

TITAN USERS GUIDE

Catalog of Applications

Mike Dixon
Research Applications Laboratory
National Center for Atmospheric Research
Boulder Colorado USA

Introduction

This document lists the most important TITAN applications, grouped by functionality.

Algorithms

The following applications perform algorithms on data sets. These are the brains of TITAN, the rest of the applications are intended as support for these algorithms.

Application	Description
Titan	Thunderstorm Identification, Tracking, Analysis and Nowcasting application. It identifies storms in 3-D radar data stored in MDV format, tracks the storms and forecasts their position using extrapolation.
ctrec	ctrec tracks echo motion using a cross-correlation technique to match pattern movement from one image to the next in time. Input is radar reflectivity, or a similar field (such as satellite IR) in MDV format. Output is U/V fields in MDV format, as well as a copy of the original tracked field.
ApRemoval	Identifies Anomalous Propagation (AP) echoes in a DSR-format radar data FMQ. It filters the AP and writes the filtered data back to the FMQ.
BrightBand	Filters bright-band reflectivity from radar data in a Cartesian grid MDV file.
MdvMedian	Computes the median value, at each grid point, for a time series of MDV files. Intended to compute median clutter values for a clutter map. ClutterRemove is then used to remove the clutter which does not exceed the median values.
ClutterAuto	ClutterAuto computes the auto-correlation for points in a series of MDV files. The auto-correlation, standard deviation and mean data are written out to MDV files. Points with high correlation are

	regarded as having a high likelihood of representing clutter. The output files are used by ClutterRemove.
ClutterRemove	Filters reflectivity data in an MDV file using a median file computed by MdvMedian. Reflectivity at a point is filtered if it does not exceed the median value by a specified margin.
stratiform_filter	Attempts to filter the stratiform regions from radar data stored in an MDV file.
tops_filter	Filters regions with low tops from radar data stored in an MDV file.
PrecipAccum	Calculates accumulated precipitation depth from radar files in MDV format. The 2-D precipitation fields are written out to MDV files.
Mdv2Vad	Compute Velocity Azimuth Display sounding (VAD) from MDV polar radar file. Writes the output as sounding in SPDB.
MdvRadarShear	Compute azimuthal and radial shear from velocity field in MDV polar radar file. Output is written to polar MDV file.
Mdvx2Vil	Computes VIL (vertically-integrated liquid) and related values from an MDV file, and stores these in an output MDV file.
NoiseFilter	Performs simple noise filtering on radar data in a DSR FMQ, based on data values in a single radar field.
FilterStorms	Filters storms in a TITAN SPDB database. The storms can be filtered by size or by information in an associated likelihood grid.
GridForecast	makes a forecast using a gridded field, such as radar reflectivity, and a field of U/V velocity values. Input and output data are both in MDV format.
TitanVectors2Mdv	Convert the vectors in a TITAN SPDB database to an MDV file with U and V fields.
TstormTemporalSmooth	Perform simple temporal smoothing on the motion fields in a TITAN SPDB database.
UpdateStormFields	Used to update certain field values in one TITAN SPDB database using the same field values from the closest storm in another TITAN SPDB database. The database being updated is referred to as the storm database while the database that provides the new field values is referred to as the field database.

UpdateStormSpeeds	Used to nudge the speeds of storms in a given TITAN SPDB database by a given factor. For example, this program could be used to increase all of the storm speeds in a database by 10%.
BasinPrecip	Computes the precipitation for defined basins, based on the geographical extent of the basins and a precipitation field provided in MDV format. The output is written to SPDB.
combine_ltg_radar	Combines lightning data with a radar grid (dbz or vil) to form a pseudo-dbz or pseudo-vil grid. Allows you to use lightning to augment radar data where it is sparse.

TITAN Analysis applications

The following applications are intended for analysis of TITAN algorithms.

