTPTSRC/SYNOPTIC	POINT	51-60	real-time synoptic data
RTPTSRC/UPPERMAND	POINT	11-20	real-time mandatory level RAOB data
RTPTSRC/UPPERSIG	POINT	21-30	real-time significant level RAOB data
RTWXTEXT/ SFCHOURLY	TEXT (OBTX)	—	default dataset name used by the SFCRPT command.
RTWXTEXT/SYNOPTIC	TEXT (OBTX)	—	default dataset name used by the SYNRPT command
RTWXTEXT/TERMFCST	TEXT (OBTX)	—	default dataset name used by the TAFRPT command
RTWXTEXT/UPPERAIR	TEXT (OBTX)	—	default dataset name used by the RAOBRPT command

Removing the McIDAS-XCD software

Use the steps below to remove the McIDAS-XCD software package from your McIDAS-X workstation. Enter the commands exactly as shown. When you type a command, press **Enter** to run it.

1. Remove the client redirections by entering the command below from the McIDAS-X Text and Command Window.

   Type: **BATCH "RMXCDDEC.BAT**

2. Open a Unix window and log on to the workstation as user **oper**.

3. Change to the directory **~oper/mcidas**.

   Type: **cd ~oper/mcidas**

4. Run the shell script **xcd_init** to initialize the environmental variables needed to remove the McIDAS-XCD package. You must leave a space between the two periods (. .) when typing the command.

   Type: **. ./xcd_init**

5. Switch to user **mcidas**.

   Type: **su mcidas**

6. Remove the weather text and observation servers.

   Type: **./xcd_uninstall addeservers**

7. Exit from user **mcidas**.

   Type: **exit**

8. Run the shell script **xcd_uninstall** to remove the McIDAS-XCD package. From an **oper** xterm,

   Type: **./xcd_uninstall**

   This command removes the McIDAS-XCD files from the **~oper/mcidas/src**, **~oper/mcidas/data**, and **~oper/mcidas/bin** directories. It then removes the directory **~oper/mcidas/xcd7.4** and its contents. The only remaining files are **xcd.tar7.4.Z** and **xcd_install**. To completely remove the -XCD package, delete these files as well.

McIDAS-XCD Administrative Commands

This chapter contains command documentation for the system configuration of McIDAS-XCD, including administrative commands for file management and data availability. All of the operational McIDAS-XCD client commands for accessing conventional data have been removed from the McIDAS-XCD package. These commands have been replaced with ADDE (Abstract Data Distribution Environment) commands distributed with McIDAS-X. See the *McIDAS-X User's Guide* for more information.

Only authorized administrative staff should use these commands. To run them, you must be logged on as **oper**. If the error message "Permission Denied" is displayed, your logon does not correspond to the logon in the installation procedure. See Chapter 2, *McIDAS-XCD Software Installation*, for more information.

The administrative commands are listed in alphabetical order below with a short description of their function and page number.

BILDTEXT

Builds the rapid access pointer and text files for observational data.

Format

BILDTEXT ADD *id pfile*
BILDTEXT DEL *id pfile*
BILDTEXT INIT *pfile tfile maxsta maxreps idtype maxobs minhrs nbytes decnam idfile maxtxt [keyword]*
BILDTEXT LIST *pfile*

Parameters

ADD adds a station to an existing pointer file

DEL deletes a station from an existing pointer file

INIT initializes the *pfile* and deletes the existing *pfile* and *tfile*

LIST lists the configuration of a pointer file

id station ID to add or delete

pfile pointer file name (no default)

tfile text file name (no default)

maxsta maximum number of stations to store

maxreps maximum number of reports to store per observation time per station (default=1)

idtype
C4 4-character station ID
C8 8-character station ID
IDN station block number

maxobs maximum number of observation periods per station to store online (default=2)

minhrs minimum number of hours between observation blocks (default=1)

nbytes number of bytes necessary to store each line of an observation (default=80)

decnam decoder name for building the initial station ID list (no default)

idfile station ID file to use to build the initial station pointer list (default=**MASTERID.DAT**)

maxtxt maximum number of megabytes to store in a text file (default=32)

Keyword

CIR= list of defined circuits in *idfile* to build the initial station pointer list (default=all)

Remarks

BILDTEXT creates a pointer file and text file for observational data used by rapid access routines.

The INIT option is typically run only once per observation type to initialize the file structure. Running INIT deletes the existing versions of *pfile* and *tfile*. This command is run automatically for SAO, RAOB, SYN, and terminal forecasts when the McIDAS-XCD server software package is installed.

To list the valid circuit names from which to build your ID tables, type:

IDU LIST CIRCUIT

To list the valid decoder names from which to build your ID tables, type:

IDU LIST DECODER

When a station is added or deleted from a pointer file, the change does not take effect until the data monitor is restarted.

Examples

BILDTEXT INIT RAOB.RAP RAOB.RAT 1500 5 IDN 4 3 80 RAOB
This entry builds the pointer file RAOB.RAP which stores five reports for every 3-hourly observation for up to 1500 stations. Four observation periods are stored online for use with rapid access text applications. The raw text is stored in the file RAOB.RAT. The IDs are stored as station block numbers. The ID list built for the RAOB.RAP file is generated from the same ID list used by the RAOB decoder.

BILDTEXT ADD UES SAOMETAR.RAP
This entry adds the station UES to the pointer file SAOMETAR.RAP.

BILDTEXT INIT TERMFCST.RAP TERMFCST.RAT 2500 4 C4 6 1 80 TERMFCST X 8
This entry builds the pointer file TERMFCST.RAP which stores up to four reports per observation time and keeps up to six observation times available. The TERMFCST decoder builds the station list; the maximum size of the text file generated is eight megabytes. The raw text is stored in the file TERMFCST.RAT. The IDs are stored as character IDs.

CHKERR

Lists the output from an error file

Format

CHKERR *file day time [keyword]*

Parameters

file	file name (no default)
day	Julian day, YYDDD (no default)
time	time, HH (no default)

Keyword

NUM=	number of lines to output (default=20)

Remarks

CHKERR lists the errors generated by a data monitor. User-written data monitors must call the subroutine ERMESS to write a file readable by CHKERR.

You can use CHKERR to isolate system problems such as periodic aborts caused by corrupt pointer files.

When you install the McIDAS-XCD server software, error messaging is not active for decoders. To activate error messaging, edit the .CFG file appropriate for the decoder. SSEC recommends keeping the error messaging inactive unless there is a problem.

Examples

CHKERR DMSFC.ERR
This entry lists the last 20 lines written to the file DMSFC.ERR.

CHKERR DMSFC.ERR 93025 NUM=30
This entry lists the 30 lines preceding day 93025 in the file DMSFC.ERR.

CIRCUIT

Data circuit utility.

Formats

CIRCUIT ADD *circuit [keywords] "description*
CIRCUIT DEL *circuit*
CIRCUIT EDIT *circuit [keywords] "description*
CIRCUIT LIST *circuit*
CIRCUIT SET *circuit action*

Parameters

ADD	adds a circuit to the configuration file	
DEL	deletes a circuit from the configuration file	
EDIT	edits an existing circuit in the configuration file	
LIST	lists the specified circuit configuration (default=lists all circuits)	
SET	sets circuit processing to active or inactive	
circuit	circuit name; four characters maximum (no default)	
action	**ACTIVE**	activates a circuit
	INACTIVE	deactivates a circuit
"*description*	80-character circuit description	

Keywords

CONFIG=	circuit configuration file name
INGESTOR=	name of the ingestor to use; for example, **INGETEXT** or **INGEBIN**
SPOOL=	spool file name; used for the INGEBIN ingestors

Remarks

CIRCUIT is an operational utility that adds, deletes, edits, activates and deactivates circuits. All other configuration information about the circuit is entered in the circuit's configuration file using a text editor. See the example in Chapter 1.

If you change any parameters in the configuration file, you must inactivate the circuit for associated ingestors, wait for the circuit to stop, and then activate the circuit for the associated ingestors.

Examples

CIRCUIT LIST
This entry lists the circuit configurations for all circuits.

CIRCUIT ADD DDS INGESTOR=INGETEXT CONFIG=DDS.CFG "Domestic Data Service
This entry adds DDS to the list of circuits. Data from the circuit is processed when the circuit is activated. The configuration file name for this circuit is DDS.CFG.

CIRCUIT SET DDS ACTIVE
This entry activates the DDS circuit. The next time the STARTXCD program checks the circuit list, the DDS ingestion is started.

CIRCUIT EDIT DDS CONFIG=DDS01.CFG
This entry changes the name of the DDS circuit configuration file to DDS01.CFG.

DATACQ

Plots data availability from MD files.

Format

DATACQ *map mdf time [keywords]*

Parameters

map	map for the data plot; use any of the predefined maps used by the MAP command (default=WORLD)
mdf	MD file number (no default)
time	time of the data, HH (default=0)

Keywords

COL=	*rep miss*	reporting and missing data color levels (default=7 5)
ELE=	*min max*	TV element range for the data plot
GRA=	graphics frame number (default=current)	
LAT=	*min max*	latitude range to define the map
LIN=	*min max*	TV line range for the data plot
LON=	*min max*	longitude range to define the map
MDC=	*min max*	column range from the MD file to plot (default=all)
MDR=	row from the MD file to plot (default=row containing the time determined by the *time* parameter)	
SIZE=	size of the plot points, in pixels (default=2)	

Remarks

To plot the MD file data availability, the TIME, MOD, LAT and LON keys must be in the following locations in the MD file.

