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 ........................................ 1-3
  Terminology ...................................................... 1-3
  Data receiving and processing ................................ 1-4
  Ingestors ......................................................... 1-4
  Data monitors ..................................................... 1-7
  McIDAS-XCD Status window .................................... 1-9

McIDAS-XCD Software Installation ............................. 2-1
  System requirements ........................................... 2-1
  First-time installation procedures ............................ 2-3
  Adding the scadda group ....................................... 2-3
  Assigning directory permissions .............................. 2-3
  Installation procedures ....................................... 2-4
  Obtaining the McIDAS-XCD software ......................... 2-4
  Loading the McIDAS-XCD software ............................ 2-6
  Configuring the McIDAS-XCD files ............................ 2-8
  Configuring data communications ............................. 2-11
  Activating the GRIB decoder .................................... 2-13
  Starting the McIDAS-XCD package ............................. 2-13
  Configuring the scadda account ............................... 2-15
  Removing the McIDAS-XCD software ........................... 2-17

McIDAS-XCD Administrative Commands ....................... 3-1
  BLDTEXT ......................................................... 3-2
  CHKERR ........................................................... 3-4
  CIRCUIT .......................................................... 3-5
  DATAQ ............................................................ 3-7
  DATAQCV ......................................................... 3-9
  DECFINFO ......................................................... 3-10
  DELWXT .......................................................... 3-12
  DGROUP .......................................................... 3-12
  IMON ............................................................. 3-13
  IDU ............................................................... 3-15
  NMCAMT .......................................................... 3-18
  QRTMDG .......................................................... 3-21
  REMRF ............................................................ 3-22
  REMRF1 ........................................................... 3-24
  SYNTHMC .......................................................... 3-25
  SIGCO .............................................................. 3-27
  STARTXCD ........................................................ 3-28
  STAT ............................................................... 3-29
  SUBGRD ............................................................ 3-30
  UPDIS ............................................................. 3-31
  WMORTE ........................................................... 3-32
  statdisp .......................................................... 3-35

Revised 5/98
McIDAS-XCD Administrator’s Guide
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.

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>client</td>
<td>workstation that requests and receives data from a server workstation</td>
</tr>
<tr>
<td>data block</td>
<td>WMO header description and text data</td>
</tr>
<tr>
<td>data monitor</td>
<td>process that runs one or more decoders</td>
</tr>
<tr>
<td>DDS</td>
<td>Domestic Data Service</td>
</tr>
<tr>
<td>decoder</td>
<td>program that converts raw data into McIDAS data files</td>
</tr>
<tr>
<td>HRS</td>
<td>High Resolution Data Service</td>
</tr>
<tr>
<td>IDS</td>
<td>International Data Service</td>
</tr>
<tr>
<td>ingester</td>
<td>program that receives data through a communications port</td>
</tr>
<tr>
<td>NFS</td>
<td>Network File System</td>
</tr>
<tr>
<td>PPS</td>
<td>Public Products data Service</td>
</tr>
<tr>
<td>server</td>
<td>workstation that stores and supplies data to client workstations</td>
</tr>
</tbody>
</table>
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 $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 ingestas 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 circuit. 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 ccyydddXCD, where cc is the first two characters of the circuit name, and yyyddd is the Julian day. The naming convention of the index file is hhyyyddd.IDX where hh is a 2-character WMO header and yyyddd 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 D1Dyyddd.XCD and ID1Dyddd.XCD, respectively. However, both ingestors write their directory information into the same index file, FTYyyddd.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 U1yyddd.IDX. If a WMO header is not forced into a particular index file, it is filed in the ZZyyddd.IDX file, which is a miscellaneous index.

Binary ingestors

The binary data ingestor, INGBIN, ingests a binary data stream regardless of the data format. INGBIN stores the data in a circular spool file, occ:SPD, where it can be processed by a data monitor. The data that INGBIN 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
# -------
PORT=/dev/tty0 #port tty name (this will vary among workstations)