Application	Description
StormInit2Field	StormInit2Field reads the SPDB GenPts written by StormInitLocation and writes out gridded MDV files with a gaussian placed at each StormInit point.
StormInitBdry	Designed to facilitate the study of the relation between storm initiation and boundaries. The storm initiation data comes from the output of the StormInitLocation program, and the boundary information typically comes from the colide algorithm.
StormInitClimatology	StormInitClimatology reads the storm initialization data written by StormInitLocation and writes a climatology gridded field to an MDV database. The -yesterday option can be used to run the program with cron on yesterday's data in near realtime.
StormInitDetect	This can be thought of as a lightweight version of StormInitLocations which operates at a higher speed (but gives results that are less complete).
StormInitFieldExtract	Takes an MDV field and a StormInit SPDB data base, extracts the value of selected fields at the init point and time. This is intended to allow the user to investigate the field values relevant to storm initiation.
StormInitLocDump	Prints the location of the initial storm location to stdout.

StormInitLocation	Reads a Titan database, determines those storms which are considered significant using longevity and storm area criteria, and write out the lat/lon at which these storms initiated to an SPDB database.
Storms2Xml	Reads a Titan database and writes XML files representing the storms tracked by the Titan system.
TitanSelect	Selects specified Titan track data from the database and prints it to stdout. It watches the Rview shmем segments for user activity and reacts when a new track is specified.
TrackMatch	Intended for use in evaluation of operational weather modification efforts. In a randomized experiment, PartialProps and CaseStats are used to evaluate the differences between the seed and no/seed pool. TrackMatch is intended to provide the 'no-seed' pool in an operational experiment by finding partner storms to actual cases which have similar characteristics as the seeded storm at decision time. (Still under development.)
Tracks2Ascii	Prints out storm or track data in column ASCII format, suitable for use by other applications such as spread-sheets. Storm properties are selected using the TRACK_ENTRY option, while aggregate track properties are selected using the COMPLETE_TRACK option.
TrackGridStats	Computes geographically-distributed statistics from titan storm and track files. Output is in MDV format.
VerifyGrid	Computes verification statistics for gridded forecasts. The forecast and truth grids are read in from MDV files. Output is printed to stdout.
VerifyTracks	Computes verification statistics for TITAN tracks and prints them to stdout. Also updates verification values in the track files. You need to run VerifyTracks before Tracks2Ascii if you want access to the POD, FAR and CSI verification values in the track files.
PartialProps	Computes the properties for partial storm tracks defined in the case tracks file. A 'partial track' is that part of a track considered relevant when analyzing storm track cases, for example in weather modification evaluation. A partial track consists of all elements of a track which exist between a specified period before the seeding decision time and a specified period after seeding decision time. The partial track is intended to capture that part of a track which (a) characterizes the track before seeding and (b) allows analysis of the effects of seeding, but does not include effects which are not

	attributable to seeding.
CaseStats	Computes statistics for the analysis of weather modification efforts. Uses the re-randomization technique to compute the probability that the effect was by chance. Requires randomized pools of seed/no-seed cases. Must run PartialProps first to compute the properties of the partial tracks for each case. The app reads in the partial properties files, computes the statistics and writes to output files in ASCII.
TstormTemporalSmooth	Reads a TITAN storm data base, and writes a derived data base in which the motion vectors have been smoothed over time.
UpdateStormSpeeds	Reads a TITAN storm data base, and writes a derived data base in which the motion vectors have been scaled by a constant value. For example, you can increase/decrease the speed by some factor.

System applications and scripts

The following application and scripts are used by the basic TITAN system, especially in REALTIME mode.

Application / script	Description
DataMapper	Server that maintains state table for data sets. Processes register with DataMapper when data is written to files.
DataMapper_list_check	Obtains a data set list from the DataMapper and compares this with a reference list. Reports data sets which are late or very late.
DissectURL	Splits a TITAN URL into its component parts.
DsFileDist	Distributes files from one host to another via TCP/IP.
DsServerMgr	Manages the servers. Starts servers as requested.
GetPortNumber	Gets the port number for a given TITAN URL.
InputWatcher	Watches for data arriving in a directory or its subdirectories. Optionally runs a script when data arrives. Optionally runs a script when data is late. Optionally writes a _latest_data_info file to the data directory.
Janitor	Scans a directory tree, deleting files and directories and/or compressing files which reach a specified age.