Key	Location
TIME	row header
MOD	data section
LAT	column header or data section
LON	column header or data section

To draw multiple plots in the same frame, use the McIDAS command PANEL to set up frame panels. Use the global keyword PAN to specify the frame panel where the plot is to be drawn.

Examples

DATACQ USA 4 12
This entry plots the data availability over the United States for 12 UTC from MD file 4.

DATACQ SAT 13 12
This entry plots the data availability for 12 UTC from MD file 13 over the currently displayed satellite image.

DATACQ X 33 X PAN=2
This entry plots the data availability over a world map for 0 UTC from MD file 33. The plot is drawn in panel two of a multipanel frame.

DATARECV

Plots acquired MD file data on a multiple-panel display.

Format

DATARECV *time [keywords]*

Parameter

time valid time (default=current hour)

Keywords

DAY= Julian day, YYDDD (default=current)

DEC= source decoder for the data: **ISFC**, **IRAB**, **ISHP**, **FO14**, **SYN**, **PIRP** (no default)

GRA= graphics frame number for the plot (default=current)

MAP= map for the data plot (default=world)

SIZE= height of the plotted characters, in pixels (default=2)

TIME= time for the plot (default=current)

Remarks

DATARECV is a macro that repeatedly calls the command DATACQ to plot acquired MD file data in a multiple-panel display.

The table below lists the default setting for each decoder.

Decoder	Default plotting time	MD files	Map
ISFC	Nearest hour observation	1-10	World
IRAB	Nearest 12-hour observation	11-20	World
ISHP	Nearest hour observation	31-40	World
FO14	Nearest 12-hour observation	41-50	USA
SYN	Nearest 6-hour observation	51-60	World
PIRP	Nearest hour observation	61-70	World

Examples

DATARECV DEC=ISFC SYN
This entry creates a two-panel global plot of surface hourly and synoptic data received for the current hour and synoptic time.

DATARECV DEC=ISFC IRAB ISHP SYN PIRP FO14 MAP=CA X X USA
This entry creates a six-panel display and plots the current ISFC data over California, and the SYN data over the United States. It uses the default maps to plot the current data for the IRAB, ISHP, PIRP and FO14 decoders.

DECINFO

Decoder utility.

Formats

DECINFO ADD *monitor [keywords]*
DECINFO DEL *type process [keywords]*
DECINFO EDIT *monitor decoder action [keywords] "description*
DECINFO LIST *monitor decoder*
DECINFO SET *monitor action*

Parameters

ADD	adds data monitors and decoders	
DEL	deletes data monitors and decoders	
EDIT	edits data monitors and decoders	
LIST	lists the current data monitor/decoder configurations	
SET	activates or deactivates data monitors	
monitor	data monitor name	
decoder	decoder name	
type	**DM**	deletes a data monitor
	DEC	deletes a decoder
process	data monitor or decoder to delete	
action	**ACTIVE**	activates data monitors and decoders
	INACTIVE	deactivates data monitors and decoders
"*description*	32-character description of the decoder	

Keywords

CONFIG=	configuration file name for the decoder
DEC=	decoders to add with the ADD option
DM=	data monitor from which the decoder is deleted; use with the DEL option
FORM=	**ALL** lists decoder configuration information

Remarks

DECINFO is an operational utility that adds, deletes, edits, lists, activates and deactivates data monitors and decoders.

If you add, delete, activate or deactivate a decoder, you must restart the decoder's data monitors for the action to take effect.

Examples

DECINFO LIST
This entry lists all the current data monitor/decoder configurations. If no configuration file exists, one is initialized as follows:

Data Monitor	Active	Decoder	Active
DMMISC	Yes	F14DEC	Yes
		WBXDEC	Yes
		PIRDEC	Yes
		TERDEC	Yes
		MDRDEC	Yes
DMRAOB	Yes	RABDEC	Yes
DMSFC	Yes	SAODEC	Yes
DMSYN	Yes	SYNDEC	Yes
		SHPDEC	Yes
DMGRID	No	GRIB	No

DECINFO ADD LOCAL DEC=FOUS67 TORNADO
This entry adds the data monitor LOCAL to the FOUS67 and TORNADO decoders.

DECINFO EDIT LOCAL TORNADO ACTIVE CONFIG=TORN.CFG "Tornado Warning Decoder
This entry activates the TORNADO decoder running under the data monitor LOCAL and attaches the label Tornado Warning Decoder. The configuration information for this decoder is in the TORN.CFG file.

DECINFO EDIT LOCAL FOUS67 ACTIVE "FOUS67 Decoder
This entry activates the FOUS67 decoder running under the data monitor LOCAL and attaches the label FOUS67 Decoder.

DECINFO EDIT LOCAL FOUS67 CONFIG=FOUS67.CFG
This entry changes the name of the configuration file for the FOUS67 decoder to FOUS67.CFG.

DECINFO SET LOCAL ACTIVE
This entry activates the data monitor LOCAL. The next time the STARTXCD program checks the data monitor, LOCAL is started.

DECINFO DEL DEC SHPDEC DM=DMSYN
This entry deletes the decoder SHPDEC from the data monitor DMSYN.

DELWXT

Deletes weather text and index files.

Format

DELWXT *days [keyword] "path*

Parameters

days	number of days before today to save text data, maximum of 10 (default=1)
"path	path name to search for data or index files to delete

Keyword

DAY=	deletes the specified day's files, YYDDD (no default)

Remarks

DELWXT deletes weather text and index files for a specified number of days. It should run from the system time scheduler once per day. Scheduling DELWXT to run daily frees up a considerable amount of file space by deleting old weather text and index files. At SSEC, DELWXT runs at 00:05 UTC and deletes files older than three days.

To delete a specific day's data, use the keyword DAY.

Examples

SKE 93003 00:05 999999 24 "DELWXT 3
This entry schedules DELWXT to run every 24 hours at 00:05 UTC from the system time scheduler. DELWXT saves weather text and index files containing data for the current day plus the three previous days. For more information about command SKE, see the *McIDAS-X User's Guide*.

DELWXT DAY=95017
This entry deletes the text and index files for 17 January 1995.

IDGROUP

ID group utility.

Formats

IDGROUP ADD *name [keywords]*
IDGROUP COMP *name [keywords]*
IDGROUP DEL *name [keywords]*
IDGROUP LIST *name [keywords]*
IDGROUP SAVE *name [keywords]*

Parameters

ADD	adds stations to an existing group or creates a new group
COMP	compresses the file after many groups are altered; deletes groups not marked as permanent
DEL	deletes stations from an existing group or deletes an entire group
LIST	lists stations in a group or all defined groups
SAVE	sets the save flag for a group or country
name	name of the group to update; 12 characters maximum (no default)

Keywords

DEC=		decoder types; use with the ADD option
GROUP=		group to add stations to or delete stations from; use with the ADD option
ID=		stations to add to or delete from a group
LAT=	*min max*	latitude boundaries of a group
LON=	*min max*	longitude boundaries of a group
SAVE=	**P**	creates a permanent group
	T	creates a temporary group
TYPE=	**COUNTRY**	performs an operation on a country
	GROUP	performs an operation on a group (default)

Remarks

IDGROUP is a utility for creating and editing groups of stations used with rapid text accessing applications.

You can only delete stations from a group you created.

The LAT and LON keywords are only valid when creating ID groups for the first time.

If the ID or GROUP keyword is not specified with the ADD option, IDGROUP scans the entire ID file to find stations matching either the state or country header. Valid decoder names for the DEC keyword are SAOMETAR, RAOB, FOUS14, SYNOPTIC and TERMFCST.

You can add up to 20 stations at a time using the ADD option with the ID keyword.

IDGROUP appends a zero to all WMO station block numbers with five digits.

Examples

IDGROUP ADD NH
This entry creates the group NH which contains all the stations in New Hampshire.

IDGROUP ADD GB TYPE=COUNTRY
This entry creates the country GB which contains all stations in Gambia.

IDGROUP ADD FRED ID=MSN MEM MSP 72645 72532 SEA SAC SFO
This entry creates a group named FRED containing eight stations.

IDGROUP ADD FRED ID=EGLL UUEE 26216 GROUP=NH
This entry adds stations EGLL, UUEE, 26216 and all the stations in group NH to the group FRED.

IDGROUP LIST
This entry lists all defined groups.

IDGROUP LIST FRED
This entry lists all the stations in the group FRED.

IDGROUP DEL FRED ID=UUEE 72645
This entry deletes Moscow and Green Bay from the group FRED.

IDGROUP ADD MIDWESTSFC LAT=35 50 LON=85 100 DEC=SAOMETAR SYNOPTIC
This entry creates the group named MIDWESTSFC which contains all stations between 35° and 50° N and 85° and 100° W reporting either surface hourly data or synoptic reports.

IDMON

Station ID monitoring utility.

Formats

IDMON COMP *file idfile type*
IDMON HIST *file days station*
IDMON LIST *file*

Parameters

COMP	compares the active reporting stations with the stations being decoded
HIST	lists the history of a station
LIST	lists the station data from the old or new station file
file	file name containing the list of old or new stations
idfile	file containing the master table of the station in the MD file
type	data type: **FOUS**, **ISFC**, **IRAB** or **SYN**
days	number of days before today to summarize the station reporting status (default=file creation date)
station	station ID or WMO header

Remarks

IDMON monitors the status of stations and generates lists to inform operations of new stations and stations that stopped reporting. With this information, operations can remove or add data to the station dictionary using command IDU.