# input baud rate of the circuit
# output baud rate of the port
# number of stop bits
# number of data bits
# block indefinitely
# block until at least 40 bytes have been read
# generate SIGINT on BREAK
# block until the modem is answered
# receiver enabled and characters can be read
# don't echo characters back to device
# don't visually erase characters
# don't echo KILL
# don't echo new line
# don't hang up line when last process closes the device
# disable canonical mode
# do not convert carriage returns to line feeds
# disable special extended character recognition
# do not ignore NRMK condition
# do not ignore carriage returns
# do not ignore characters with parity errors
# do not convert new line character to carriage return
# don't enable input parity error checking
# disable terminal-generated signals caused by special characters
# strip input to 7 bits
# disable start-stop input control
# do not enable start-stop output control
# flush queue when SIGINT and SIGQUIT are sent
# perform output processing
# disable parity error checking
# don't mark parity errors
# parity error checking set to even (not used)
# don't send SIGTTG for background output
```

Figure 1. DDS Configuration File
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.

<table>
<thead>
<tr>
<th>Monitor</th>
<th>Index</th>
<th>Data type</th>
<th>Decoded Data location</th>
<th>Configuration file</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMMISC</td>
<td>FO</td>
<td>FOUS14</td>
<td>MD file FO14</td>
<td>FO14DEC.CFG</td>
</tr>
<tr>
<td>PT</td>
<td></td>
<td></td>
<td>Rapid-Access System</td>
<td>TERMDEC.CFG</td>
</tr>
<tr>
<td>SD</td>
<td></td>
<td>MDR</td>
<td>Grid files</td>
<td>MDRDEC.CFG</td>
</tr>
<tr>
<td>TB</td>
<td></td>
<td>TIROS NAV</td>
<td>SYSENAV1</td>
<td>TIRDEC.CFG</td>
</tr>
<tr>
<td>UA</td>
<td></td>
<td>PIREP/AIREP</td>
<td>MD file PIRP</td>
<td>PIRPDEC.CFG</td>
</tr>
<tr>
<td>DMRAOB</td>
<td>UJ</td>
<td>TEMP/PILOT</td>
<td>MD file IRAB/RSQ</td>
<td>IRABDEC.CFG</td>
</tr>
<tr>
<td>DMSFC</td>
<td>SA</td>
<td>SAO/METAR</td>
<td>MD file ISFC</td>
<td>ISFDEC.CFG</td>
</tr>
<tr>
<td>DMSYN</td>
<td>SM</td>
<td>SYNOPTIC</td>
<td>MD file SYN</td>
<td>SYNDEC.CFG</td>
</tr>
<tr>
<td>SM</td>
<td></td>
<td>SHIP/DRIBU</td>
<td>MD file ISHP</td>
<td>ISHPDEC.CFG</td>
</tr>
</tbody>
</table>

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.
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 ingesters 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 DD94286.XCD, and that it last filed data in index location 3956 of index file UA942286.IDX. The IP5 circuit filed data in FF94286.IDX.

The example also shows the surface decoder (SAODEC) last updated the bulletin board at 23:48:46 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 (U94286.IDX) was processed because the index pointers (3168) are identical.

<table>
<thead>
<tr>
<th>Decoder Status Display</th>
<th>94286</th>
<th>23485</th>
</tr>
</thead>
<tbody>
<tr>
<td># CIRCUIT INGESTOR TIME BYTE INDEX FILENAME ORIGIN WMO PRODUCT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 DDS INKET 234757 275209 3956 UA942286.IDX KMSC UAA 3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 DDS INKET 234849 140909 8476 FF94286.IDX KANN FPPS 12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 HSS INKET 234819 123456 HSS.SPL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GRID</th>
<th>GRID</th>
<th># DECOR</th>
<th>TIME</th>
<th>BGSSPR</th>
<th>LASSTR</th>
<th>NO</th>
<th>RON</th>
<th>COL</th>
<th>TEXT</th>
<th>INDEX</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>234845</td>
<td>5312</td>
<td>53224</td>
<td>6</td>
<td>67</td>
<td>200</td>
<td></td>
<td>SA94286.IDX</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>234830</td>
<td>3108</td>
<td>3108</td>
<td>26</td>
<td>14</td>
<td>3</td>
<td></td>
<td>UJ92486.IDX</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>234830</td>
<td>12451</td>
<td>12452</td>
<td>96</td>
<td>8</td>
<td>5011</td>
<td></td>
<td>OW94286.IDX</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>234703</td>
<td>512</td>
<td>512</td>
<td>76</td>
<td>0</td>
<td>19</td>
<td>Watch #23</td>
<td>WM94286.IDX</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td>234000</td>
<td>0</td>
<td>0</td>
<td>70</td>
<td>70</td>
<td>4000</td>
<td></td>
<td>HSEC 70 KMSC 40000</td>
</tr>
</tbody>
</table>

Figure 4. Sample Decoder Status Display
The table below defines each field in the decoder status display.

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

McIDAS-XCD Software Installation

The McIDAS-XCD software installation process makes the directories `~oper/acidas/xod7.4`, `~oper/acidas/xod7.4/src` and `~oper/acidas/xod7.4/data`, places the source and data files in them, and builds the software. When the build is complete, the source, help, data, and binaries are copied to the directories `~oper/acidas/src`, `~oper/acidas/help`, `~oper/acidas/data`, and `~oper/acidas/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 acidas account according to the specifications in Chapter 1, Installation and Configuration, in the McIDAS User’s Guide. Be sure to include the directory `~/acidas/bin` in the environment variable PATH.
- The Unix workstation running the McIDAS-XCD software must have the group name `acidas` 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.

<table>
<thead>
<tr>
<th>Circuit/data type</th>
<th>Daily space requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDS circuit</td>
<td>75 MB</td>
</tr>
<tr>
<td>IDS circuit</td>
<td>24 MB</td>
</tr>
<tr>
<td>PPS circuit</td>
<td>26 MB</td>
</tr>
<tr>
<td>Surface hourly MD file (ISFC)</td>
<td>25 MB</td>
</tr>
<tr>
<td>RAOB MD file (IRAB/IRSG)</td>
<td>7 MB</td>
</tr>
<tr>
<td>Synoptic MD file (SYN)</td>
<td>7 MB</td>
</tr>
<tr>
<td>Ship/buoy MD file (ISHP)</td>
<td>4 MB</td>
</tr>
<tr>
<td>FOUS14 MD file (FO14)</td>
<td>2 MB</td>
</tr>
<tr>
<td>PIREP/AIREP/ACARS (PIRP)</td>
<td>5 MB</td>
</tr>
<tr>
<td>Approximate total</td>
<td>169 MB</td>
</tr>
</tbody>
</table>

(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 mcdas group
- assigning directory permissions

Adding the mcdas group
