2 Introduction

Version 03.10.00, 2010-05-11

StateDMI is a tool that can be used to process and format data for CDSS models, including the StateCU (consumptive use) and StateMod (surface water) models. The "DMI" corresponds to "Data Management Interface," which is a general term for a tool that translates data from one form to another. TSTool is a DMI utility for processing time series. StateDGI and StatePP are other CDSS DMI utilities, which process GIS data and generate input for the MODFLOW groundwater model. StateDMI's input data are read from the State of Colorado's HydroBase database, spatial data files (e.g., ESRI shapefiles), text files, and existing StateMod and StateCU data files. Output is written to StateCU, StateMod, and text formats. StateDMI can be considered the middle application in the modeling process:

1. StateView and the CDSS web site are general HydroBase data-viewing tools, for initial data evaluation.
2. StateDMI processes model data from HydroBase and other sources into model files (see also TSTool, which performs a similar function for time series data).
3. The StateCU and StateMod models and graphical user interfaces are the final end-user applications for modeling.

StateDMI uses a workflow command language (similar to TSTool) to describe how data should be processed. The command language approach has a number of benefits:

1. It allows control of whether a data processing step occurs (or not).
2. It allows control of the order of data processing steps.
3. It allows complicated data processing sequences to be broken into manageable steps, which allows evaluation of different combinations and facilitates troubleshooting.
4. It allows data processing procedures to be saved and rerun at a later time. Consequently, complicated data processing steps can be remembered.
5. It allows data processing to be automated. For example, rather than interactively executing the same steps each time data need to be processed, an effort can be made once to determine data processing steps and record the steps in command files. The same steps can then be rerun later with little effort.
6. It allows comments to be inserted in the data processing procedures. For example, data that are read from HydroBase can be edited using commands and comments can be inserted with the commands to explain the reason for the edits. Consequently, data processes are self-documenting.
7. It allows commands to be updated and reused for other situations. For example, a sequence of commands that is appropriate for one geographic region may also be appropriate for another region. An existing command file can be read, modified slightly, and rerun for the new situation.
8. It facilitates extending software features. For example, a new model file format or database can be implemented by adding new commands within the existing framework.
9. It allows tests for command workflows to be automated, simplifying software and process testing.

In spite of these benefits, command workflows can be somewhat intimidating. To address this issue, the StateDMI interface provides a framework that provides interactive editors for commands and performs checks on input and results. Documentation is also available for all commands.

The following chapters are available in this documentation:

Chapter 1 – Acknowledgements – recognizes contributors to the development and maintenance of the StateDMI software.

Chapter 2 - Introduction (this chapter) provides background information about StateDMI and the CDSS modeling framework and procedures.

Chapter 3 – Getting Started provides an overview of the StateDMI interface features.

Chapter 4 – Creating StateCU Data Set Files provides guidelines and examples of how StateDMI can be used to create StateCU data set files.

Chapter 5 – Creating StateMod Data Set Files provides guidelines and examples of how StateDMI can be used to create StateMod data set files.

Chapter 6 – Troubleshooting provides troubleshooting information.

Chapter 7 – Quality Control provides information about how StateDMI software and modeling processes can be quality controlled.

The **Command Reference** provides a complete command reference with commands listed in alphabetical order. **Chapter 4** and **Chapter 5** summarize the use of commands for each product. The **Command Reference** is by far the longest part of the documentation. The **Command Glossary** at the start of the **Command Reference** provides a list of parameters that are used in commands, which promotes standardization of parameters.

The **Installation and Configuration Appendix** provides information about installing and configuring StateDMI.

The **Release Notes Appendix** summarizes important software changes for each StateDMI version.

See also the *doc\Training* folder under the software installation, which includes slideshows and example files for self-paced training.

2.1 How to Use this Documentation

The documentation is organized into chapters that provide overview material, with extensive reference material at the end of the documentation. It is recommended that the documentation be used as follows:

1. New users should review the **Introduction** and **Getting Started** chapters to understand in general how StateDMI operates.
2. When processing StateCU or StateMod files, review the introductory pages of the corresponding chapters (**Chapter 4** for StateCU and **Chapter 5** for StateMod) to gain an appreciation of the data files that will need to be processed.
3. To produce files for a specific data component (e.g., diversion stations), refer to the section in the model chapter corresponding to the data component. Review the example(s) that are provided and utilize similar steps when creating new commands files. The documentation provides examples taken from actual data sets and, although not universally applicable, provides a good starting point for new work. Refer to command files and documentation available with downloadable data sets for the most current examples of production work.