To activate or deactivate station ID monitoring, edit the appropriate decoder configuration file. Decoders developed at SSEC create two station files: OLD*type*.IDM and NEW*type*.IDM where *type* is one of the following data types.

Data type	Decoder
ISFC	SAO/METAR
IRAB	upper air
SYN	synoptic
FO14	FOUS14
PIRP	pilot report
TERM	terminal forecast

File OLD*type*.IDM contains a record of all stations that are reporting data and are included in the master table of stations in the decoder's MD file.

File NEW*type*.IDM contains all stations reporting data, but are not included in the master table of stations in the decoder's MD file.

To add new stations reporting data or delete stations no longer reporting, use command UPDIDS. SSEC recommends updating station files monthly for locally developed decoders. If you update your own master ID table, please document the changes and notify SSEC.

SSEC sends updated ID files with each McIDAS upgrade.

Examples

IDMON HIST OLDISFC.IDM 50
This entry lists the stations filed in OLDISFC.IDM that have reported since station monitoring was activated, but have not reported in 50 days or more.

IDMON HIST OLDISFC.IDM X MSN
This entry lists the last date and time data was reported from Madison, Wisconsin, and filed in OLDISFC.IDM.

IDMON HIST NEWIRAB.IDM
This entry lists the new RAOB stations that are reporting data but are not currently being filed in the MD file.

IDMON COMP OLDSYN.IDM SYNDEC.IDT
This entry lists the old synoptic stations that exist in the column headers of the MD file but have never reported.

IDMON LIST NEWFO14.IDM
This entry lists the station location of new stations that did not previously report for the FOUS14 decoder.

IDU

Station dictionary utility.

Formats

IDU ADD CIRCUIT *circuit*
IDU ADD DECODER *decoder*
IDU ADD *id idn* **CO= LAT= LON= ELE=** "*description*
IDU DEL CIRCUIT *circuit*
IDU DEL DECODER *decoder*
IDU DEL *station*
IDU EDIT *station [keywords]* "*description*
IDU LIST CIRCUIT
IDU LIST DECODER
IDU LIST *station*
IDU LIST "*description*

Parameters

ADD	adds data to the station dictionary
DEL	deletes data from the station dictionary
EDIT	edits the station dictionary
LIST	lists data from the station dictionary
CIRCUIT	performs a function on a circuit
DECODER	performs a function on a decoder
circuit	circuit name for the station dictionary; four characters maximum
decoder	decoder name for the station dictionary; eight characters maximum
id	station identification characters
idn	6-digit station number which includes the WMO number plus air weather service number
station	station ID or WMO header; eight characters maximum
"*description*	24-character station name and location

Keywords

CIR=	*c1 . . cn*	circuit names to activate for a decoder (default=all)
CO=	2-character country code	
DEC=	*d1 . . dn*	decoders to add to or delete from a station list
ELE=	*pri sec*	primary and secondary station elevations meters
FILE=	station dictionary file name (default=**MASTERID.DAT**)	
ID=	*name*	new station name
	BLANK	removes the character ID
IDN=	new station number	
LAT=	*pri sec*	primary and secondary station latitudes
LON=	*pri sec*	primary and secondary station longitudes
ST=	2-character state code	
SWI=	**YES**	activates a decoder or circuit for a station (default)
	NO	inactivates a decoder or circuit for a station

Remarks

IDU generates a local station dictionary in the file MASTERID.DAT. Decoders read this file to determine if a station's data should be processed. Decoders also use the list to generate initial MD file headers.

You can add, delete, or change stations. When adding a station, specify a 6-digit station number, latitude, longitude and station elevation. The 6-digit number is derived from the WMO 5-digit station number according to Air Weather Service Pamphlet 105-52.

You can change the decoder and circuit names that process a station's data, state and country codes, elevation, latitude and longitude, or a station's description, name or number. You can also enter secondary latitudes, longitudes and elevations for any station.

The station dictionary can store a maximum of 51,200 stations, 32 circuits and 128 decoders.

Use command IDMON to identify old or new reporting stations; edit the appropriate decoder configuration file to activate monitoring.

Deleting a decoder removes it from all stations. Adding a decoder to the list of valid decoders does not activate any station for the decoder. Use the EDIT option to manually activate each station.

Examples

IDU ADD MSN 726410 LAT=43 LON=89:20 ELE=262
"Madison Truax
This entry adds the station MSN to the station dictionary. The WMO station block number is 726410. MSN is located at 43° N and 89:20° W with an elevation of 262 meters.

IDU ADD EGLL 037720 LAT=51.5 LON=0 ELE=24 CO=UK
"London Heathrow
This entry adds the station EGLL to the station dictionary. Note the WMO station block number is entered as six digits.

IDU EDIT MSN ST=WI CO=US LON=X 89:21 LAT=X 41.8
This entry edits the station data for Madison, adding the state and country codes and secondary latitude and longitude values.

IDU ADD CIRCUIT DDS
This entry adds the circuit DDS to the station dictionary.

IDU ADD DECODER FOUS14
This entry adds the decoder FOUS14 to the station dictionary.

IDU LIST CIRCUIT
This entry lists the currently defined circuits.

IDU LIST DECODER
This entry lists the currently defined decoders.

IDU EDIT MSN DEC=FOUS14
This entry adds Madison to the FOUS14 decoder station list for all circuits.

IDU EDIT DAB DEC=SAOMETAR CIR=DDS PPS SWI=NO
This entry deletes Daytona Beach from the station list for the SAOMETAR decoder and DDS and PPS circuits.

IDU LIST 726450
This entry lists all information in the station dictionary for station 726450.

IDU LIST LAS
This entry lists all information in the station dictionary for Las Vegas.

IDU EDIT 726450 ID=DAVE
This entry changes the character station name for station 726450, Green Bay, to DAVE.

NMCAMT

Lists the number of real-time grids received for a specified day.

Format

NMCAMT [*keywords*]

Keywords

DAY= Julian day, YYDDD (default=current)

FORM= **STD** lists a brief description of the grids; see the Remarks (default)

ALL lists detailed information about the grids; see the Remarks

GRIDF= grid file numbers to check (default=lists all real-time grid files for the specified day)

MOD= **ALL** searches all model types (default)

m1 . . mn searches a range of models, valid options:

ETA lists grids in the ETA model projection

NGM lists nested grid models

MRF lists Medium Range Forecast grids

TIME= **ALL** searches all run times (default)

t1 . . tn searches the specified range of run times, H, HH:MM or HH:MM:SS

Remarks

NMCAMT lists both complete and partial real-time grids. A partial grid contains incomplete grid sections. Currently, the Medium Range Forecast (MRF) model is the only model transmitted in pieces; thus, it is the only model containing partial grids.

The FORM=STD option provides the following information:

- grid file number
- grid number
- parameter
- level
- run time
- forecast time
- model name of partial grids

The screen below shows an example of the FORM=STD output.

```
McIDAS0: rickk@outfield [T=1]
 GridF Grid Parm Level  RT VT Mod
  5109  257 V    1000   0   0 MRF
  5109  287 T    850    0   6 MRF
  5119   67 RH   SFC    0  30 MRF
  5129    2 T    700    0  60 MRF
  5129   11 T    MAXW   0  60 MRF
total number of grids= 832
total number of partial grids= 5
NMCAMT: done
```

The projection of an MRF grid determines how it is sent. Grids that are 145 rows by 289 columns are sent in the following eight sectors:

Sector	Latitude range	Longitude range
1	0°N-90°N	60°E-30W
2	0°N-90°N	150°E-60°E
3	0°N-90°N	120°W-150°E
4	0°N-90°N	30°W-120°W
5	0°S-90°S	60°E-30°W
6	0°S-90°S	150°E-60°E
7	0°S-90°S	120°W-150°E
8	0°S-90°S	30°W-120°W

Grids that are 73 rows by 73 columns are sent in the following four sectors:

Sector	Latitude range	Longitude range
1	0°N-90°N	180°E-0°E
2	0°N-90°N	0°W-180°W
3	0°S-90°S	180°E-0°E
4	0°S-90°S	0°W-180°W

If a grid is missing a section, the FORM=ALL option marks the missing section with an asterisk (*). In the example below, grid 257 in grid file 5109 is missing the fourth section. The Xs indicate that sectors 1-3 and 5-8 were received.

```
McIDAS0: rickk@outfield [T=1]
 GridF Grid Parm Level  RT VT Mod  location  nr  nc received  sectors
  5109  257 V    1000   0   0 MRF     21407 145 289  070733  |xxx*xxxx|
  5109  287 T    850    0   6 MRF   1331842 145 289  071923  |xxxxx*xx|
  5119   67 RH   SFC    0  30 MRF  28971257 145 289  062425  |xx*xxxxx|
  5129    2 T    700    0  60 MRF  11252157 145 289  084818  |xx*xxxxx|
  5129   11 T    MAXW   0  60 MRF  11588316 145 289  085109  |xxxxxxx*|
total number of grids= 832
total number of partial grids= 5
NMCAMT: done
```

The FORM=ALL option provides the following additional information:

- byte location of the message in the spool file
- the number of rows and columns of the grid
- the time the grid was received

Examples

NMCAMT
This entry lists all the real-time grids that are missing grid sections and the total number of grids received for the current day.