The workstation running the McIDAS-XCD software must have the group name mcdas which contains the user oper. Use this procedure to add the mcdas 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.

   mcdas:gid:1:oper

3. Log out of the root account.

Assigning directory permissions
Use the steps below to assign write privileges to the directory -mcdas/data.

1. Log on to the workstation as user mcdas and change the group for the -mcdas/data directory to the group mcdas.

   Type: chgrp mcdas -mcdas/data

2. Change the privileges for the directory -mcdas/data so only the user mcdas or the members of the group mcdas have write permissions.

   Type: chmod 775 -mcdas/data

3. Log out of the mcdas account.
Installation procedures

The MclDAS-XCD software installation consists of eight tasks:

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

Obtaining the MclDAS-XCD software

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

The MclDAS-XCD 7.4 package contains the following files.

<table>
<thead>
<tr>
<th>File name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>xcd7.4.tar.Z</td>
<td>compressed tar file that contains all source and data files</td>
</tr>
<tr>
<td>xcd_init</td>
<td>shell script that initializes the environmental variables for the -XCD installation</td>
</tr>
<tr>
<td>xcd_chksys</td>
<td>shell script that checks for the proper setting of the environmental variables used during installation</td>
</tr>
<tr>
<td>xcd_install</td>
<td>shell script that installs the MclDAS-XCD software</td>
</tr>
<tr>
<td>xcd_README_7.4</td>
<td>lists information to review before installation</td>
</tr>
</tbody>
</table>

Obtaining MclDAS-XCD from the MUG Web Site

1. Use your Web browser to download the files listed on the previous page. Access the MclDAS User’s Group Web Site at http://www.nsc.ucr.edu/mug, and follow the link for MclDAS-XCD software. Each site has its own login and password for downloading files. Contact the MclDAS Help Desk if you can’t remember yours.

2. Log on to the MclDAS-XCD workstation as user oper and move the downloaded files to the -oper/scidas 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/scidas

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/scidas directory.

   Type: cd -oper/scidas

6. Run the command below for each of the downloaded -XCD files.

   Type: chmod oper file

Obtaining MclDAS-XCD via tape or CD

1. Log on to the MclDAS-XCD workstation as user oper.

2. Change to the -oper/scidas directory.

   Type: cd -oper/scidas

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 McKCD_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 ingesters 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 ~/datadir
   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.

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Active</th>
<th>Comm. Port</th>
<th>Command</th>
<th>Configuration File</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDS</td>
<td>yes</td>
<td>/dev/ttyC0</td>
<td>INGETEXT</td>
<td>DDS.CFG</td>
</tr>
<tr>
<td>PPS</td>
<td>no</td>
<td>/dev/ttyC1</td>
<td>INGETEXT</td>
<td>PPS.CFG</td>
</tr>
<tr>
<td>IDS</td>
<td>no</td>
<td>/dev/ttyC2</td>
<td>INGETEXT</td>
<td>IDS.CFG</td>
</tr>
<tr>
<td>HRS</td>
<td>no</td>
<td>/dev/ttyC3</td>
<td>INGEBIN</td>
<td>HRS.CFG</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Circuit</th>
<th>Active</th>
<th>File</th>
<th>Command</th>
<th>Configuration File</th>
</tr>
</thead>
<tbody>
<tr>
<td>NTXT</td>
<td>yes</td>
<td>/tmp/jmb.fifo.1</td>
<td>INGETEXT</td>
<td>NTXT.CFG</td>
</tr>
<tr>
<td>NBIN</td>
<td>yes</td>
<td>/tmp/jmb.fifo.2</td>
<td>INGEBIN</td>
<td>NBIN.CFG</td>
</tr>
</tbody>
</table>
10. List the active data monitors, their associated decoders, and status.

Type: DECINFO

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

<table>
<thead>
<tr>
<th>Data Monitor</th>
<th>Decoder</th>
<th>MD File</th>
<th>Status</th>
<th>Description</th>
<th>Configuration File</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMSPC</td>
<td>SAODEC</td>
<td>1-10</td>
<td>A</td>
<td>Surface hourly</td>
<td>ISPCDEC.CFG</td>
</tr>
<tr>
<td>DMRAOB</td>
<td>RABDEC</td>
<td>11-30</td>
<td>A</td>
<td>Upper air</td>
<td>IRABDEC.CFG</td>
</tr>
<tr>
<td>DMSYN</td>
<td>SYNDIC</td>
<td>51-66</td>
<td>A</td>
<td>Synoptic</td>
<td>SYNDEC.CFG</td>
</tr>
<tr>
<td>SHPDEC</td>
<td>31-40</td>
<td>A</td>
<td>Ship/Buoy</td>
<td>ISHPDEC.CFG</td>
<td></td>
</tr>
<tr>
<td>DMMISC</td>
<td>F14DEC</td>
<td>41-50</td>
<td>A</td>
<td>FOUSI4</td>
<td>FO14DEC.CFG</td>
</tr>
<tr>
<td>PIRDEC</td>
<td>PIREP/AIREP</td>
<td>61-70</td>
<td>A</td>
<td>PIRPDEC.CFG</td>
<td></td>
</tr>
<tr>
<td>TERDEC</td>
<td>A</td>
<td>Terminal Fort</td>
<td>TERMDIC.CFG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TIRDEC</td>
<td>I</td>
<td>TIROS NAV</td>
<td>TIRDEC.CFG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MDRDEC</td>
<td>A</td>
<td>MDR grids</td>
<td>MDRDEC.CFG</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DMGRID</td>
<td>GRIBDEC</td>
<td>I</td>
<td>NMC GRIDS</td>
<td>GRIBDEC.CFG</td>
<td></td>
</tr>
</tbody>
</table>

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 ingester, there are two options for circuit configuration. If your McIDAS-XCD software is on the same workstation as the NOAAPORT SDI ingester, 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/ttyyn 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/ttyyn` value on the `PORT=` line to the value determined in step 1. The default values are listed below.

<table>
<thead>
<tr>
<th>Path and file name</th>
<th>Installation defaults</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-oper/mcidas/data/DDS.CFG</code></td>
<td>PORT=/dev/ttyC0</td>
</tr>
<tr>
<td><code>-oper/mcidas/data/PPS.CFG</code></td>
<td>PORT=/dev/ttyC1</td>
</tr>
<tr>
<td><code>-oper/mcidas/data/IDS.CFG</code></td>
<td>PORT=/dev/ttyC2</td>
</tr>
<tr>
<td><code>-oper/mcidas/data/HRS.CFG</code></td>
<td>PORT=/dev/ttyC3</td>
</tr>
</tbody>
</table>

3. Log on to the root account and add the following lines to the end of the file `/etc/re.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#7 is set to ttyC7.

   ```bash
   if [ -f /dev/tty#0 ]; then
     chown root /dev/tty#0
     chmod 776 /dev/tty#0
   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 ingester.