4. To fully understand how to use a command, whether in a new command file or an existing command file that is being updated, refer to the **Command Reference** section at the end of the documentation.

2.2 CDSS Modeling Overview

For CDSS, a major focus has been to develop an integrated data-centered system that can create basin-wide data sets for planning purposes. The end result is basin models with hundreds or thousands of model nodes, with associated water rights, time series data, etc. StateDMI breaks up the data processing into sessions that focus on specific model data components that have corresponding data files. A command file controls the creation of each model file. Although the overall modeling process is complicated, StateDMI is organized to help facilitate creating an entire data set and individual model files. See also the TSTool documentation – TSTool is used to process time series data in CDSS.

The primary purpose of the StateCU model is to estimate irrigation water requirement, although it does also estimate non-irrigation requirements. Several input files need to be prepared to run the StateCU model. The number of files depends on the complexity of the analysis. The StateCU documentation describes the StateCU model files in detail.

The StateMod model is used to simulate surface water use considering the Prior Appropriation Doctrine (first in time, first in right). Its primary purpose is to evaluate the water demand and supply in order to allocate water. Whereas StateCU data sets focus primarily on historical data, StateMod data sets can have several variations in order model various water allocation conditions and issues. The StateMod documentation describes the StateMod model files in detail.

For CDSS modeling, the StateCU and StateMod models have some interdependency. For example, to estimate acreage, water rights data (consistent with StateMod) can be used to turn parcels off if water rights did not exist. Similarly, StateMod depends on the demand data produced by StateCU. Typically, full StateCU data sets are prepared before StateMod data sets; however, as shown in the example above, there is a need to produce some StateMod files when creating a StateCU data set. StateDMI supports this by providing StateMod commands for products needed by StateCU. Once data sets for both models have been created for a basin, it becomes easier to share model files and update them over time.

StateMod data sets are typically created for historical conditions, calculated demands (using full supply demands), and baseline (the current system) cases. If historical or simulated diversions are available, they can be provided to StateCU to evaluate a water supply limited condition. See the next section for more information on various data sets types.

Given that StateCU and StateMod have numerous input files and a variety of run options, it can be difficult to understand and maintain data sets. StateDMI helps streamline data processing so that data flow is clearer.

When modeling, some efficiency can be gained by selecting key stream gages (those with a significant period of record) and determining for each structure type (diversion, reservoir, instream flows, and wells) the key and non-key structures. Key structures are modeled explicitly within StateCU and StateMod while non-key structures may be aggregated. The use of aggregation is discussed where appropriate in this documentation and is addressed in the model data set documentation.

2.3 Data Set Folder and File Conventions

The conventions used for StateCU and StateMod data set directories and files have changed over time, in particular as new modeling challenges have been faced (e.g., groundwater, augmentation plans). Older conventions are not discussed because CDSS data sets have generally been updated to current standards. If necessary, refer to the model data set documentation for older data sets.

CDSS model data sets are developed by the State and contractors and are provided on the CDSS web site. The data sets typically only contain input files in zip files and are named with an abbreviation of the basin and the year of release (e.g., *cm2005* for the Colorado basin data set released in 2005). Note that the ending year of the model data is often less than the year of the release. Output files may not be made available due to the size of the files; consequently, users will need to rerun the models to produce output and/or refer to the data set documentation. StateCU and StateMod data sets are typically provided separately and StateCU data sets are typically released earlier than StateMod data sets.

Folders under the main data set folder are described in the following table (adapted from "Recommended Data Structure", Ray Bennett, September 19, 2005). These conventions may change – see model data set documentation and files for conventions used with specific data sets. Folders are listed alphabetically in the following table; however, the order of processing is indicated by StateDMI menus and is described in model data set documentation and command files. Guidelines fir data sets are as follows:

- **Top-level Data Set Folder**. The top-level data set folder (e.g., *cm2005*) will include all data and results for the model data sets. An exception is GIS files, which may be located in a shared location like the *\cdss\gis* folder, allowing multiple data sets to share GIS files, which can be large. However, if possible, it is recommended that GIS files are included with a data set to allow for stand-alone data sets.
- **Relative Paths**. The "flat" organization of data set folders facilitates the use of relative paths. Model response files and command files should utilize relative paths when referring to folders (e.g., *..\Diversions\cm2005.dds*). This facilitates transport of data sets from one location/computer to another.
- **Final Model Folder**. The final model folder (e.g., *StateMod* for the StateMod model), will contain:
    1. input files produced by data processing,
    2. miscellaneous files that do not require processing (e.g., response and control files)
    3. output files from the model run
- **Folder Variations.** Folders in addition to those described in the following table may be used to simplify maintenance and use. For example, *Historic*, *Calculated*, and *Baseline* folders may be used under the *StateMod* folder to separate main model variations. Additional data folders for processing may be included if they clarify data management and processing.
- **Supporting Files**. Miscellaneous support files should be stored in folders with related data. For example, historical reservoir end of month time series files (in addition to data that will be queried from HydroBase) should be stored in the *Reservoirs* folder. If necessary, use a sub-folder to clarify data management.
- **Log Files**. The `StartLog()` command can be used as the first command in a command file to record processing that is performed. The log file can facilitate troubleshooting and serve as a useful artifact if a data set needs to be reviewed at a later date.
- **Quality Control**. The complexity of modeling and the decisions that are made based on the results require that quality control measures are implemented. Data checks can be performed using the `Check*()` commands. See also the **Quality Control** chapter of this documentation.

Performing quality control activities throughout modeling will help to minimize uncertainty about the validity of the model results.

- **Comments**. Hand-edited data files and command files should include comments of the top indicating the source and date for data. Comments should be included throughout command files to describe processing.

CDSS Data Set Folder Conventions

Folder	Primary Application (1)	Description
C:\CDSS\data\Basin		Main folder where basin includes data set release date (e.g., cm2005).
		Consumptive Use Application
.\ClimateCU	CU	Climate stations; temperature, precipitation, and frost time series associated with StateCU.
.\Crops	CU	Crop characteristics and coefficients; crop pattern and irrigation practice time series.
.\DelayCU	CU	Delay tables and assignment for StateCU limited supply analysis.
.\DocCU	CU	Documentation associated with a consumptive use application.
.\LocationCU	CU	CU locations and support list files.
.\StateCU	**CU**	**StateCU model files (all input and output for a consumptive use application).**
		Surface Water Application
.\ClimateSW	SW	Precipitation and evaporation time series associated with StateMod.
.\DelaySW	SW	Delay tables (monthly and daily) associated with StateMod.
.\DocSW	SW	Documentation associated with the surface water application.
.\Diversions	CU, SW, GW	Diversion stations and rights, historical and demand time series (monthly and daily), surface water aggregate, system, and multi-structure lists.
.\Instream	SW	Instream flow stations and rights, demand time series (average monthly, monthly, and daily).
.\Network	SW	StateMod network, generalized XML network.
.\Reservoirs	CU, SW	Reservoir stations and rights, end of month content and target time series.
.\StateMod	**SW**	**StateMod model files (all input and output files for a surface water application).**
.\StreamSW	SW	Stream files associated with StateMod (stream stations, historical time series, stream estimate coefficients, etc.).
.\Wells	CU, SW	Well stations and rights, historical pumping and demand time series, aggregation and system lists.
		Groundwater Application
.\Agg	GW	Aggregate polygons for StatePP.
.\DocGW	GW	Documentation associated with a groundwater application.
.\Edge	GW	Boundary conditions.

Folder	Primary Application (1)	Description
.\MIPumping	GW	M&I pumping.
.\ModFate	GW	Fate of surface water returns.
.\Modflow	GW	MODFLOW files (all input and output files for a ground water application).
.\PptRecharge	GW	Precipitation recharge associated with MODFLOW.
.\Prop	GW	Aquifer properties (K, SS, Sy, L).
.\RimInflows	GW	Rim inflows.
.\StateDGI	GW	GIS processing.
.\StatePP	GW	MODFLOW preprocessor.
.\StreamGW	GW	Stream files associated with the MODFLOW stream package.
.\StreamInflow	GW	Stream inflow to the groundwater model.
.\Survey	GW	Stream survey data.
.\URF	GW	Unit response development.

(1) Primary Application: CU = consumptive use, SW = surface water allocation, GW = groundwater

Both StateCU and StateMod data sets include some files that are typically not automatically created. These files include the main response and control files and the StateMod operational right file. However, most other files can be created in an automated fashion. The processing of data files typically occurs in a sequential fashion. Although modelers may have different approaches, StateDMI menus and documentation are generally organized according to data component/product dependency. For example, if one file depends on concepts or data from another file, then the dependent file is listed after the independent file in menus and procedures. In this way, the creation of a file avoids “forward referencing” another file that has not yet been created. However, some circular dependencies do occur in data preparation and are discussed with examples.