LdataInformer	Contacts the DataMapper on behalf of another process. This is a proxy for the normal registration with the DataMapper.
LdataWatcher	Watches a directory in which data is arriving and a <code>_latest_data_info</code> file is being written. Optionally calls a script when data arrives. Optionally calls a script when data is late. Optionally adds an <code>_latest_data_info</code> FMQ. This is useful for directories that have data arriving from non-TITAN processes.
LdataMultWatcher	Same as LdataWatcher, except it watches multiple directories, and can call different scripts for each directory. This simplifies the setup and allows you to run fewer processes.
LdataWriter	Writes the <code>_latest_data_info</code> files to a directory. This is a proxy for the normal situation in which applications write these files directly. Useful to call from scripts.
LogFilter	Reads stdin, copies it to a log file in a dated directory. Adds time marks to the output. Used to filter error messages from applications and place them in a log directory.
PrintDataMap	Contacts the DataMapper on a host, reads the data table, prints the results to stdout in a number of formats.
PrintMdv	Reads an MDV file or requests a server to do so. Prints out the MDV data in a number of ways. Useful for checking the contents of MDV files and for debugging. Has a GIS option which prints the output in a manner conforming to GIS standards.
ResolveUrl	Resolves a TITAN URL, filling in the missing parts.
Scout	Traverses a data directory tree gathering summary information on the data sets. Communicates this information to the DataMapper.
SpdbQuery	Queries an SPDB data base, interprets the data if possible, prints out the interpreted data.
SyncDataMap	Synchronize a number of DataMapper applications, on remote hosts, to a single host. Has the effect of combining DataMapper information from a number of hosts into a single list.
auto_restart	Auto-restart perl script. Routinely contacts procmap, obtains the process status table, compares this against a reference process list. If a process is missing or late in registering, calls the kill script for that process, or <code>snuff_inst</code> . Then restarts the process using the start script.

<code>check_Janitor_files.pl</code>	Scans a directory tree for _Janitor parameter files. Reports the presence of _Janitor files and summaries the action which the Janitor would take based on those files with respect to file deletion.
<code>ipcrm_linux</code>	Removes shared memory segments on a LINUX system. Used for cleaning up when stopping TITAN.
<code>not_running</code>	Bourne-shell script. Checks to see if a process is NOT running. Returns 1 if running, 0 if not running.
<code>nuke_ipcs.linux</code>	Removes shared memory segments on a LINUX system. Used for cleaning up when stopping TITAN.
<code>print_procmap</code>	Contacts procmap on a host, retrieves the process status table, prints it out.
<code>procmap</code>	Process mapper. Keeps a state table for all process which register with it. Process register with procmap on a regular basis, normally once per minute.
<code>procmap_list_check</code>	Perl script. Contacts procmap, compares the returned process list with a reference process list. Prints information about process which are not running, or whether all processes are running.
<code>procmap_list_kill</code>	Perl script. Reads in a reference process list, starts all processes on the list by calling their start scripts.
<code>procmap_list_start</code>	Perl script. Reads in a reference process list, kill all processes on the list by calling snuff_inst or the kill scripts.
<code>rm_junk</code>	Recursively scans from a directory downwards, remove nuisance files. Remove core file, emacs ~ files, a.out, errlog files and TAGS files.
<code>running</code>	Bourne-shell script. Checks to see if a process IS running. Returns 0 if running, 1 if not running.
<code>snuff</code>	Bourne shell script. Kills a process given a string to match against a ps list.
<code>snuff_inst</code>	Bourne shell script. Kills a process based on the process name and instance.
<code>utime_decode</code>	Decodes a UNIX time into year/month/day hour:min:sec

DSR radar format - reading/writing

The DSR format is an NCAR format designed to generically handle beam-by-beam moments data, such as reflectivity, velocity, etc. The format handles any data field with a single value at each gate. Therefore it handles dual polarization data as well.