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

<table>
<thead>
<tr>
<th>Path and file name</th>
<th>Installation defaults</th>
<th>Circuit Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>~oper/scidas/data/NTXT.CFG</td>
<td>/tmp/jmb.fifo.1</td>
<td>Text</td>
</tr>
<tr>
<td>~oper/scidas/data/NBIN.CFG</td>
<td>/tmp/jmb.fifo.2</td>
<td>Binary</td>
</tr>
</tbody>
</table>

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 ingester workstation.

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

2. Edit the configuration files NTXT.CFG and NBIN.CFG in the
~oper/scidas/data directory. In each file, comment out the FILE= and PORT= 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 ingesters. In each file, change the default value on the HOST= line to the IP or name address of the SDI ingester. You should not have to modify the HOST_PORT= line. The default values are listed below.

<table>
<thead>
<tr>
<th>Path and file name</th>
<th>HOSTug</th>
<th>HOST_PORT</th>
<th>Circuit Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>~oper/scidas/data/NTXT.CFG</td>
<td>127.0.0.1</td>
<td>1501</td>
<td>Text</td>
</tr>
<tr>
<td>~oper/scidas/data/NBIN.CFG</td>
<td>127.0.0.1</td>
<td>1502</td>
<td>Binary</td>
</tr>
</tbody>
</table>

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 open McIDAS-X session.

Type: DECINFO EDIT DMDGRID GRIB ACTIVE CONFIG=GRIBDEC.CFG

Type: DECINFO SET DMDGRID 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 ingesters as needed. If a data monitor or ingester stops, STARTXCD automatically restarts it. If you cancel STARTXCD, cancel the associated data monitors and ingesters. If you exit McIDAS-X, your decoders and ingesters will stop running.

If you activate or deactivate a data monitor or ingester, 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 nnn represents the number of days of data to keep online.

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

Type: SKE #Y 00:01:00 999999 24 "DEWLXST nnn 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 DEWLXST 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/scidas/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 ~scidas/data directory on the server. Use this path in step 4. From an xterm session,

Type: echo ~scidas/data

3. Start a McIDAS-X session.

Type: scidas

4. Create the McIDAS string MCDATA to contain the fully expanded path of the ~scidas/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 XCDADD.E 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 XCDADD.E.BAT

When the message "BATCH : DONE" appears, the installation is complete.
The datasets created by the XCDADDX.BAT batch file are listed below.

<table>
<thead>
<tr>
<th>ADDE dataset</th>
<th>Type</th>
<th>Files</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTGRIDS/ALL</td>
<td>GRID</td>
<td>5001-5400</td>
<td>all model grids decoded by McIDAS-XCD</td>
</tr>
<tr>
<td>RTGRIDS/ETA</td>
<td>GRID</td>
<td>5011-5050</td>
<td>real-time ETA model grids</td>
</tr>
<tr>
<td>RTGRIDS/MRF</td>
<td>GRID</td>
<td>5101-5200</td>
<td>real-time MRF model grids</td>
</tr>
<tr>
<td>RTGRIDS/NEM</td>
<td>GRID</td>
<td>5051-5090</td>
<td>real-time NEM model grids</td>
</tr>
<tr>
<td>RTGRIDS/RUC</td>
<td>GRID</td>
<td>5200-5280</td>
<td>real-time RUC model grids; may not be available on all workstations</td>
</tr>
<tr>
<td>RTGRIDS/MISC</td>
<td>GRID</td>
<td>5001-5010</td>
<td>miscellaneous real-time grids</td>
</tr>
<tr>
<td>RTPTRC/AIRCRAFT</td>
<td>POINT</td>
<td>61-70</td>
<td>real-time AIREP and PIREP data</td>
</tr>
<tr>
<td>RTPTRC/FOUS14</td>
<td>POINT</td>
<td>41-50</td>
<td>real-time FOUS14 data</td>
</tr>
<tr>
<td>RTPTRC/SFCHOURLY</td>
<td>POINT</td>
<td>1-10</td>
<td>real-time surface hourly data</td>
</tr>
<tr>
<td>RTPTRC/SHIPRESS</td>
<td>POINT</td>
<td>31-40</td>
<td>real-time ship and buoy reports</td>
</tr>
<tr>
<td>RTPTRC/SYNOPIC</td>
<td>POINT</td>
<td>51-60</td>
<td>real-time synoptic data</td>
</tr>
<tr>
<td>RTPTRC/UPPERMAND</td>
<td>POINT</td>
<td>11-20</td>
<td>real-time mandatory level RAOB data</td>
</tr>
<tr>
<td>RTPTRC/UPPERSIG</td>
<td>POINT</td>
<td>21-30</td>
<td>real-time significant level RAOB data</td>
</tr>
<tr>
<td>RTPMTXT/TEXT</td>
<td>TEXT</td>
<td>(OBTX)</td>
<td>default dataset name used by the SFCRPT command.</td>
</tr>
<tr>
<td>RTKMTXT/SYNOPIC</td>
<td>TEXT</td>
<td>(OBTX)</td>
<td>default dataset name used by the SYNRP command</td>
</tr>
<tr>
<td>RTKMTXT/TERMCAST</td>
<td>TEXT</td>
<td>(OBTX)</td>
<td>default dataset name used by the TAFRPT command</td>
</tr>
<tr>
<td>RTKMTXT/UPPERAIR</td>
<td>TEXT</td>
<td>(OBTX)</td>
<td>default dataset name used by the RAORPT command</td>
</tr>
</tbody>
</table>

**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 redirection 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/scidas`.
   
   **Type:** cd /oper/scidas

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 scidas.
   