NMCAMT MOD=MRF FORM=ALL TIME=0
This entry lists, in expanded form, all the 0 UTC model run MRF grids.

QRTMDG

Deletes real-time grid or MD files.

Formats

QRTMDG GRID *bfile efile numdays*
QRTMDG MD *bfile efile numdays*

Parameters

GRID	deletes grid files
MD	deletes MD files
bfile	beginning file in the range to delete; must end with the number 1
efile	ending file in the range to delete; must end with a zero (default=*bfile* + 9)
numdays	number of days of real-time data to store online; the range is 2 to 9 (default=4)

Remarks

SSEC recommends scheduling QRTMDG to run at 00:00:01 UTC daily to delete old grid and MD files.

During the first 10 days of each new year, QRTMDG deletes the appropriate files for sites keeping more than five days of data online to ensure that no collisions occur. For example, if a site keeps six days of data online from day 91003, the valid days are 91003, 91002, 91001, 90365, 90364 and 90363. However, a file collision occurs between days 91003 and 90363. To prevent this collision, QRTMDG deletes data from day 90363.

Example

SKE 93129 00:01:00 999999 24 "QRTMDG MD 1 40 4
This example stores the current day's MD data plus the previous three days for MD files in the range 1 to 40. When this command runs on day 93307, all MD files in the range 1 to 40 are deleted except: 4, 5, 6, 7, 14, 15, 16, 17, 24, 25, 26, 27, 34, 35, 36 and 37.

REMRF

Regrids MRF data from the high resolution format to a lower resolution format and sends grids to the mainframe.

Format

REMRF *[keywords]*

Keywords

DAY= day to acquire real-time data (default=current)

DES= *gridf run time fcst*

gridf first destination grid file on the mainframe (no default)

run interval between model runs (default=2 hours)

time valid time interval for storing grids (default=24 hours)

fcst maximum valid forecast time to contain unique storage grid (default=96 hours)

LOW= **NO** do not include low resolution source grids (default)

YES include low resolution source grids

MAX= maximum number of grids in the destination grid file (default=1000)

RUN= model run time for acquiring real-time data (default=most recent 12-hour period)

SCR= scratch grid files to use for build before sending grids to the mainframe (default=99991 - 99995)

SMO= **AVERAGE** smooth and reduce grid resolution by averaging

SAMPLE smooth and reduce grid resolution by sampling (default)

Remarks

Run this command from the local scheduler every 15 minutes while the workstation is logged on to the mainframe as the user **oper**. Do not run this command from the command line.

High resolution grids are 145-row by 289-column global Mercator grids. These grids have 1.25° latitude and 1.25° longitude resolution between data points.

Low resolution grids are 73-row by 73-column global Mercator grids. These grids have 2.5° latitude and 5.0° longitude resolution between data points.

If you run REMRF, you should not run command SENNMC.

This program calls the command REMRF1 which calculates the regridded fields.

Example

REMRF DES=5001
This entry reformats high resolution grids into low resolution grids and sends them to the following mainframe grid files.

```
  5001-5010    00Z MRF    00hr    <=    valid time    <=    24hr forecast
  5011-5020    00Z MRF    24hr    <     valid time    <=    48hr forecast
  5021-5030    00Z MRF    48hr    <     valid time    <=    72hr forecast
  5031-5040    00Z MRF    72hr    <     valid time    <=    96hr forecast
  5041-5050     00Z MRF     >96hr forecast
  5051-5060    12Z MRF    00hr    <=    valid time    <=    24hr forecast
  5061-5070    12Z MRF    24hr    <     valid time    <=    48hr forecast
  5071-5080    12Z MRF    48hr    <     valid time    <=    72hr forecast
  5081-5090    12Z MRF    72hr    <     valid time    <=    96hr forecast
  5091-5100     12Z MRF     >96hr forecast
```

REMRF1

Reformats MRF grids from high resolution to low resolution.

Format

REMRF1 *sgridf bgrid egrid dgridf [keywords]*

Parameters

sgridf source grid file to reformat (no default)

bgrid beginning grid number to reformat (default=1)

egrid ending grid number to reformat (default=all)

dgridf destination grid file (no default)

Keywords

AUD= **NO** do not use auditing

YES use auditing to track which grids are reformatted for a run time (default)

HIS= history audit file name (default=**MRFSENT**)

LOW= **NO** do not include low resolution source grids (default)

YES include low resolution source grids

MAX= maximum number of grids in the destination grid file (default=1000)

SMO= **SAMPLE** smooth and reduce grid resolution by sampling (default)

AVERAGE smooth and reduce grid resolution by averaging

Remarks

Do not run this command; it is called by the REMRF command.

If the REMRF1 command encounters a grid with the model name AVN, it converts it to MRF. This ensures that the destination grid is sent to the mainframe consistently.

Use keyword AUD to track the last grid checked in each grid file. Each time REMRF1 runs, it only checks to see if new grids are needed and does not retransmit previously sent grids. REMRF1 stores the audit information in the file MRFSENT. Use keyword HIS to specify a different history audit file name.

SENNMC

Sends real-time grids from McIDAS-XCD to the mainframe.

Format

SENNMC *dgridf [keywords]*

Parameter

dgridf first destination grid file number on the mainframe

Keywords

DAY= Julian day of the data to send (default=current)

ETA= **NO** do not send grids in the ETA model projection, i.e., tangent cone Lambert conformal (default)

YES send only the ETA model in this projection

ALL send all grids, regardless of the model, received in the ETA projection

LOG= name of the file that logs the last grid sent (default=**GRIDSENT**)

MAX= maximum number of grids to store in the mainframe grid file (default=2000)

MOD= list of models to send (default=all)

RUN= run time for acquiring data (default=most recent 12-hour period)

SCR= scratch grid file used to send data (default=99990)

SMO= **SAMPLE** smooth and reduce grid resolution by sampling (default)

AVERAGE smooth and reduce grid resolution by averaging

THIN= **NO** do not reformat high resolution grids to low resolution before sending to mainframe

YES reformat high resolution grids to low resolution and send to the mainframe (default)

Remarks

The value for *dgridf* should be the same value stored in SYSKEY table word 3100 on the mainframe.

Run this command from the local scheduler every 15 minutes while the workstation is logged on to the mainframe as the user **oper**. Do not run this command from the command line.

High resolution grids are 145-row by 289-column global Mercator grids. These grids have 1.25° latitude and 1.25° longitude resolution between data points.

Low resolution grids are 73-row by 73-column global Mercator grids. These grids have 2.5° degree latitude and 5.0° longitude resolution between data points.

If you run SENNMC, you should not run the command REMRF.

Example

SENNMC 16000
This entry sends 00 UTC model run grids to grid files 16001 through 16010 and 12 UTC model run grids to grid files 16011 through 16020. High resolution MRF grids are reformatted into low resolution grids. No ETA model data is sent to the mainframe.

SIGCO

Significant level upper air storage utility.

Formats

SIGCO ADD *country*
SIGCO DEL *country*
SIGCO LIST *country*

Parameters

ADD	adds a country to the list
DEL	deletes a country from the list
LIST	lists countries for which significant level upper air data is saved (default)
country	2-character country code

Remarks

SIGCO specifies the countries for which significant level upper air data is decoded and filed. The list of countries is stored in the file SIGCO.DAT. Changes to the list are implemented when the upper air decoder is restarted. Use the McIDAS-X command CCODE to obtain a list of valid two-letter country codes. See the *McIDAS-X User's Guide* for more information.

Examples

SIGCO ADD VN
This entry adds Venezuela to the list of countries for which significant level data is saved.

SIGCO DEL MX
This entry deletes Mexico from the list of countries for which significant level data is saved.

STARTXCD

Starts the ingestor and decoder programs.

Format

STARTXCD *dtime*

Parameter

dtime number of seconds to pause between programs (default=120)

Remarks

STARTXCD is the parent program that automatically starts and stops the McIDAS-XCD ingestors and decoders.

Never have more than one STARTXCD command running at a time.

STAT

Lists McIDAS-XCD decoder and ingestor status.

Format

STAT *[keyword]*

Keyword

TOL= warning tolerance in minutes; if an ingestor or decoder does not process data within the tolerance, an asterisk (*) appears next to the time stamp

Remarks

STAT provides a snapshot of the bulletin board status display.

SUBGRD

Creates geographic subsectors of Mercator grids.

Format

SUBGRD *sgridf bgrid egrid dgridf [keywords]*

Parameters

sgridf	source grid file
bgrid	beginning grid number to subsect (default=1)
egrid	ending grid number to subsect (default=all)
dgridf	destination grid file number

Keywords

LAT=	*slat nlat*	destination latitude extents (no default)
	slat	southern latitude extent
	nlat	northern latitude extent
LON=	*elon wlon*	destination longitude extents (no default)
	elon	eastern longitude extent
	wlon	western longitude extent
MAX=		maximum number of grids in destination grid file (default=*egrid-bgrid*+1)

Remarks

The command SUBGRD only creates geographic subsectors of Mercator projection grids.

Example

SUBGRD 1000 1 10 1200 LAT=20 60 LON=40 150
This entry creates grid subsectors with the geographic domain 20° to 60° North and 40° to 150° West from grids 1 through 10 in grid file 1000. The grid subsectors are stored in grid file 1200, which stores 10 grids maximum.