Application	Description
Bprp2Dsr	Reads beam-by-beam data via TCP/IP from RDAS radar acquisition system, which provides data in BPRP format. Converts the data into DSR format, writes to FMQ.
Dorade2Dsr	Reads NCAR Dorade format radar sweep files. Converts the data into DSR format, writes to FMQ.
Dsr2File	Reads beam-by-beam data from a DSR-format FMQ, writes the writes those messages to a binary file on disk. This application is used to archive the exact data from the FMQ so it can be played back again later using File2Dsr.
Dsr2Mdv	Reads DSR-format beam-by-beam radar data from FMQ, performs nearest-neighbor transformation of the data onto a regular grid and writes to MDV. This application requires prior knowledge of the radar scan strategy. A lookup table must be pre-computed using Dsr2MdvLookup. This application has been replaced by Dsr2Vol.
Dsr2MdvLookup	Computes nearest-neighbor lookup table for Dsr2Mdv, based on the radar scan strategy and output grid geometry.
Dsr2UF	Reads radar beam-by-beam data from a DSR-format FMQ and saves the data out in Universal Format files.
Dsr2Vol	Reads DSR-format beam-by-beam radar data from FMQ, performs interpolation or nearest-neighbor transformation of the data onto a regular grid and writes to MDV. This application does not require prior knowledge of the radar geometry or scan strategy.
DsrGrabber	DsrGrabber reads an input FMQ containing radar data, and writes out summary data in various ASCII formats.
edge2dsr	Reads radar beam-by-beam data from an EEC Edge system. Reformats the data into DSR and writes it to an FMQ.
File2Dsr	Reads DSR messages from a binary file previously saved for this

	purpose by Dsr2File.
GemScan2Dsr	Reads beam-by-beam radar data from a Gematronik scan file, reformats into DSR and writes to an FMQ.
GemVolXml2Dsr	Reads beam-by-beam radar data from a Gematronik XML volume scan file, reformats into DSR and writes to an FMQ.
Hiq2Dsr	Reads ABP data, via UDP, from HIQ computer running Windows, computes moments and writes to DSR FMQ.
HiqDos2Dsr	Reads ABP data, via UDP, from HIQ computer running DOS, computes moments and writes to DSR FMQ.
Iq2Dsr	Reads time series data from an FMQ or files, computes moments, writes output to DSR FMQ.
Mdv2Dsr	Reads beam-by-beam radar data from an MDV file which was written out in polar radar space, converts the data into DSR and writes to an FMQ.
NcRadar2Dsr	Reads radar beam-by-beam data from NetCDF sweep files. Reformats the data into DSR and writes it to an FMQ.
NcRadarSplit	Splits NetCDF sweep files containing radar volumes into separate PPI files.
NoiseFilter	Reads a DSR FMQ, filters the data based on thresholding a specified field, and writes the filtered data out to a DSR FMQ.
nexrad2dsr	Reads radar beam-by-beam data from NEXRAD sweep files. Reformats the data into DSR and writes it to an FMQ.
Rapic2Dsr	Reads radar beam-by-beam data from a RAPIC volume file, reformats it into DSR format and writes it to an FMQ.
Rdas2Dsr	Reads radar beam-by-beam data via TCP/IP from an RDAS2000 radar data acquisition system. Reformats the data into DSR and writes it to an FMQ.
Sigmat2Dsr	Reads radar beam-by-beam data from a SIGMET volume file. Reformats the data into DSR and writes it to an FMQ.
SigmatUDP2Dsr	Reads radar beam-by-beam data from a SIGMET RVP8 or IRIS UDP stream. Reformats the data into DSR and writes it to an FMQ.
Test2Dsr	Reads an MDV polar radar volume file from disk, and uses the data to

	simulate a DSR data stream. Writes the DSR data to an FMQ.
Uf2Dsr	Reads radar beam-by-beam data from a Universal Format file. Reformats the data into DSR and writes it to an FMQ.

The following table lists the applications which handle DSR format data.

ar moments data, i.e. reflectivity, velocity, etc.

DSR radar format - monitoring and relaying

Radmon is a useful application for monitoring radar data in an FMQ.

There are circumstances in which you may wish to send DSR radar data across a wide-area network. Where bandwidth is an issue, it makes sense to buffer up a number of beams and take advantage of data compression in order to reduce the bandwidth requirements. Fmq2MultFmq and MultFmq2Fmq work as a pair to perform this type of function. Fmq2MultFmq would be set up to write the buffered data to an FMQ on a remote host. MultFmq2Fmq would then be used to retrieve the data.