   **Type:** su scidas

6. Remove the weather text and observation servers.
   
   **Type:** ./xcd_uninstall addservers

7. Exit from user scidas.
   
   **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/scidas/src, /oper/scidas/data, and /oper/scidas/bin directories. It then removes the directory `~/oper/scidas/xcd/4.2` and its contents. The only remaining files are `xod.tar.gz` 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 opex. 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 3-2
- **CHKERR** lists the output from an error file 3-4
- **CIRCUIT** data circuit utility 3-5
- **DATACQ** plots data availability from MD files 3-7
- **DATAECCV** plots MD file data on a multiple-panel display 3-9
- **DECINFO** decoder utility 3-10
- **DELWXT** deletes weather text and index files 3-12
- **ENGBUP** ID group utility 3-12.1
- **IDMON** station ID monitoring utility 3-13
- **IDU** station dictionary utility 3-15
- **NMCAMT** lists the number of real-time grids received 3-18
- **QRTMDG** deletes real-time grid or MD files 3-21
- **REMRF** regrids MRF data to a lower resolution 3-22
- **REMRF1** reformats MRF grids to low resolution 3-24
- **SENMMC** sends real-time grids to the mainframe 3-25
- **SIGCO** significant level upper air storage utility 3-27
- **STARTXCD** starts the ingester and decoder programs 3-28
- **STAT** lists the decoder and ingester status 3-29
- **SUBGRD** creates geographic subsectors of Mercator grids 3-30
- **UPDIDS** updates the station reporting list 3-31
- **WMO]:TE** maintains a data routing table of WMO headers 3-32
- **stamede**: Unix command for displaying the status window 3-35
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 mtxt (default=32)
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=2)

**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=MASTER.ID.DAT)

**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, ROAB, 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 ROAB.RAP RAOB.RAT 1500 5 IDN 4 3 80 RAOB  
This entry builds the pointer file ROAB.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 ROAB.RAT. The IDs are stored as station block numbers. The ID list built for the ROAB.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 TERMF CST.RAP TERMF CST.RAT 2500 4 C4 6 1 80 TERMF CST X 8  
This entry builds the pointer file TERMF CST.RAP which stores up to four reports per observation time and keeps up to six observation times available. The TERMF CST decoder builds the station list; the maximum size of the text file generated is eight megabytes. The raw text is stored in the file TERMF CST.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 ingester to use; for example, INGETEXT or INGBIN
SPOOL= spool file name; used for the INGBIN ingesters

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 ingesters, wait for the circuit to stop, and then activate the circuit for the associated ingesters.
**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** reporting and missing data color levels (default=7 5)
- **ELE** TV element range for the data plot
- **GRA** graphics frame number (default=current)
- **LAT** latitude range to define the map
- **LIN** TV line range for the data plot
- **LON** longitude range to define the map
- **MDC** 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)
To plot the MD file data availability, the TIME, MOD, LAT and LON keys must be in the following locations in the MD file:

<table>
<thead>
<tr>
<th>Key</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME</td>
<td>row header</td>
</tr>
<tr>
<td>MOD</td>
<td>data section</td>
</tr>
<tr>
<td>LAT</td>
<td>column header or data section</td>
</tr>
<tr>
<td>LON</td>
<td>column header or data section</td>
</tr>
</tbody>
</table>

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**

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

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

**DATAQC 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 DATAQC to plot acquired MD file data in a multiple-panel display.

The table below lists the default setting for each decoder.

<table>
<thead>
<tr>
<th>Decoder</th>
<th>Default plotting time</th>
<th>MD files</th>
<th>Map</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISFC</td>
<td>Nearest hour observation</td>
<td>1-10</td>
<td>World</td>
</tr>
<tr>
<td>IRAB</td>
<td>Nearest 12-hour observation</td>
<td>11-20</td>
<td>World</td>
</tr>
<tr>
<td>ISHP</td>
<td>Nearest hour observation</td>
<td>31-40</td>
<td>World</td>
</tr>
<tr>
<td>FO14</td>
<td>Nearest 12-hour observation</td>
<td>41-50</td>
<td>USA</td>
</tr>
<tr>
<td>SYN</td>
<td>Nearest 6-hour observation</td>
<td>51-60</td>
<td>World</td>
</tr>
<tr>
<td>PIRP</td>
<td>Nearest hour observation</td>
<td>61-70</td>
<td>World</td>
</tr>
</tbody>
</table>

**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

**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:

<table>
<thead>
<tr>
<th>Data Monitor</th>
<th>Active</th>
<th>Decoder</th>
<th>Active</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMMISC</td>
<td>Yes</td>
<td>F14DEC</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WBXDEC</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PEDDEC</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TERDEC</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MDRDEC</td>
<td>Yes</td>
</tr>
<tr>
<td>DMRAOB</td>
<td>Yes</td>
<td>RABDEC</td>
<td>Yes</td>
</tr>
<tr>
<td>DMSFC</td>
<td>Yes</td>
<td>SAODEC</td>
<td>Yes</td>
</tr>
<tr>
<td>DMSYN</td>
<td>Yes</td>
<td>SYNDRC</td>
<td>Yes</td>
</tr>
<tr>
<td>DMRGRID</td>
<td>No</td>
<td>GRIB</td>
<td>No</td>
</tr>
</tbody>
</table>

**DECINFO ADD LOCAL DEC=F0US67 TORNADO**

This entry adds the data monitor LOCAL to the F0US67 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 F0US67 ACTIVE "FOUS67 Decoder**

This entry activates the F0US67 decoder running under the data monitor LOCAL and attaches the label F0US67 Decoder.

**DECINFO EDIT LOCAL F0US67 CONFIG=F0US67.CFG**

This entry changes the name of the configuration file for the F0US67 decoder to F0US67.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 06:05 999999 24 "DELWXT 3"
This entry schedules DELWXT to run every 24 hours at 06:05 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=96017
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)
IDGROUP

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, FOU514, 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: OLDDtype.IDM and NEWtype.IDM where type is one of the following data types.