Although StateDMI’s interface is organized based on a logical creation order of the StateCU and StateMod files, it does not strictly impose rules on the order of creating files. StateDMI does encourage the use of standard StateCU and StateMod file extensions, as described in each model’s documentation. It does so by displaying the standard extensions in file choosers, although in most cases the user can override with any file extension.

The above information describes the general folder structure for a data set. The guidelines for naming the main data set folders are described below. Standard names for basin data set directories have been adopted to promote consistency and simplify data review. This naming convention reflects the following aspects of a data set:

- basin name, typically as an abbreviation (e.g., “rg” for Rio Grande)
- scope or scenario for the data modeled (e.g., whether a fraction or 100% of the consumptive use is modeled)
- year that the data set was created (may not agree with the last year included in the model)

The naming convention has changed over time and therefore legacy data set names do not agree with current conventions. For example, early data sets modeled approximately 75% of the consumptive demand. The next iteration of data modeled 100% of the consumptive demand, using aggregate stations where necessary, and these data sets were designated with a “T”. Current conventions are to include all effects by default and not use any special indicator like “T”. Therefore, the current naming convention focuses on the year that the data set was prepared and it is assumed that the data set takes advantage of all modeling capabilities. Short names are used because of an 8.3 character file name length limitation in StateMod, although this limitation may be removed in the future.

The following table lists examples of standard data set names, based on currently available data sets:

Standard Names for Baseline Data Sets

Basin	Data Set Name (1)
Arkansas	No data sets have been produced (ar?)
Upper Colorado Main Stem	cm
Gunnison	gm
Rio Grande	rg
San Juan/San Miguel/Dolores	sj
South Platte	sp
White	wm
Yampa	ym

The data set name recommendations have evolved over time and should be evaluated for each data set. For example, to facilitate future updates (e.g., extending data sets by additional years of data), it may be useful to NOT include the year in individual file names, using the year only for the main directory. However, this practice may lead to confusion when comparing data files from different versions of data sets because the year will not be included in the name. Conventions for each CDSS modeling effort should be evaluated and discussed with State of Colorado project managers.

To generate a calibrated StateMod model includes developing three inter-related data sets (see **Section 2.5** below for more information):

1. historical (also referred to as historic)
2. calculated
3. baseline

Example StateMod File Base Names

Model Run (StateMod Response File)	Key Properties of Data Set
cm2005H.rsp	Historical data set with 100% consumptive use included. Demands are generally the historical diversions. Reservoir targets are generally the historical end of month contents. Because historical files are often shared with other data set variants, the H may be omitted.
cm2005C.rsp	Calculated data set with 100% consumptive use included. Demands are calculated to equal the estimated headgate requirement (e.g., maximum of StateCU irrigation water requirement divided by average monthly efficiency AND historical diversions). Reservoir targets are generally forecasted.
cm2005B.rsp	Baseline data set with 100% consumptive use included. Demands are the same as the calculated data set; however, municipal, industrial, and trans-basin demands are set to a present or future value and facilities constructed during the study period are estimated to be on-line for the entire simulation.

Many of the files used in the historic, calculated and baseline data are the same. It is common for all the data to be the same except for the diversion demands and reservoir targets files. Refer to model data set documentation for detailed information about variations in data sets.

2.4 Standard Procedures for Creating StateCU and StateMod Data Sets

The previous sections described standard conventions for organizing data sets, including naming directories and files within data sets. **Chapter 4 – Creating StateCU Data Set Files** and **Chapter 5 – Creating StateMod Data Set Files** describe how to create each of the files necessary for each model. The recommended standard procedure for creating model files for each data type is to follow the steps in these chapters, illustrated by working examples from actual data sets.

The steps described in **Chapters 4** and **5** provide general guidelines related to data analysis and formatting. The following sections provide additional information related to variations in StateMod data sets. These variations should be considered when determining the level of modeling to be performed for a basin.

2.5 Variations in StateMod Data Sets

Chapters 4 and **5** discuss how to create all model files. However, some files (e.g., calculated demands) are used only in the calculated and baseline data sets. The following sections describe the differences between data sets.