Application	Description
FmqMon	FmqMon is similar to RadMon except that it does not have knowledge of the internal format, it can only monitor at the message level. However, it is useful for performance monitoring of FMQs communicating over a wide-area network.
Fmq2Fmq	Reads messages from an input FMQ, and writes the messages to an output FMQ. Generally used for relaying the messages to a remote host, in which case this application contacts the DsFmqServer on the remote host and writes the messages to the server via a TCP/IP socket.
Fmq2MultFmq	Reads messages from an input FMQ, buffers the messages up into large messages and writes the buffered messages to an output FMQ.
MultFmq2Fmq	Reads messages from a buffered input FMQ, splits the buffer into its original messages and writes the individual messages to an output FMQ.

Radar applications, time series, dual polarization

Radmon is a useful application for monitoring radar data in an FMQ.

There are circumstances in which you may wish to send DSR radar data across a wide-area network. Where bandwidth is an issue, it makes sense to buffer up a number of beams and take advantage of data compression in order to reduce the bandwidth requirements. Fmq2MultFmq and MultFmq2Fmq work as a pair to perform this type of function. Fmq2MultFmq would be set up to write the buffered data to an FMQ on a remote host. MultFmq2Fmq would then be used to

retrieve the data.

Application	Description
RadMon	Monitors a DSR radar data FMQ. Prints out status in a variety of forms suitable for real-time monitoring or debugging.
suncal	Prints a list of sun locations, suitable for performing a sun calibration to position a radar antenna.
SunCal	Analyzes time series data of a sun scan.
CalCompute	Computes radar calibration from previously-written calibration file.
Cov2Mom	Cov2Mom reads covariances in an input DsRadar FMQ, computes moments and writes them out to DsRadar queue.
IntfRemove	Reads DSR messages from an input FMQ, searches for interference signals (based on a histogram technique), removes the interference and writes out the filtered data also to a DSR FMQ.
Iq2Dsr	Reads time series data from an FMQ or files, computes moments, writes output to DSR FMQ.
MdvPartRain	Reads a dual-polarization polar-radar MDV file, computes Particle ID (PID) and rain rates, writes the results to an MDV file.
PartRain	Reads a dual-polarization DSR FMQ, computes Particle ID (PID) and rain rates, writes the result to an output DSR FMQ.
RadarCal	Analyzes radar calibration data, writes results in ASCII file.
RadarConst	Computes radar constant, based on parameters provided
RadarMdvCompare	Compares MDV data from 2 radars, and computed biases in the overlap region between the radars. Intended to help with calibration consistency between radars.
SpectraPrint	Reads time series data from FMQ or files, reformats and writes into SPDB data base. Intended for use with SpectraPlot (See Displays).
TsCalAuto	Controls signal generator using SCPI commands, to inject known powers for calibration. Reads time series data, computes power values. Computes calibration, writes calibration XML file.
TsFile2Fmq	Reads time series data from files, writes to FMQ.

TsFmq2Tcp	This is a time series server. Reads time series data from an FMQ, sends to clients which have connected via a socket.
TsPrint	Reads time series data from an FMQ or files, computes summary data and prints to stdout.
TsSmartSave	Reads time series data from an FMQ, writes files. So-called ‘smart’ since it can decide which data to save and where to save it, based on scan strategy.
TsTcp2Fmq	Reads time series data from a time series server via a socket, writes the data to a time series FMQ.
UfRadarPrint	Prints data in UF radar files.
VertCompute	Reads vertically-pointing mode time series or moments data data from files or FMQ, computes ZDR bias.

Data ingest applications

The following applications read data from systems external to TITAN and convert the data into a format which TITAN can handle.

Application	Description
AcPosnAscii2Spdb	Read aircraft data in selected ASCII format, save to SPDB.
BomAws2Spdb	Read Australian BOM AWS surface station data, save in SPDB.
EsdAcIngest	Read ASCII aircraft data from a serial port and convert into SPDB. Input formats accepted are (a) ESD comma-delimited format, (b) WMI format and (c) UND format.
GenPt2Spdb	Convert ASCII generic point data to SPDB.
Gini2Mdv	Convert satellite data in GINI format into MDV. Input data is in un-calibrated counts, therefore a lookup table is required for interpretation of the data.
Grib2Mdv	Convert gridded model data in GRIB format into MDV format. Handles RUC, AVN, WAFS.
GtsSockIngest	Read GTS data from a TCP/IP socket, save out into files.