<table>
<thead>
<tr>
<th>Data type</th>
<th>Decoder</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISFC</td>
<td>SAOMETAR</td>
</tr>
<tr>
<td>IRAB</td>
<td>upper air</td>
</tr>
<tr>
<td>SYN</td>
<td>synoptic</td>
</tr>
<tr>
<td>FOUS14</td>
<td>FOUS14</td>
</tr>
<tr>
<td>PIRP</td>
<td>pilot report</td>
</tr>
<tr>
<td>TERM</td>
<td>terminal forecast</td>
</tr>
</tbody>
</table>
File OLDtype.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 NEWtype.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 NEWIRBD.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 SYNEC.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= circuit names to activate for a decoder
   (default=all)
CO= 2-character country code
DEC= decoders to add to or delete from a station list
ELE= 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= primary and secondary station latitudes
LON= primary and secondary station longitudes
ST= 2-character state code
SWI= 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.
NMCA MT

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

Format
NMCA MT [keywords]

Keywords
DAY= Julian day, YYDD (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=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
NMCA MT 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.

The projection of an MRF grid determines how it is sent. Grids that are 145 rows by 269 columns are sent in the following eight sectors:

<table>
<thead>
<tr>
<th>Sector</th>
<th>Latitude range</th>
<th>Longitude range</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0°N-90°N</td>
<td>60°E-360°W</td>
</tr>
<tr>
<td>2</td>
<td>0°N-90°N</td>
<td>150°E-60°E</td>
</tr>
<tr>
<td>3</td>
<td>0°N-90°N</td>
<td>120°W-150°E</td>
</tr>
<tr>
<td>4</td>
<td>0°N-90°N</td>
<td>30°W-120°W</td>
</tr>
<tr>
<td>5</td>
<td>0°S-90°S</td>
<td>60°E-30°W</td>
</tr>
<tr>
<td>6</td>
<td>0°S-90°S</td>
<td>150°E-60°E</td>
</tr>
<tr>
<td>7</td>
<td>0°S-90°S</td>
<td>120°W-150°E</td>
</tr>
<tr>
<td>8</td>
<td>0°S-90°S</td>
<td>30°W-120°W</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Sector</th>
<th>Latitude range</th>
<th>Longitude range</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0°N-90°N</td>
<td>180°E-0°E</td>
</tr>
<tr>
<td>2</td>
<td>0°N-90°N</td>
<td>0°W-180°W</td>
</tr>
<tr>
<td>3</td>
<td>0°S-90°S</td>
<td>180°E-0°E</td>
</tr>
<tr>
<td>4</td>
<td>0°S-90°S</td>
<td>0°W-180°W</td>
</tr>
</tbody>
</table>

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.
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.
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= grid/ run time fcast
grid/ 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)
fcast 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=999991 - 999995)
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 orepo. Do not run this command from the command line.

High 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.

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 SKNNMC.

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 sgрид/ bgrid egrid dgrid/ [keywords]
```

**Parameters**

- `sgрид/` source grid file to reformat (no default)
- `bgrid` beginning grid number to reformat (default=1)
- `egrid` ending grid number to reformat (default=all)
- `dgrid/` 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 dgrid/ [keywords]
```

**Parameter**

- `dgrid/` 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
- **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=3000)
- **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 opac. Do not run this command from the command line.

High resolution grids are 146-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 06 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 CCODR 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 agrid bgrid egrid dgrid [keywords]

Parameters

agrid source grid file
bgrid beginning grid number to subsect (default=1)
egrid ending grid number to subsect (default=all)
dgrid 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=agrid-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 minum 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
minum 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 89002, the file name for the index file is CS90002.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=6x12)
-geometry +offset+offset 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/acidas/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 (+).
statdisp -bg white -fg black -warn magenta 4
This entry starts the status display with a white background, black foreground, and magenta warning messages from the Unix command window. The 4 (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 DELOXT 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 -acidas/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 -acidas/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/ncidas/bla/xcdrelay.xcdrelay
/home/oper/ncidas/data/fosrelay.cfg
```

Also, read the error file `/tmp/xcdrelayxxxxxx` 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/ncidas/data/fosrelay.cfg` may not point to the correct directory. Check that the paths specified by the `PBS_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/xcdrelayxxxxxx` 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.conf`.

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
graphic 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 has eleven positions
separated by the pipe character (|). Below is a description of each position.
NOGRIB.CFG format

<table>
<thead>
<tr>
<th>Position #</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>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/acidas/data/gbt2pds001.bv1 file contains a complete list of the known values.</td>
</tr>
</tbody>
</table>

**Common values**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>39</td>
<td>Nested Grid Model</td>
</tr>
<tr>
<td>64</td>
<td>Regional Optimal Interpolation</td>
</tr>
<tr>
<td>77</td>
<td>Spectral Model, Aviation Run</td>
</tr>
<tr>
<td>78</td>
<td>Medium Range Forecast Model</td>
</tr>
<tr>
<td>83</td>
<td>80 km ETA model</td>
</tr>
<tr>
<td>84</td>
<td>40 km ETA model</td>
</tr>
<tr>
<td>85</td>
<td>30 km ETA model</td>
</tr>
<tr>
<td>86</td>
<td>MAPS model</td>
</tr>
<tr>
<td>2</td>
<td>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.</td>
</tr>
<tr>
<td>3</td>
<td>End of the model run time range to discard.</td>
</tr>
<tr>
<td>4</td>
<td>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.</td>
</tr>
<tr>
<td>5</td>
<td>End of the model valid time range to discard.</td>
</tr>
<tr>
<td>6</td>
<td>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/acidas/data/gbt2pds001.bv1 contains a list of the IDs and their corresponding geographic coverage and projections.</td>
</tr>
<tr>
<td>7</td>
<td>End of the geographic ID range to discard.</td>
</tr>
<tr>
<td>8</td>
<td>Beginning of the pressure level range to discard. If this value is -1, the pressure level is not used as selection criteria.</td>
</tr>
<tr>
<td>9</td>
<td>End of the pressure level range to discard.</td>
</tr>
</tbody>
</table>

**Examples of NOGRIB.CFG entries**