UPDIDS

Updates the station reporting list for decoders.

Formats

UPDIDS ACT *decoder source minnum type [keywords]*
UPDIDS INACT *decoder source cutday type idtable [keywords]*

Parameters

ACT		activates decoding for a station list
INACT		deactivates decoding for a station list
decoder		decoder name
source		source ID file written by IDNEW
minnum		minimum number of station references needed to activate the decoder
cutday		inactivates stations that have not reported in the past number of *cutdays* (default=100 days)
type	**CID**	character ID (default)
	IDN	station block number
idtable		current ID table used for comparison

Keywords

CIR=	activates stations for specified circuits (default=**ALL**)
FILE=	file name to update (default=**MASTERID.DAT**)

Remarks

UPDIDS activates or deactivates decoding for specified stations. Stations must exist in the station dictionary to successfully activate a decoder for a station. Use command IDU to add stations to the station dictionary. The *source* station list used with UPDIDS is generated in the decoders. These file names are stored in the .CFG file associated with each decoder.

Examples

UPDIDS ACT SAOMETAR NEWISFC.IDM 10 CID
This entry activates stations that have reported 10 or more times in the file NEWISFC.IDM for the decoder SAOMETAR.

UPDIDS INACT RAOB OLDIRAB.IDM 50 IDN IRABDEC.IDT
This entry deactivates stations for the RAOB decoder that have not reported in the last 50 days.

WMORTE

Maintains a data routing table of WMO headers.

Formats

WMORTE ADD CIR *circuit*
WMORTE ADD INDEX *index* **ALIAS=**
WMORTE ADD WMO *[keywords]*
WMORTE DEL CIR *circuit*
WMORTE DEL INDEX *index* **ALIAS=**
WMORTE DEL WMO *[keywords]*
WMORTE EDIT INDEX *index* **ALIAS=**
WMORTE EDIT WMO *[keywords]*
WMORTE LIST *circuit*

Parameters

ADD	adds a circuit, index, or WMO header
DEL	deletes a circuit, index, or WMO header
EDIT	edits an index or WMO header
LIST	lists the WMO headers, indices and aliases for a circuit
CIR	circuit
INDEX	index
WMO	WMO header specified with keyword HEADER
circuit	circuit name
index	2-character index name

Keywords

ALIAS=	alias file name used with the *index* options; eight characters maximum (default=*index*)
CIRCUIT=	circuit name for adding, editing, and deleting the WMO specification
HEADER=	WMO headers to add, edit or delete; two characters maximum
INDEX=	index for adding or editing WMO headers; two characters maximum; you can specify more than one index when adding multiple WMO headers (default=WMO header specified)

Remarks

WMORTE creates a routing table describing the location of text data. It is created by assigning one or more WMO headers, ingested from each circuit, to an index name. The file that stores the text data is defined by the index name. For example, if the index specified is CS and the date of the data contained in the file is 93002, the file name for the index file is CS93002.IDX.

The WMO headers and index names can only be two characters. If data is ingested with a WMO header that is not on the list of defined headers, the data is filed in the miscellaneous index file ZZ.

Each index file can have an associated alias name. The alias name or index name can then be used by the ADDE weather text server. For example, since data ingested with the WMO header CS is climatological information, you could assign an alias name CLIMATE to the index CS. A user could then specify either CS or CLIMATE when using the WXTLIST command.

You can define a maximum of 32 circuits, 512 indices and 1024 WMO headers using this routing system.

If you alter the contents of a circuit's routing table, other than the alias name, you must restart the circuit to activate the new or updated routing table.

The first time WMORTE is run, it initializes the file IDXALIAS.DAT to a predefined routing table for the DDS, IDS, PPS and Carswell circuits. This must be done before starting the circuits with STARTXCD.

Each WMO header is stored as either a primary or secondary index. Use the command **WMORTE LIST** *circuit* (where *circuit* is a valid circuit name, for example, DDS) to list your system's indexing. Primary indices are listed under the INDEX column; secondary indices are listed under the WMO HEADERS column.

Examples

WMORTE LIST DDS
This entry lists the aliases, indices, and WMO headers for the DDS circuit.

WMORTE ADD CIR CDS
This entry adds the circuit CDS to the defined circuit names.

WMORTE ADD INDEX RW ALIAS=RIVER
This entry adds the index name RW to the list of defined indices and gives it the alias name RIVER.

WMORTE ADD WMO HEADER=RR RW CIRCUIT=CDS INDEX=RW RW
This entry adds the WMO headers RR and RW, which are filed in the index RW to the CDS circuit.

WMORTE EDIT INDEX SA ALIAS=SURFACE
This entry changes the alias name of the index SA to SURFACE.

WMORTE DEL CIR CARS
This entry deletes the circuit CARS from the list of defined circuit names.

WMORTE DEL WMO HEADER=WF WU CIRCUIT=IDS DDS
This entry deletes the WMO headers WF and WU from the IDS and DDS circuits.

WMORTE DEL INDEX SM
This entry deletes the index SM from the routing table.

statdisp

Unix command that starts the McIDAS-XCD status display.

Format

`statdisp` *[flags]*

Flags

`-bg` *color*	background color (default=black)
`-display` *display*	workstation name and window manager to use for display
`-fg` *color*	foreground color (default=white)
`-font` *font*	font size to use (default=6 x 12)
`-geometry` *+xoffset+yoffset*	position of status window
`-resize`	stops automatic window resizing
`-sample` *seconds*	screen refresh sampling time (default=5 seconds)
`-threshold` *minutes*	warning threshold time (default=5 minutes)
`-warn` *color*	warning color (default=red)

Remarks

This command starts an X window to display the status of McIDAS-XCD decoders and ingestors. Active decoders and ingestors are displayed in the foreground color. Decoders and ingestors that are inactive longer than the warning threshold time are displayed in the warning color.

To cancel the status display, click on Quit the Window in the Title bar. To display a window in a smaller size than is necessary to view the entire bulletin board, use the `-resize` option.

To force `statdisp` to get data from a file other than `~oper/mcidas/data/DECOSTAT.DAT`, set the environmental variable `XCD_disp_file` to the fully expanded file name.

To start `statdisp` from the McIDAS-X command window, precede it with **OS "**. Run it in the background using the **&** (ampersand) shell option.

Use the `-geometry` option to specify the screen position of the status window. Specify the horizontal and vertical offsets in pixels. Offsets must be preceded by plus signs (+).

Example

```
statdisp -bg white -fg black -warn magenta &
```

This entry starts the status display with a white background, black foreground, and magenta warning messages from the Unix command window. The `&` (ampersand) shell option runs the command in the background.

Troubleshooting

This chapter lists problems that may occur with McIDAS-XCD and the -XCD data relay. Under each symptom or error message, possible solutions are given.

McIDAS-XCD problems

-XCD is not receiving real-time data

The user reports no real-time data, or the ingestor status display is red.

The file system may be full. You can use the McIDAS-XCD commands QRTMDG and DELWXT to delete older text, point files and grid files. Do not delete any files for the current day.

Check for an obstruction in the antenna and verify that all receiving hardware is working properly.

Contact your source provider to see if they are having a problem with the broadcast.

-XCD is not receiving grid data

The GRIB decoder can't file grids if it can't find `RTMODELS.CFG`, which contains information about real-time grid file locations.

The file `RTMODELS.CFG` should reside in `~mcidas/data` when McIDAS-XCD is installed correctly. Either the decoder can't reach the file or it is missing. If it's missing, recreate the file or copy a new version of the default file from `~mcidas/xcd1.1/data/RTMODELS.CFG`.

Data is garbled or missing

When more than one ingestor is trying to read the same circuit, text data may be missing or text output garbled. If you are decoding grids, grids may be missing.

Only one ingetext process should be running for each text circuit, and only one ingebin process should be running for each binary circuit. Check the number of ingetext and ingebin processes running.

1. Find the process IDs of all -XCD processes.

   Type: **`ps -u | grep oper`**

2. Stop the -XCD processes in the following order:
   - **`STARTXCD`**
   - **`INGETEXT`**
   - **`INGEBIN`**
   - **`DM*`**

   Type: **`kill -9`** *processid*

3. Restart -XCD with the McIDAS command STARTXCD.

If this process doesn't work, check for an obstruction in the receiving antenna.

McIDAS-XCD data relay problems

The mainframe tries to connect but xcdrelay on the -XCD workstation does not run

You must use the fully qualified file names when inetd is running. Change the file **`/etc/inetd.conf`** to include the fully qualified command and configuration file names; for example,

```
/home/oper/mcidas/bin/xcdrelay xcdrelay
/home/oper/mcidas/data/fosrelay.cfg
```

Also, read the error file **`/tmp/xcdrelay`***xxxxxx* on the -XCD workstation. *xxxxxx* is a time stamp of when the error occurred.

The TCP/IP link is established but no data is received on the mainframe

The configuration file, **`~oper/mcidas/data/fosrelay.cfg`** may not point to the correct directory. Check that the paths specified by the **`FOS_PATH=`** and **`BINARY=`** lines are the location of the data files. Check the redirection table in the **`oper`** account to verify that the appropriate -XCD files are in the directories specified in the configuration file.