Ltg2Spdb	Read in lightning data in ASCII files and convert to SPDB.
Metar2Spdb	Read METAR data in ASCII format, convert to SPDB.
Nids2Mdv	Read NIDS NEXRAD Level 3 data in polar space, write out to MDV. Supports polar and Cartesian grid output.
NidsRadial2Mdv	Read NIDS NEXRAD Level 3 data in polar space, convert to Cartesian grid, write out to MDV.
NidsVad2Spdb	Reads VAD data from NIDS stream, writes out to SPDB.
npvtLtg2Spdb	Reads lightning data from a NOAA-port, reformats and writes to SPDB.
Plain2Mdv	Reads in a plain binary array, converts it to MDV format.
RainGauge	Reads rain gauge data in as ASCII, converts to SPDB.
Rhi2Spdb	Reads MDV RHI files, determines the time and azimuth angles, and stores the info in SPDB.
SatOrbit2AcPosn	Computes the orbit of a satellite, writes the data out as ac_posn data to SPDB.
SerialIngest	Reads ASCII data from an RS232 serial port or TCP/IP socket, assembles data into buffers and writes out the buffers to a plain ASCII file at regular intervals.
SigAirMet2Spdb	Reads SIGMET and AIRMET (Aviation Significant Met) data from ASCII files, writes to SPDB.
sndgIngest	Reads CLASS sounding format data, reformats and writes to SPDB.
Socket2File	Reads a selected number of bytes from a socket and dumps the result to a file.
Taf2Spdb	Reads TAF (Aviation Terminal Area Forecast) data from an ASCII file, writes to SPDB.
WafsTiled2Mdv	Reads WAFS data from tiles, which are part of the global domain. Combines the data from the tiles, writes out a merged MDV data set.
WsiNidsIngest	Reads NIDS data from a TCP/IP socket on a WSI server. Identifies products in the data stream, writes these out as individual files.
bdry2spdb	Converts a boundary (e.g. gust front) in ASCII format to SPDB.

wmi_ac_ingest	Reads aircraft position data in WMI format, reformats and writes to SPDB. Has been superseded by EsdAcIngest.
wsim2mdv	Reads WSI Radar Mosaic, converts into MDV.
WorldTerrain	Reads WSI Radar Mosaic, converts into MDV.

Data conversion applications

The following applications convert data from one format to another within the TITAN system.

Application	Description
LtgSpdb2Mdv	Read in lightning data in SPDB data, compute count and rate on a grid and save out as MDV.
SatOrbit2AcPosn	Computes the orbit of a satellite, writes the data out as ac_posn data to SPDB.
ltgSpdb2GenPt	Reads lightning data from SPDB data base in normal format, converts it into Generic Point format.
TitanVectors2Mdv	Reads storm motion vectors from TITAN data files, writes an MDV file with gridded U/V components of the smoothed titan vectors.
Tstorms2Xml	Reads TITAN storm and track files, converts selected track properties to XML format. Output is either to ASCII files or SPDB. Output SPDB data is used by Tstorms2Symprod.
Tstorms2GenPoly	Reads TITAN storm and track files, stores the storm shapes as a generic polyline data type in SPDB.
Tstorms2Spdb	Reads TITAN storm and track files, converts selected track properties to SPDB format. Output SPDB data is used by Tstorms2Symprod.

Data servers

The following applications are the data servers for TITAN. This table is repeated from the section on the TITAN data system - you can find more information on the servers in that section.