```
77 | -1 | -1 | -1 | -1 | -1 | -1 | -1 | 37 | 44 | -1 | -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 | -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 | -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/acidas/data and begin with the characters gbtbpdf as shown below.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Section/Byte</th>
<th>File</th>
</tr>
</thead>
<tbody>
<tr>
<td>processing center</td>
<td>PDS/5</td>
<td>gbtbpdf001.0v1</td>
</tr>
<tr>
<td>parameter/unit</td>
<td>PDS/9</td>
<td>gbtbpdf001.2v2</td>
</tr>
<tr>
<td>forecast time</td>
<td>PDS/18</td>
<td>gbtbpdf001.4v1</td>
</tr>
<tr>
<td>model</td>
<td>PDS/6</td>
<td>gbtbpdf001.av1</td>
</tr>
<tr>
<td>geographic location</td>
<td>PDS/7</td>
<td>gbtbpdf001.bv1</td>
</tr>
</tbody>
</table>

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 sortgrid 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 RMODELS.CFG. If the model information is not explicitly described in RMODELS.CFG, the grid is filed in a scratch grid file. The format of RMODELS.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 spoil 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= fgrid runint stinit maxst GEO= min max information for filing a particular model, for example, NGM or AVN

fgrid= 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, stinit and maxst 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, stinit, and maxst are not necessary

3 same as 1 except no grids are assumed beyond the maxst forecast time

runint= first grid file in the range to use for this model

stinit= interval between model run times (hhmmss)

stint= forecast period interval to separate forecast grids

maxst= 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 /opt/mcdas/data/gtbtbgs001.bvl contains a list of the IDs and their corresponding geographic coverage and projections (default=1 250)

RTMODELS.CFG example

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

<table>
<thead>
<tr>
<th>SCRATCH</th>
<th>NGM</th>
<th>AVN</th>
<th>MAPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>411</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1000000</td>
<td>240000</td>
<td>480000</td>
<td></td>
</tr>
<tr>
<td>1410000</td>
<td>240000</td>
<td>480000</td>
<td>GEO= 211, 211</td>
</tr>
<tr>
<td>2010000</td>
<td>240000</td>
<td>960000</td>
<td></td>
</tr>
<tr>
<td>3013000</td>
<td>401400</td>
<td>5-420</td>
<td>All other grids</td>
</tr>
</tbody>
</table>

Grid Files | Model | Run | Forecast Range | Coverage |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>101-110</td>
<td>NGM</td>
<td>00Z</td>
<td>00hr &lt; Forecast Time &lt;= 24hr</td>
<td>All except Regional CONUS (Lambert Conformal)</td>
</tr>
<tr>
<td>111-120</td>
<td>NGM</td>
<td>00Z</td>
<td>24hr &lt; Forecast Time &lt;= 48hr</td>
<td>All</td>
</tr>
<tr>
<td>121-130</td>
<td>NGM</td>
<td>12Z</td>
<td>00hr &lt; Forecast Time &lt;= 24hr</td>
<td>All</td>
</tr>
<tr>
<td>131-140</td>
<td>NGM</td>
<td>12Z</td>
<td>24hr &lt; Forecast Time &lt;= 48hr</td>
<td>All</td>
</tr>
<tr>
<td>141-150</td>
<td>NGM</td>
<td>00Z</td>
<td>00hr &lt; Forecast Time &lt;= 24hr</td>
<td>All</td>
</tr>
<tr>
<td>151-160</td>
<td>NGM</td>
<td>00Z</td>
<td>24hr &lt; Forecast Time &lt;= 48hr</td>
<td>All</td>
</tr>
<tr>
<td>161-170</td>
<td>NGM</td>
<td>12Z</td>
<td>00hr &lt; Forecast Time &lt;= 24hr</td>
<td>All</td>
</tr>
<tr>
<td>171-180</td>
<td>NGM</td>
<td>12Z</td>
<td>24hr &lt; Forecast Time &lt;= 48hr</td>
<td>All</td>
</tr>
<tr>
<td>201-210</td>
<td>AVN</td>
<td>00Z</td>
<td>00hr &lt; Forecast Time &lt;= 24hr</td>
<td>All</td>
</tr>
<tr>
<td>211-220</td>
<td>AVN</td>
<td>00Z</td>
<td>24hr &lt; Forecast Time &lt;= 48hr</td>
<td>All</td>
</tr>
<tr>
<td>221-230</td>
<td>AVN</td>
<td>00Z</td>
<td>48hr &lt; Forecast Time &lt;= 72hr</td>
<td>All</td>
</tr>
<tr>
<td>231-240</td>
<td>AVN</td>
<td>00Z</td>
<td>72hr &lt; Forecast Time &lt;= 96hr</td>
<td>All</td>
</tr>
<tr>
<td>241-250</td>
<td>AVN</td>
<td>00Z</td>
<td>96hr &lt; Forecast Time</td>
<td>All</td>
</tr>
<tr>
<td>251-260</td>
<td>AVN</td>
<td>12Z</td>
<td>00hr &lt; Forecast Time &lt;= 24hr</td>
<td>All</td>
</tr>
<tr>
<td>261-270</td>
<td>AVN</td>
<td>12Z</td>
<td>24hr &lt; Forecast Time &lt;= 48hr</td>
<td>All</td>
</tr>
<tr>
<td>271-280</td>
<td>AVN</td>
<td>12Z</td>
<td>48hr &lt; Forecast Time &lt;= 72hr</td>
<td>All</td>
</tr>
<tr>
<td>281-290</td>
<td>AVN</td>
<td>12Z</td>
<td>72hr &lt; Forecast Time &lt;= 96hr</td>
<td>All</td>
</tr>
<tr>
<td>291-300</td>
<td>AVN</td>
<td>12Z</td>
<td>96hr &lt; Forecast Time</td>
<td>All</td>
</tr>
<tr>
<td>301-310</td>
<td>MAPS</td>
<td>00Z</td>
<td>All Forecast Times</td>
<td>All</td>
</tr>
<tr>
<td>311-320</td>
<td>MAPS</td>
<td>03Z</td>
<td>All Forecast Times</td>
<td>All</td>
</tr>
<tr>
<td>321-330</td>
<td>MAPS</td>
<td>06Z</td>
<td>All Forecast Times</td>
<td>All</td>
</tr>
<tr>
<td>331-340</td>
<td>MAPS</td>
<td>09Z</td>
<td>All Forecast Times</td>
<td>All</td>
</tr>
<tr>
<td>341-350</td>
<td>MAPS</td>
<td>12Z</td>
<td>All Forecast Times</td>
<td>All</td>
</tr>
<tr>
<td>351-360</td>
<td>MAPS</td>
<td>15Z</td>
<td>All Forecast Times</td>
<td>All</td>
</tr>
<tr>
<td>361-370</td>
<td>MAPS</td>
<td>18Z</td>
<td>All Forecast Times</td>
<td>All</td>
</tr>
<tr>
<td>371-380</td>
<td>MAPS</td>
<td>21Z</td>
<td>All Forecast Times</td>
<td>All</td>
</tr>
<tr>
<td>401-410</td>
<td>WWFM</td>
<td>All</td>
<td>All Forecast Times</td>
<td>All</td>
</tr>
</tbody>
</table>
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 ingests 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 `start` 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; SSBC 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 HRB) 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 /open/midas/data/foorselay.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 foorselay.cfg looks like the one below. Pound signs (#) indicate comments.

```plaintext
# McIDAS-XCD relay configuration file.
# FOR_PATH -contains the fully qualified directory where the
# *.XCD files are located on the McIDAS-XCD workstation
# FOR_TEXT -contains the list of Family of Services text data
# BINARVY -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.

FOR_PATH=/home/midas/data
FOR_TEXT=DDS PPS IDS
BINARVY=/home/oper/midas/data/KRS.SPL
BUF_TIME=30

# Note that in the default configuration listed above, the
# circuits would be given the following protocol assignments:

# circuit number directory where data resides
# DDS 1 /home/midas/data
# PPS 2 /home/midas/data
# IDS /home/midas/data
# KRS 4 /home/oper/midas/data

# The order of protocol assignments MUST match the ACN names
# for the Async defined in MCDAP.PARMLIB(XCDCORE) on McIDAS-MVS
```