2.5.1 Creating a Historical Data Set

A historical data set is used to calibrate the model and match historical conditions. Historical time series (e.g., diversions, well pumping) are used for demands. Differences between simulated results and the historical time series are minimized by adjusting return flow patterns, stream estimate proration factors, and other data. See the StateMod documentation for more information about historical data sets.

2.5.2 Creating a Calculated Data Set

A "calculated" data set is one that uses estimated demands, rather than simply using historical data (e.g., diversion time series and historical reservoir levels). To produce a calculated data set, revise the following files from those used in the historic data simulation:

- The calculated control file (**C.ctl*) is the same as the historical control file (**H.ctl*) except header cards are revised to indicate it is a calculated data set.
- The calculated diversion demand file (**C.ddm*) is similar to the historical diversion demand file (**H.ddm*) except agricultural demands equal the estimated diversion headgate requirement for full supply rather than historical diversions.
- The calculated well demand file (**C.wem*) is similar to the historical well demand file (**H.wem*) except agricultural demands equal the estimated well pumping requirement (full supply) rather than historical pumping.
- The calculated reservoir target file (**C.tar*) is similar to the historical reservoir target file (**H.tar*) except reservoir targets are typically set to forecasted values. For example, individual time series files stored in the supporting files directory may be combined into the complete file.

2.5.3 Creating a Baseline Data Set

A baseline data set represents current or future conditions, allowing an evaluation of the system for "what if?" scenarios. To create a baseline data set, revise the following files from those used in the calculated data simulation:

- The baseline control file (**B.ctl*) is the same as the calculated control file (**C.ctl*) except header cards are revised to indicate it is a baseline data set.
- The baseline diversion demand file (**B.ddm*) is similar to the calculated diversion demand file (**C.ddm*) except municipal, industrial and trans-basin demands are revised to equal the present or estimated future demand. In addition, any diversions that may have been constructed during the study period will be estimated to be on-line for the entire study period. Demands are typically implemented by creating replacement time series files that are combined into the final model file.
- The baseline well demand file (**B.wem*) is similar to the calculated well demand file (**C.wem*) except municipal, industrial and trans-basin demands are revised to equal the present or estimated future demand. In addition, any wells that may have been constructed during the study period will be estimated to be on-line for the entire study period. Demands are typically implemented by creating replacement time series files that are combined into the final model file.
- The baseline reservoir target file (**B.tar*) is similar to the calculated reservoir target file (**C.tar*) except any reservoirs that may have been constructed during the study period will be estimated to be on-line for the entire study period. These reservoir targets are typically implemented by creating replacement files by hand.
- The baseline reservoir station file (**B.res*) is similar to the calculated reservoir station file (**C.res*) except any reservoirs that have been constructed during the study period may have a different initial content value. These reservoir station files are typically implemented by using data resets in the initial content.

2.5.4 Creating a Data Set with Aggregated Structures

In CDSS projects, the approach to modeling 100% of a basin's consumptive use (CU) has been to explicitly model key structures that include approximately 75% of the basins CU and aggregate the remaining CU into aggregated stations. The model data sets are reviewed and enhanced over time to improve the model's representation of actual conditions. The aggregation process is typically implemented as follows (see data set documentation for details for each basin):

1. Aggregated irrigation structures are identified in GIS software (e.g., the CDSS Toolbox software) from an irrigated acreage coverage as those not explicitly modeled.
2. Aggregated irrigation groups are defined based on location and cumulative aggregated acreage. Often aggregated groups are selected to coincide with a streamflow gage.
3. Aggregated reservoirs are defined based on non-explicitly modeled reservoir water rights. Often aggregated groups are selected to coincide with a streamflow gage.
4. Aggregated M&I demands are defined based on non-explicitly modeled M&I demands based on regional population data and per capita use estimates. Often aggregated groups are selected to coincide with a streamflow gage.
5. Aggregated water right classes are defined based on class size and typical call dates in a basin. These call dates are typically identified from an evaluation of historical call records and basin interviews.
6. Aggregated irrigation, reservoir and M&I structures are added to the network file (**.net*).
7. Aggregated irrigation structures, reservoirs and M&I uses are often located on the main stem in order to include their CU without developing new hydrology data on small tributaries. StateDMI commands recognize aggregate stations and process data accordingly.

In addition to diversion aggregate nodes, "systems" and "MultiStruct" nodes may be utilized in modeling. See the StateMod diversion stations description for more information.