Server name	Default	Remarks
--------------------	----------------	----------------

	port	
DsServerMgr	5435	Manages the servers. Starts servers as requested.
ChillSdbServer	2111	Reads Dsr format moments from a DsRadar FMQ, reformats into CHILL SDB moments format. Generally used to send data to vchilld.
DsMdvServer	5440	Reads/writes MDV data to/from remote hosts
DsSpdbServer	5441	SPDB data server
DsFmqServer	5443	FMQ data server
DsFCopyServer	5445	File copy server - used by DsFileDist to distribute files
DsLdataServer	2111	Latest data info server. Used by applications which need to monitor a remote host for the arrival of data.
DsTitanServer	5446	TITAN data server
Ltg2Symprod	5450	Lightning data - convert to Symprod
AcTrack2Symprod	5451	Aircraft track data - convert to Symprod
Acars2Symprod	5467	ACARS aircraft data - convert to Symprod
Bdry2Symprod	5452	Boundary (gust front) data - convert to Symprod
Metar2Symprod	5456	Metar data - convert to Symprod
Pirep2Symprod	5457	PIREP and AIREP data - convert to Symprod
Tstorms2Symprod	5460	TITAN storm and track data - convert to Symprod
WxHazards2Symprod	5463	Weather hazards - convert to Symprod
BasinGenPt2Symprod	5464	Hydro basin generic point data - convert to Symprod
GenPt2Symprod	5465	Generic point data - convert to Symprod
GenPtField2Symprod	5466	Generic point field data - convert to Symprod
HydroStation2Symprod	5468	Hydrology station data - convert to Symprod
SigAirMet2Symprod	5469	SIGMETs and AIRMETS - convert to Symprod
GenPoly2Symprod	5472	Generic polygon data - convert to Symprod

Rhi2Symprod	5473	RHI location data - convert to Symprod
--------------------	------	--

MDV utilities

The following applications allow the user to manipulate MDV files in various ways, perform conversions and derive fields.

Application	Description
Mdv2Ascii	Reads MDV files, converts the data to simple ASCII format.
Mdv2Plain	Reads MDV files, converts data to plain binary array.
MdvClump	Can be used in conjunction with MdvThresh to remove small clumps from data. All non-missing data are considered as valid clump points (which is why MdvThresh should be used to mark undesirable points as missing).
MdvCombine	Collects MDV fields from several input URLs and copies them to a single output directory.
MdvComp	Reads MDV data, computes the composite (maximum at any height) and writes the results to MDV.
MdvCompare	Reads two MDV fields, possibly from two different files, performs some statistics on the differences between the two fields over a sub-grid. The differences are computed as the first field specified minus the second.
MdvConvert	Converts MDV file data in various ways and writes out to MDV. For example, you can change the compression type, encoding type, grid properties or number of vertical levels.
MdvFillMissing	Runs in archive mode only. It can be used to replace bad or missing values in the data with other values.
MdvFromCircles	Write MDV files from simple ASCII files that specify circular regions to be filled in the MDV grid (which is assumed to be LAT_LON and is specified in the param file).
MdvGrad	Computes the gradient of MDV data in two dimensions. Writes result to MDV.
MdvMask	Uses a template to mask out unwanted regions and set the values

	therein to missing. In conjunction with a land/sea mask, this could be used to only allow vales over land.
MdvMedian	Computes the median value, at each grid point, for a time series of MDV files. Intended to compute median clutter values for a clutter map. ClutterRemove is then used to remove the clutter which does not exceed the median values.
MdvMerge2	Merges MDV files on two or more grids into a single grid. Intended to merge data from multiple radars onto a single mosaic. Works on both 2D and 3D data.
MdvSmooth	Runs an averaging window over MDV data. User can select Min., Mean or Max filtering.
MdvTComp	Performs a temporal composite process on Mdv data. It uses a time series of files to create a composite.
MdvTSmooth	Performs temporal smoothing on Mdv data. It uses a time series of files to smooth out current file.
Mdv2TextHdr	Reads MDV files, writes out small text files containing MDV meta-data.
MdvThresh	MdvThresh thresholds MDV data. Data which fail the threshold test are set to missing. Data which pass the threshold may be set to a constant value if desired, effectively creating a boolean pass/fail field.
mdv_be2le	Swaps big-endian MDV data to little-endian.
mdv_le2be	Swaps little-endian MDV data to big-endian.
UpdateMdvFieldNames	Changes the names of fields in MDV files.
UpdateMdvOrigin	Modifies the origin of the grid in MDV files.

Display applications

The following applications form the display system for TITAN. Apart from the main displays, CIDD and Rview/TimeHist, there are a number of small applications designed to help you create map overlays.