Read the error file **`/tmp/xcdrelay`***xxxxxx* on the -XCD workstation. *xxxxxx* is a time stamp of when the error occurred. The output in the file may provide an indication of an incorrect call to **`xcdrelay`** in the file **`/etc/inetd.cfg`**.

The mainframe will not connect to the port on the Unix workstation

There is no physical connection between the mainframe and the -XCD workstation. To determine the connection status, type the command below from a Unix command window on the -XCD workstation.

Type: **`ping`** *ipaddress*

Verify that you put the leading zeros in the IP address for the workstation in the file MCIDAS.PARMLIB(XCDCORE1).

Verify that the port number in MCIDAS.PARMLIB (XCDCORE1) corresponds to the **`/etc/services`** file on the -XCD workstation.

Verify the **`/etc/inetd.conf`** file is configured correctly. This file tells the system how to start and connect. See Chapter 6, *Configuring the McIDAS-XCD Data Relay*.

No connection is established when starting XCDCORE1 from the console
The ASYNCs are not running. You must have all the ASYNCs running that the relay expects before you start XCDCORE1.

"EM3708 HAS LOST CONTACT" message on the mainframe master console
This message comes from the mainframe relay task XCDCORE1. The mainframe has lost the TCP connection to the relay system on the -XCD workstation. Restart the McIDAS-XCD workstation and server software.

Decoding GRIB Messages

The McIDAS-XCD GRIB decoder converts the binary data stream of the High Resolution Data Service (HRS) sent by the National Weather Service (NWS) into McIDAS grid files. This section describes the steps required to ingest and decode the data stream into McIDAS grids.

- processing the GRIB message
- converting GRIB codes
- filing the grid in McIDAS

Processing the GRIB message

When DMGRID data monitor successfully reads a complete message, it calls the McIDAS-XCD GRIB decoder. The decoder first decodes the Product Definition Section (PDS) to determine the type of data contained in the message. After processing this section of the message, the decoder has enough information to determine whether to continue processing. The administrators at your site can configure the decoder to process or discard messages based on various criteria.

GRIB messages may be discarded based on the model generating the message, the model run time, the valid time of the forecast fields, the geographic location the message represents, the level the data represents, or the meteorological parameter. If disk space is a concern, you can save only those fields that you typically use: 500 and 1000 mb height and temperature fields, for example. Several models are sent in more than one projection. If you only need one of them, configure the decoder so the other projections are discarded.

The configuration file where this information is stored is `NOGRIB.CFG`. This file is read when the data monitor is started. If you change the values in `NOGRIB.CFG`, you must restart the data monitor. If the decoder cannot find `NOGRIB.CFG`, all messages are decoded. The file is has eleven positions separated by the pipe character (|). Below is a description of each position.

NOGRIB.CFG format

Position #	Description
1	Model number to discard. If this value is -1, the model number is not used as selection criteria. This is the value stored in byte 6 of the PDS. The commonly used values are listed below. The `~oper/mcidas/data/gbtbpds001.av1` file contains a complete list of the known values.

Common values	Description
39	Nested Grid Model
64	Regional Optimal Interpolation
77	Spectral Model, Aviation Run
78	Medium Range Forecast Model
83	80 km ETA model
84	40 km ETA model
85	30 km ETA model
86	MAPS model

2	Beginning of the model run time range to discard. If this value is -1, the model run time is not used as selection criteria. This value is stored in byte 16 of the PDS.
3	End of the model run time range to discard.
4	Beginning of the model valid time range to discard. If this value is -1, the valid time of the model is not used as selection criteria. This value is stored in bytes 19 and 20 of the PDS.
5	End of the model valid time range to discard.
6	Beginning of the geographic ID range to discard. If this value is -1, the geographic ID is not used as selection criteria. This value is stored in byte 7 of the PDS. The file `~oper/mcidas/data/gbtbpds001.bv1` contains a list of the IDs and their corresponding geographic coverage and projections.
7	End of the geographic ID range to discard.
8	Beginning of the pressure level range to discard. If this value is -1, the pressure level is not used as selection criteria.
9	End of the pressure level range to discard.

10	Beginning of the parameter number range to discard. If this value is -1, the parameter number is not used as selection criteria. This value is stored in byte 9 of the PDS. The file `~oper/mcidas/data/gbtbpds001.2v2` contains a complete list of the known values. Below is a table of commonly used values.

Common values	Description
1	Pressure
2	Pressure reduced to MSL
7	Geopotential Height
11	Temperature
33	u-component wind
34	v-component wind
52	Relative Humidity

11	End of the parameter number range to discard.

Examples of NOGRIB.CFG entries

```
77 | -1 | -1 | -1 | -1 | 37 | 44 | -1 | -1 | -1 | -1 |
```

This entry discards all fields of the aviation run (77) from projections 37 through 44. These projections are associated with the high resolution "thinned" grid format.

```
39 | -1 | -1 | -1 | -1 | -1 | -1 | 500 | 700 | 52 | 52 |
```

This entry discards relative humidity fields (52) from 500 to 700 millibars for the Nested Grid Model (39).

```
-1 | 12 | 12 | 36 | 42 | -1 | -1 | -1 | -1 | -1 | -1 |
```

This entry discards any field from a 12 UTC model run with a valid time between 36 and 42 hours, inclusive.

Converting GRIB codes

For users to understand GRIB messages, the decoder must change portions of them into meteorological values. For example, a value of 11 in the ninth byte of the PDS is meaningless until it is converted to temperature in degrees Kelvin. Other attributes that must be converted include the geographic location, the forecast time units, and the generating model name and originating location.

Currently, five ASCII file lookup tables are included with the McIDAS-XCD GRIB decoder for this purpose. These files are found in **~oper/mcidas/data** and begin with the characters **gbtbpds** as shown below.

Attribute	Section/Byte	File
processing center	PDS/5	**gbtbpds001.0v1**
parameter/unit	PDS/9	**gbtbpds001.2v2**
forecast time	PDS/18	**gbtbpds001.4v1**
model	PDS/6	**gbtbpds001.av1**
geographic location	PDS/7	**gbtbpds001.bv1**

When the decoder finds a value for one of these attributes, it checks the appropriate lookup table for information about the value. If it cannot find the information, the message is discarded.

Filing the grid in McIDAS

When the unpacking process is complete and the entire GRIB message is successfully decoded, the decoder passes the GRIB structures to DMGRID to be reformatted for McIDAS.

Once the message is converted to McIDAS format, DMGRID uses the **mcrtgrdf** function to determine the grid file for storing the message. The correct grid file is determined by using stored grid header information, consisting of the model, the runtime of the model, and the forecast time, geographic coverage, and the configuration file **RTMODELS.CFG**. If the model information is not explicitly described in **RTMODELS.CFG**, the grid is filed in a scratch grid file. The format of **RTMODELS.CFG** is described on the next page followed by an example.

Once the correct grid file is determined, DMGRID checks if this grid can be filed as is, or if the grid must be pieced together with a previously filed grid. Piecing together is often necessary because most of the gridded fields that cover the globe are sent in 4 or 8 pieces. When a partial grid is received, DMGRID checks if a similar grid has recently been filed. If so, DMGRID pieces the two fields together, refiling the new grid into the same location. If no match is found, the grid is filed as the first grid of this type.

Finally, DMGRID updates the Status Window, telling the administrator that a new grid has been filed and its location. DMGRID then checks the spool file for new data to process.

RTMODELS.CFG format

SCRATCH= a group of 10 grid files for storing grids based on models not specified in RTMODELS.CFG; if SCRATCH=1000, the range of grid files used is 1001 through 1010 based on the Julian day of the model run time

model= *ftype fgridf runint vtint maxvt* **GEO=***min max* information for filing a particular *model*, for example, **NGM** or **AVN**

ftype filing format for this model

- **0** everything from the model is stored in one grid file per model run time; if this value is used, *vtint* and *maxvt* are not necessary
- **1** grids are filed based on model run time and valid forecast time
- **2** all grids from a model run are filed in the same grid file regardless of run time or forecast time; if this value is used, *runint, vtint,* and *maxvt* are not necessary
- **3** same as **1** except no grids are assumed beyond the *maxvt* forecast time

fgridf first grid file in the range to use for this model

runint interval between model run times (hhmmss)

vtint forecast period interval to separate forecast grids

maxvt maximum forecast time, after which all grids are stored in the same grid file

GEO= *min max* range of geographic IDs to store in this grid file range; the file **~oper/mcidas/data/gbtbpds001.bv1** contains a list of the IDs and their corresponding geographic coverage and projections (default=1 255)

RTMODELS.CFG example

If the file **RTMODELS.CFG** contains the following information, messages are stored in the grids listed below.

```
SCRATCH= 411
NGM=     3   101   120000   240000   480000
NGM=     3   141   120000   240000   480000   GEO=211 211
AVN=     1   201   120000   240000   960000
MAPS=    0   301   30000
WWFM=    2   401
```