StateCU and StateMod model files do not include information to describe collections. Consequently, StateDMI relies on commands like `SetDiversionAggregateFromList()` to supply information to be used during processing. Neglecting to provide this information will impact the results (e.g., diversion time series will contain smaller values because the aggregation is not occurring).

2.5.5 Creating a StateMod Data Set with Daily Data

The steps necessary to create a daily historical data set from a monthly data set is described in detail in the **Frequently Asked Questions** section of the StateMod documentation.

2.5.6 Creating a StateMod Data Set with Wells

The steps necessary to create a data set with wells are described in detail in the **Frequently Asked Questions** section of the StateMod documentation.

2.6 Commands and Processing Sequence

The StateDMI interface allows a list of commands to be created, which when processed result in the creation of model data files and other output products. Several commands are often needed to create a single model file, as shown in the following example:

```
#
# StateDMI commands to create the Rio Grande Climate Stations File
#
# Step 1 - read climate stations
#
# The following reads from a list file...
ReadClimateStationsFromList(ListFile="climate.lst",IDCol="1")
#
# Step 2 - set data manually
#
SetClimateStation(ID="newid",Latitude=100,Elevation=1999,Region1="ADAMS",
  Name="my station",IfNotFound=Add)
#
# Step 3 - fill climate station information
#
FillClimateStationsFromHydroBase(ID="*")
#
# Step 4 - write the climate stations file
#
WriteClimateStationsToStateCU(OutputFile="rgTW.CLI")
#
# Step 5 – check data
#
CheckClimateStations(ID="*")
WriteCheckFile(OutputFile="cli.commands.StateDMI.check.html")
```

The general sequence of commands when creating a model file is:

1. Read data from an existing source (e.g., a list file, the HydroBase database, or a model file) using `Read*()` commands. Delimited list files typically contain an identifier column, and data are then often read from HydroBase. List files can be created from the model network, StateView, etc.
2. If appropriate, set additional data (e.g., add information that was not present after the first item) using `Set*()` commands. Existing or new data may be added.
3. If appropriate, fill data (e.g., fill all latitude values that have not been previously specified) using `Fill*()` commands. Missing data can be filled but new data objects are not created.
4. If appropriate, further process data with commands that perform calculations (e.g., limit filled diversion time series to water rights that were in effect at the time). Various data products require commands of varying complexity.
5. Write output to model files, using `Write*()` commands.
6. Perform checks to ensure that data are suitable for modeling using `Check*()` commands.

The menus that list commands to process a specific file are generally listed in the above order, to emphasize the order that commands should be used. In some cases, additional commands will be shown because of additional processing that is required. Although StateDMI lists menus in the general order that they would be used, commands should be used in the order that is appropriate to accomplish a task. In particular, there are no restrictions on setting or filling values after a calculation has occurred.

StateDMI commands are free-format, using the syntax:

```
CommandName(Param1=Value1,Param2=”Value2”,...)
```

The command name corresponds to the command menus and each command is documented in the **Command Reference** at the end of this manual. Parameters can be listed in any order, separated by commas. In many cases, parameters have default values and do not need to be specified. Parameter values that include white space or commas should be enclosed in double quotes. The StateDMI GUI command editor dialogs help edit all commands.

StateCU and StateMod files each typically correspond to lists of *objects*. For example, StateMod data sets include a list of diversion stations (corresponding to the *.dds* file). StateCU has a list of consumptive use locations (corresponding to the *.str* file). Relationships between data objects occur through shared data fields (e.g., station identifiers). For example, diversion historical time series use the diversion station identifier.

StateDMI maintains lists of these objects in memory and manipulates the objects as commands are processed. For example, a list of diversion stations can be read from a StateMod diversion station file (*dds*). Additional diversion stations may then be added to the list using "set" commands. Because it is possible that lists of objects may be created from multiple input sources, StateDMI usually allows lists of objects to be appended. For example, both StateMod diversion stations (*dds* file) and wells (*wes* file) may be considered as locations where irrigation water requirement should be estimated in StateCU. Such locations are collectively referred to as *CU Locations*. Sort commands are available for most data types to facilitate consistent output.

Because a model data set may contain many files, it is convenient to create the files in a logical order, separating the work of creating a data set by using multiple command files. The convention used in this documentation is to describe using one command file to create one model file. The model data set documentation describes the order and logic in creating each model file.