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

```
FOR_PATH=/home2/midas/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_rlycl  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 MCDAP.PARMLIB(XCDCORE)1 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_rlycl stream tcp nowait oper
    /home/oper/midas/bin/xcdrelaysh xcdrelaysh
    /home/open/midas/data/foorselay.cfg /home/oper
```

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>xcd_rlycl</td>
<td>service name listed in /etc/services</td>
</tr>
<tr>
<td>stream</td>
<td>socket type</td>
</tr>
<tr>
<td>tcp</td>
<td>protocol to use</td>
</tr>
<tr>
<td>nowait</td>
<td>command to start an asynchronous server</td>
</tr>
<tr>
<td>oper</td>
<td>user name to run the data relay</td>
</tr>
<tr>
<td>xcdrelaysh</td>
<td>command script to run at startup</td>
</tr>
<tr>
<td>foorselay.cfg</td>
<td>circuit configuration file</td>
</tr>
<tr>
<td></td>
<td>/home/oper</td>
</tr>
</tbody>
</table>

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 /open/midas/bin when McIDAS-XCD is installed.
Full pathnames are required for commands to run in the /etc/inetd.conf file. If the opem account is set up under a file system other than /home, it must be reflected in the pathname.

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

```
MCPATH=/home/mcidas
MCPATH=/home/mcidas/data
MCPATH=${HOME}/mcidas/data
export MCPATH
unset MCSRoot
```

Modify the path set with the MCSRoot 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 7 8 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  PARM=EVERYDAY,TIME=0000,REGION=128K
/STEPLIB DD  DISP=SHR,DATA=MCIDAS,AFFILIB
/CTRL DD  DISP=SHR,DATA=MCIDAS,PARMLIB(XCDCORE1)
```

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

```
APPL3708 VIPLIB TYPE=APPL
XCDRLYCL APPL AUTH=(AQ,SVFACES)
XCDRLYCL2 APPL AUTH=(AQ,SVFACES)
```

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 XCDCOREI. It will consist of one line and contain the following:

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

For example:

```
XCDRLYCI A 144.092.108.151 SDD JMBXXJ JMBXXJ JMBXXJ
```

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.

```
JMBXXJ L7C AKDS
V NET,ACT,Id=L7C
V NET,JMSCY1,Id=LOJC
```

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:

```
JMBXXJ XCDRLYCI AKDS
V NET,ACT,Id=LOJC
D NET,E,Id=JMBXXJ
```

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.

```
VAR 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,M=DDS1,TYPE=2
START ASYNC.PPS,L=PPS1,TYPE=2
START ASYNC.IDE,L=IDE1,TYPE=2
START ASYNC.HRS,L=HRS1,TYPE=2
START XCDCOREI
```

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

```
C DDS
C PPS
C IDE
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/ncidas/data/fixrelay.cfg called
   -oper/ncidas/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_ryll1 is the service name for the local relay process. Note the last two characters in the service name are the letter 1 and the number 1

   `xcd_ryll1 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_ryll1 is the service name for the local relay process and /home/oper/ncidas/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_ryll1 stream tcp nowait oper
    /home/oper/ncidas/bin/xcdrelaysh xcdrelaysh
    /home/oper/ncidas/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:

   `//LEFFPROC EMEC PARM=E370S,TMS=1440,REGION=128K
    //STEP1 LIB DD DISP=SHR,STM=NCIDAS,APPLIB
    //CMTL DD DISP=SHR,STM=NCIDAS,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 XCDRLY1L. 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.

   `XCDRLY1L 144.092.108.151 503:MMMXXP`

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.

   `MMMXXP MCIDSLCL1.AAPFOO
    V NET,ACT,ID=XCDRLY1L
    D NET,E,ID=MMMXXP`

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,FOO1,L=FOO1,TYPE=Z
    START XCDLCL1`