Grid Files	Model	Run	Forecast Range	Coverage
101 - 110	NGM	00Z	00hr <= Forecast Time <= 24hr	All **except** Regional CONUS (Lambert Conformal)
111 - 120	NGM	00Z	24hr < Forecast Time <= 48hr	
121 - 130	NGM	12Z	00hr <= Forecast Time <= 24hr	
131 - 140	NGM	12Z	24hr < Forecast Time <= 48hr	
141 - 150	NGM	00Z	00hr <= Forecast Time <= 24hr	Regional CONUS (Lambert Conformal)
151 - 160	NGM	00Z	24hr < Forecast Time <= 48hr	
161 - 170	NGM	12Z	00hr <= Forecast Time <= 24hr	
171 - 180	NGM	12Z	24hr < Forecast Time <= 48hr	
201 - 210	AVN	00Z	00hr <= Forecast Time <= 24hr	All
211 - 220	AVN	00Z	24hr < Forecast Time <= 48hr	All
221 - 230	AVN	00Z	48hr < Forecast Time <= 72hr	All
231 - 240	AVN	00Z	72hr < Forecast Time <= 96hr	All
241 - 250	AVN	00Z	> 96hr Forecast Time	All
251 - 260	AVN	12Z	00hr <= Forecast Time <= 24hr	All
261 - 270	AVN	12Z	24hr < Forecast Time <= 48hr	All
271 - 280	AVN	12Z	48hr < Forecast Time <= 72hr	All
281 - 290	AVN	12Z	72hr < Forecast Time <= 96hr	All
291 - 300	AVN	12Z	> 96hr Forecast Time	All
301 - 310	MAPS	00Z	All Forecast Times	All
311 - 320	MAPS	03Z	All Forecast Times	All
321 - 330	MAPS	06Z	All Forecast Times	All
331 - 340	MAPS	09Z	All Forecast Times	All
341 - 350	MAPS	12Z	All Forecast Times	All
351 - 360	MAPS	15Z	All Forecast Times	All
361 - 370	MAPS	18Z	All Forecast Times	All
371 - 380	MAPS	21Z	All Forecast Times	All
401 - 410	WWFM	All	All Forecast Times	All
411 - 420	All other grids			

RTMODELS.CFG defaults

When McIDAS-XCD is installed, the file `RTMODELS.CFG` contains the default values below. These values create the grid files below.

```
SCRATCH= 5001
ETA=     3   5011  120000  240000  480000
NGM=     3   5051  120000  240000  480000
MRF=     1   5101  120000  240000  960000
MAPS=    0   5201  30000
```

Grid File	Model	Run	Forecast Range	Coverage
5001-5010	Miscellaneous grids			
5011-5020	ETA	00Z	00hr <= Forecast Time <= 24hr	All
5021-5030	ETA	00Z	24hr < Forecast Time <= 48hr	All
5031-5040	ETA	12Z	00hr <= Forecast Time <= 24hr	All
5041-5050	ETA	12Z	24hr < Forecast Time <= 48hr	All
5051-5060	NGM	00Z	00hr <= Forecast Time <= 24hr	All
5061-5070	NGM	00Z	24hr < Forecast Time <= 48hr	All
5071-5080	NGM	12Z	00hr <= Forecast Time <= 24hr	All
5081-5090	NGM	12Z	24hr < Forecast Time <= 48hr	All
5101-5110	MRF	00Z	00hr <= Forecast Time <= 24hr	All
5111-5120	MRF	00Z	24hr < Forecast Time <= 48hr	All
5121-5130	MRF	00Z	48hr < Forecast Time <= 72hr	All
5131-5140	MRF	00Z	72hr < Forecast Time <= 96hr	All
5141-5150	MRF	00Z	> 96hr Forecast Time	All
5151-5160	MRF	12Z	00hr <= Forecast Time <= 24hr	All
5161-5170	MRF	12Z	24hr < Forecast Time <= 48hr	All
5171-5180	MRF	12Z	08hr < Forecast Time <= 72hr	All
5181-5190	MRF	12Z	72hr < Forecast Time <= 96hr	All
5191-5200	MRF	12Z	> 96hr Forecast Time	All
5201-5210	MAPS	00Z	All Forecast Times	All
5211-5220	MAPS	03Z	All Forecast Times	All
5221-5230	MAPS	06Z	All Forecast Times	All
5231-5240	MAPS	09Z	All Forecast Times	All
5241-5250	MAPS	12Z	All Forecast Times	All
5251-5260	MAPS	15Z	All Forecast Times	All
5261-5270	MAPS	18Z	All Forecast Times	All
5271-5280	MAPS	21Z	All Forecast Times	All

Configuring the McIDAS-XCD Data Relay

This section provides the following information.

- system requirements for the McIDAS-XCD data relay
- procedures for configuring McIDAS-XCD and McIDAS-MVS for the data relay
- steps for adding a second relay process

The McIDAS-XCD data relay is a software extension included with McIDAS-XCD. It replaces the IBM 3708 protocol converter that ingested the National Weather Service (NWS) Family of Services (FOS) and HRS data. On October 4, 1994, the NWS upgraded the FOS data circuits from 2400 baud to 9600 baud for DDS, IDS, and PPS, and from 19.2 KB to 56 KB for HRS. You should install this package if you plan to ingest FOS or HRS data on your McIDAS-MVS system. The 3708 converters cannot run at high data rates and must be abandoned for this part of the system.

When the system is configured correctly, a McIDAS-MVS program running on the McIDAS-MVS system makes a TCP/IP connection to a port on the Unix workstation running the McIDAS-XCD software. When the connection is established, the command `inetd` starts a program on the Unix workstation that monitors data ingested by McIDAS-XCD. When a circuit receives new data, the Unix program sends a copy of the data to the mainframe. The asynchronous data circuit ingestors on the mainframe read this data stream as if it came from a 3708.

System requirements

To configure the McIDAS-XCD data relay, you must have these system requirements.

- McIDAS-XCD server software, version 7.1 minimum, installed on your Unix workstation according to the system requirements documented in Chapter 2, *McIDAS-XCD Software Installation*
- hardware capable of receiving multiple data circuits at the new NWS baud rates; SSEC recommends the Central Data Corporation STS1008+ SCSI Terminal Server
- TCP/IP for MVS installed and running on your mainframe
- a TCP/IP connection from the McIDAS-XCD workstation to the mainframe
- McIDAS-MVS version 93319 (November 1993) or later

Configuration procedures

The procedures for configuring the McIDAS-XCD workstation and McIDAS-MVS to relay the core set of circuits (DDS, PPS, IDS, and HRS) are described below. If your site receives more than five circuits, follow the procedures below, then complete the procedure titled *Adding another relay circuit.*

Configuring a McIDAS-XCD workstation

1. Login as user **oper** and modify the appropriate keyword values in the McIDAS-XCD file **~oper/mcidas/data/fosrelay.cfg**. This file describes the following.

   - the relay software which circuits to relay
   - the location of the ingested data
   - the maximum number of minutes of buffering to perform if the TCP/IP link goes down between the mainframe and the McIDAS-XCD workstation

   When you first install the McIDAS-XCD package, the file **fosrelay.cfg** looks like the one below. Pound signs (#) indicate comments.

```
# McIDAS-XCD     relay configuration file.

# FOS_PATH           -contains the fully qualified directory where the
#                     *.XCD files are located on the McIDAS-XCD
#                     workstation
# FOS_TEXT           -contains the list of Family of Services text data
#                     circuits that are to be ingested
# BINARY             -contains the list of binary data circuits that are
#                     to be relayed. Note that in this list you include
#                     the fully qualified path name.
# BUF_TIME           -is the maximum number of minutes of buffering that
#                     is to be done when the system restarts when the
#                     mainframe goes down. It is recommended that this
#                     not exceed 60 minutes.

FOS_PATH=/home/mcidas/data
FOS_TEXT=DDS PPS IDS
BINARY=/home/oper/mcidas/data/HRS.SPL
BUF_TIME=30

# Note that in the default configuration listed above, the
# circuits would be given the following protocol assignments:
#
# circuit circuit number  directory where data resides
# ------- --------------  ---------------------------
# DDS       1             /home/mcidas/data
# PPS       2             /home/mcidas/data
# IDS       3             /home/mcidas/data
# HRS       4             /home/oper/mcidas/data
#
# The order of protocol assignments MUST match the ACBnames
# for the ASYNCS listed in MCIDAS.PARMLIB(XCDCORE1)on McIDAS-MVS
```


If your data directory containing the raw ***.XCD** files is **/home2/mcidas/data**, for example, make the change below for the keyword **FOS_PATH**.

```
FOS_PATH=/home2/mcidas/data
```

For steps 2 through 4, you must have root permissions.

2. Add a line to the file **/etc/services** similar to the example below, replacing *NNN* with a unique 3-digit port name not currently used by any other process in **/etc/services**.

```
xcd_rlyc1       NNN/tcp       # xcd core data stream relay 1
```

   Adding this line allows the service to be found throughout the system by name.

   The value for *NNN* will also be entered as the port number in the mainframe member MCIDAS.PARMLIB(XCDCORE1) used in step 4 of the next procedure, *Configuring McIDAS-MVS*.

3. Add the following line to the file **/etc/inetd.conf**. The entry should be one line; it is displayed below as three lines due to space limitations. Each term is defined below.

```
xcd_rlyc1  stream  tcp  nowait  oper
    /home/oper/mcidas/bin/xcdrelaysh  xcdrelaysh
    /home/oper/mcidas/data/fosrelay.cfg  /home/oper
```

Term	Definition
xcd_rlyc1	service name listed in **/etc/services**
stream	socket type
tcp	protocol to use
nowait	command to start an asynchronous server
oper	user name to run the data relay
xcdrelaysh	command script to run at startup
fosrelay.cfg	circuit configuration file
/home/oper	home directory of **oper** account

The information in file **/etc/inetd.conf** tells the system to start the **xcdrelaysh** script when the McIDAS-MVS tries to connect to the *NNN* port on the Unix workstation. The **xcdrelaysh** script is created in **~oper/mcidas/bin** when McIDAS-XCD is installed.