Application	Description
-------------	-------------

AsciiSelect	Print selectively, generally to an xterm, ASCII data from an SPDB data base. In real-time mode, this app slaves from the CIDD shared memory segment.
bearing2latlon	Simple application which converts bearing and azimuth to latitude and longitude. Useful for computing the lat/lon of points on a map given a straight-edge centered at a point.
CIDD	Cartesian Interactive Data Display. Advanced display application for TITAN. Shows MDV and SPDB data superimposed on background images and maps. Shows TITAN tracks from SPDB data base.
CIDD_titan	Version of CIDD which does not require the imlib2 library. Does not support generating images and saving them.
DualDopplerLobes	Compute the location of dual Doppler lobes for a pair of radars, and write out in TITAN/CIDD map format file.
map_atd2rap	Convert NCAR EOL type maps (Zebra) to TITAN/CIDD map file formats.
MapCircles	Create a TITAN/CIDD map overlay with circles centered at a number of points.
MapGrid	Create a TITAN/CIDD map overlay with a lat/lon grid.
MetarSelect	Print selectively, generally to an xterm, METAR data from an SPDB data base. In real-time mode, this app slaves from the CIDD shared memory segment.
metar_strip	Plot METAR data time history. In real-time mode, this app slaves from the CIDD shared memory segment.
OverlayCreate	Prepare a TITAN/CIDD map format file, with circles as specified in the param file.
Qmap2Cidd	Read a Queensland, Australia format map file, reformat to TITAN/CIDD map format.
RapMap2Chill	Convert a TITAN/CIDD map format file to a CHILL map format file.
Rview	Similar application to CIDD, but simpler and with fewer features. However, shows TITAN storm tracks from native files, with more functionality than CIDD in this regard. Works in a pair with TimeHist, communicating via shared memory.

RadMon	Radar monitor. Text-based display application which monitors the FMQ of a radar DSR input stream. Can monitor in a simple manner, just showing time, azimuth and elevation. Also can verbosely print out details of the input field headers and data.
RemoteUI2Fmq	Application which allows the user to drive CIDD remotely using commands sent via a command FMQ. Intended for use with scripts which must command CIDD to generate graphics for the web.
Shape2Map	Convert a GIS shape file to a TITAN/CIDD map file format.
SoundingText	Print text for selected sounding, generally in an xterm. In real-time mode, this app slaves from the CIDD shared memory segment.
SpectraPlot	Plot spectra, from time series, at location of CIDD click point. Follows CIDD shared memory for location. Time series are read from SPDB data base which is created by SpectraPrint. See radar/dual polarization apps.
TimeHist	Time history application for TITAN storm track data. Works in conjunction with Rview to graphically reveal as much information about the storm tracks as possible. Communicates with Rview via shared memory.
TitanSelect	Selects specified Titan track data from the database and prints it to stdout. It watches the Rview shmем segments for user activity and reacts when a new track is specified. Suitable for running in a terminal window.

Printing utilities

The following applications allow you to view the contents of some of the binary data files in the TITAN system. They are useful for obtaining a detailed view of data on the system, especially when you are trying to trouble-shoot problems.

Application	Description
LdataWatcher	In debug mode, print details of _latest_data_info as new data arrives.
PrintMdv	Print headers and/or data from an MDV file, either locally or from a server on remote host. Will output data in GIS format.
PrintDataMap	Print the current state of the DataMapper
print_procmap	Print the current state of procmap .

PrintTitanFiles	Print headers and contents of TITAN storm track files.
PrintTitanServer	Print headers and contents of TITAN files, via the DsTitanServer .
PrintGenPt1D	Print entries from a 1-D generic point SPDB data base.
PrintHydroStation	Print entries from a HydroStation SPDB data base.
PrintNids	Print contents of a NIDS data file.
RadMon	Print summary, or details, of a DSR moments FMQ.
SpdbQuery	Print selected contents of an SPDB data base, either locally or from a server on a remote host.
TsPrint	Print a summary, or details, of a time series data set.
UfRadarPrint	Print the contents of a Universal Format radar data file.
XSpdbQuery	Similar to SpdbQuery , except using an X interface instead of a command-line.
fmq_print	Print the state of any generic FMQ.
FmqMon	Monitor activity in any generic FMQ.
print_server_mgr	Print the current state of DsServerMgr .