Full pathnames are required for commands to run in the **/etc/inetd.conf** file. If the **oper** account is set up under a file system other than **/home**, it must be reflected in the pathname.

4. Edit the file **~oper/.xcdrlyenv**. This file contains information about the environment required for running the **xcdrelaysh** script; it sets the **MCPATH** environment variable to the path of data directories containing -XCD files. Below is an example of this file.

```
McRoot=/home/mcidas
# DO NOT modify any of the lines below.
# ------------------------------------------------------
MCPATH=$HOME/mcidas/data
MCPATH=$MCPATH:$McRoot/data
export MCPATH
unset McRoot
```

Modify the path set with the **McRoot** environment variable. Change **/home/mcidas** to reflect the home directory of the **mcidas** account.

5. Reinitialize **inetd** so the system configuration changes will take effect. First, determine the PID number for command **inetd** by entering the appropriate command below from a Unix command window.

For Solaris, type: **ps -aux | grep inetd**

For AIX, HPUX, IRIX, type: **ps -ef | grep inetd**

A line similar to the one below is displayed.

```
root     PID  0.0  0.0   56   44 ? S   Aug 26  0:00 inetd
```

Now, reinitialize **inetd** by entering the command below, replacing *PID* with the process ID number displayed above.

Type: **kill -HUP** *PID*

This completes the configuration procedure for your McIDAS-XCD workstation. Now, complete the configuration procedure for McIDAS-MVS.

Configuring McIDAS-MVS

1. Use TSO to create the member XCDCORE1 in the proc library used for console started tasks. The libraries typically used are MCIDAS.PROCLIB, USER.PROCLIB, or SYS1.PROCLIB.

   XCDCORE1 creates the task that makes the -XCD relay connection to the Unix workstation and tells the system which JCL to run when this program is started from the operator's console.

2. Insert these lines in XCDCORE1.

```
//IEFPROC  EXEC  PGM=EM3708,TIME=1440,REGION=128K
//STEPLIB  DD    DISP=SHR,DSN=MCIDAS.APFLIB
//CNTL     DD    DISP=SHR,DSN=MCIDAS.PARMLIB(XCDCORE1)
```

3. Create a member in SYS1.VTAMLST named APPL3708 by inserting the lines below. This defines the VTAM Application Control Blocks (ACBs) for the relay system to use.

```
APPL3708  VBUILD  TYPE=APPL
XCDRLYC1  APPL    AUTH=(ACQ,NVPACE)
XCDRLYL1  APPL    AUTH=(ACQ,NVPACE)
XCDRLYL2  APPL    AUTH=(ACQ,NVPACE)
```

Use the ACB name ending in C1 for core circuits (DDS, PPS, IDS, HRS). Use the ACB names ending in L1 and L2 for all other circuits, which are considered local and require a separate relay system process. See the procedure for *Adding another relay process* in this section for more information.

4. Create a member in MCIDAS.PARMLIB called XCDCORE1. It will consist of one line and contain the following:

Column	Contents	Description
1	XCDRLYC1	ACB name for the -XCD relay to use
9	blank	
10	A	indicates this process is the active open
11	blank	
12	*xxx.xxx.xxx.xxx*	IP address of the McIDAS-XCD workstation: use leading zeros so that each group of *xxx* contains 3 characters
28	*NNN*	port number of the **xcdrelay** program (this will be the same value used in step 2 of *Configuring a McIDAS-XCD workstation*)
31		ACB name of the first ASYNC
39		ACB name of the second ASYNC
46		ACB name of the third ASYNC
53		ACB name of the fourth ASYNC

For example:

```
XCDRLYC1 A 144.092.108.151 502JMBXXK JMBXXI JMBXXH JMBXXC
```

Note that the ACB names for each ASYNC must match the order specified in the circuit configuration file used by the Unix **xcdrelay** program. See step 1 of *Configuring a McIDAS-XCD workstation*.

5. Examine the MCIDAS.PARMLIB members currently used to start your ASYNCs. These members are named in the L= clause when the ASYNCs are started. Create new members using the information found in the current members. For example, SSEC's member named DDS is shown below.

```
JMBXXK  ,L7C      ,AADDS
V NET,ACT,ID=L7C
V NET,INACT,I,ID=L7C
```

The first line has three parameters:

- ACB name for the ASYNC to use (beginning in column 1)
- VTAM LU name of the session partner (beginning in column 9),
- LW spool file name (beginning in column 18)

The second line is a VTAM command to activate the session partner. The third line is a VTAM command to deactivate the session partner.

Copy the PARMLIB member for each circuit into a new member with the number one appended; for example: DDS1.

MCIDAS.PARMLIB(DDS1) should look like this:

```
JMBXXK  ,XCDRLYC1,AADDS
V NET,ACT,ID=XCDRLYC1
D NET,E,ID=JMBXXK
```

On the first two lines, only change the session partner. The third line now contains a listing command.

For the HRS circuit, use AAHDS as the spool file name. Also use the same ACB name for the ASYNC that the NMC products circuit used when it arrived at a 3708 port.

6. Edit the file MCIDAS.PARMLIB(VTAMCMDS) to include the proper VTAM startup command.

```
VARY NET,ACT,ID=APPL3708
```

7. Edit the file MCIDAS.PARMLIB(SYSTCMDS) to include the proper startup commands. Replace your current Family Of Services and HRS ingest process startups with the START ASYNC commands below.

```
START ASYNC.DDS,L=DDS1,TYPE=Z
START ASYNC.PPS,L=PPS1,TYPE=Z
START ASYNC.IDS,L=IDS1,TYPE=Z
START ASYNC.HRS,L=HRS1,TYPE=D
START XCDCORE1
```

8. Enter the commands below from the mainframe master console to stop all currently running ASYNCs.

```
C DDS
C PPS
C IDS
C HRS
```

9. From the mainframe master console, start the relay by manually entering the VTAM command listed in step 6, followed by the commands listed in step 7.

This completes the configuration procedure for McIDAS-MVS.

Adding another relay process

You can process a maximum of five data circuits per relay process using the -XCD relay system.

To send more than five circuits to the mainframe through the relay, you must start a second relay process. Perform the steps below to add a second relay process for the McIDAS-XCD workstation and McIDAS-MVS.

McIDAS-XCD workstation

1. Log in as user **oper** and create a file similar to the file **~oper/mcidas/data/fosrelay.cfg** called **~oper/mcidas/data/localrelay.cfg** containing the necessary information about the local data sources you want to send. See step 1 of *Configuring a McIDAS-XCD workstation*.

2. Edit the file **/etc/services** and add a line similar to the one below, replacing *MMM* with a unique 3-digit port name not currently used by any other process. **xcd_rly11** is the service name for the local relay process. Note the last two characters in the service name are the letter **l** and the number **1**

```
xcd_rly11   MMM/tcp    # xcd local data stream relay 1
```

3. Edit the file **/etc/inetd.conf** and add a line similar to the one below, where **xcd_rly11** is the service name for the local relay process and **/home/oper/mcidas/data/localrelay.cfg** is the local configuration file name created in step 1 above. The entry should be one line; it is displayed below as three lines due to space limitations.

```
xcd_rly11  stream  tcp  nowait  oper
  /home/oper/mcidas/bin/xcdrelaysh  xcdrelaysh
  /home/oper/mcidas/data/localrelay.cfg  /home/oper
```

4. Reinitialize **inetd**. See step 4 of *Configuring a McIDAS-XCD workstation*.

McIDAS-MVS

1. Create a new member in the MVS PROCLIB called XCDLCL1 that looks like this:

```
//IEFPROC  EXEC  PGM=EM3708,TIME=1440,REGION=128K
//STEPLIB  DD    DISP=SHR,DSN=MCIDAS.APFLIB
//CNTL     DD    DISP=SHR,DSN=MCIDAS.PARMLIB(XCDLCL1)
```

2. Create a new MCIDAS.PARMLIB member XCDLCL1. This member will contain one line similar to the MCIDAS.PARMLIB member XCDCORE1 that you created in step 4 of *Configuring McIDAS-MVS*.

3. Change the value for column 1 to XCDRLYL1. Change the value for column 28 to the port number *MMM* used in step 2 on the previous page.

   The contents of XCDLCL1 will look similar to the line below. Make the changes shown in **bold.**

```
XCDRLYL1 A 144.092.108.151 503JMBXXP
```

4. Create PARMLIB members for each new circuit to recognize the correct mainframe software. These member names should have the number one appended to them. See step 5 of *Configuring McIDAS-MVS*.

   For example, create member MCIDAS.PARMLIB(FOO1) as shown below, where FOO is the new circuit. Make the changes shown in **bold**.

```
JMBXXP,XCDRLYL1,AAFOO
V NET,ACT,ID=XCDRLYL1
D NET,E,ID=JMBXXP
```

5. Add the following lines to MCIDAS.PARMLIB(SYSTCMDS) to automatically start the appropriate processes when the system starts up. See step 7 of *Configuring McIDAS-MVS*. Make the changes shown in **bold**.

```
START ASYNC.FOO,L=FOO1,TYPE=Z
START XCDLCL1
```