



MDCS

Mosaic Dataset Configuration Script

22 March 2013

Table of Contents

WHAT IS MDCS?	3
WHO SHOULD USE MDCS?	3
WHY CREATE MDCS?	3
INPUT TO MDCS	3
WHAT IS A CONFIGURATION FILE?	4
CREATING AND EDITING MDCS CONFIGURATION FILES.....	4
PREDEFINED COMMAND CODES AND THEIR CORRESPONDING GP FUNCTIONS RECOGNIZED BY MDCS	5
MDCS COMMAND LINE ARGUMENTS.....	5
HOW DOES MDCS WORK?	5
CALLING MDCS FROM BATCH FILES	6
CALLING MDCS FROM MODEL BUILDER.....	6
RETURN CODES	6
REPORTING	7
CONTENTS OF CONFIGURATION FILES FOR MDCS	7

What is MDCS?

MDCS is an acronym for Mosaic Dataset Configuration Script. It is a python script for creating a mosaic dataset, adding rasters, and setting all required/desired parameters. MDCS can be used to simplify the automation of creating mosaic dataset, by enabling the workflows to be parameterized and executed in a single step. The automation makes workflows more reproducible and efficient. The parameterization of the workflows also assists in encoding best practices for image management. MDCS is used in a number of the workflows defined in the **Image Management Guidebook** *[NOTE: a hyperlink will be added here when this Guidebook is published]* within the ArcGIS Help system. The parameterization of complete workflows also enables MDCS to be used to document the process of creating mosaic datasets and image services for QA/QC purposes.

MDCS is compatible with ArcGIS Desktop version 10.1.1.

Who should use MDCS?

MDCS is primarily developed for data managers that wish to automate the process of creating mosaic datasets. Developing workflows using MDCS requires knowledge of ArcGIS and a good understanding of how to create and use mosaic datasets. A certain level of expertise at editing XML files is also required. By using MDCS you should be able to define a workflow that can then be easily automated, and also replicate workflows to apply to different types of imagery. These workflows can then be run quickly when new imagery is obtained simply by a user (or automated process) running a script or model. Using MDCS, tools can be built with very simple user interfaces, so less experienced users can process, manage and serve imagery using ArcGIS.

Why create MDCS?

The traditional approach to building a mosaic dataset in ArcGIS requires using ArcGIS for Desktop to invoke multiple GP tools, one at a time, in order to achieve the desired results. This approach works well for creating ad-hoc mosaic datasets. However this manual approach can be prone to error and is time consuming in cases where multiple similar mosaic datasets need to be created. MDCS enables workflows for the creation of mosaic datasets to be parameterized and quickly applied to different types of data (e.g. elevation vs. high resolution satellite data vs. pre-processed orthophotos).

The many configuration settings in a mosaic dataset can also be difficult to understand, so even though many of the same GP tools are required for each workflow, some of the recommended parameters must typically change depending on the type of data and desired usage.

MDCS is designed to make the process of creating and configuring mosaic datasets more efficient and more repeatable. It enables separate GP tools to be called via simple commands, and the parameters for each of the commands to be defined in configuration files. In this way the most appropriate commands and parameters can be used commonly across many workflows, thus saving time by facilitating an environment for automating any workflow process.

One of the primary objectives for MDCS is to provide a single tool with a single input that can replace a large number of GP tools to make the process of creating and configuring mosaic datasets more automated, and more repeatable.

Input to MDCS

MDCS takes a configuration file (XML) as input which defines all the necessary processes/GP tools with arguments that are required to create, populate, and configure a mosaic dataset. Each GP tool function predefined in the MDCS library is bound to a unique command code. Using these predefined codes, it is possible to chain together (or omit) required operations to create new mosaic datasets, or to apply changes to an existing mosaic dataset.

What is a configuration file?

The only required input to MDCS at the command line is the XML template file for a given workflow. This configuration file contains all the necessary information to process a given mosaic dataset. Information within the configuration file includes the names of GP processes to be executed, with their corresponding arguments, mosaic dataset properties, and workspace information. The combination of one or more predefined GP command codes are used to define the entire workflow for creation of one mosaic dataset. In short, each configuration file can be thought of as a container defining all the required tools for a workflow which can be shared and passed on to anyone else to build a similar mosaic dataset.

Creating and editing MDCS configuration files

The configuration files are XML. In most cases the recommended method for creating such a configuration file is to copy an existing file for a similar workflow and edit it using a standard XML (or text) editor. In some repeatable workflows the majority of the parameters remain the same and only a few of the parameters need to be changed. It is therefore simple to create automated workflows that copy an existing template configuration file, modify some of the nodes, and then run the script.

A master configuration file (\scripts\master.xml) is provided. This contains all the available commands and parameters. A valid method of creating a configuration file is therefore to copy the master configuration file, review and edit all the nodes.

MDCS also comes with a Configuration File Editor tool (CFE). This can be used in the Model Builder environment to prompt users to change only a few necessary parameters in the configuration file. CFE can be configured as a GPTool to expose only the required parameters in a configuration file, e.g. those that must be unique such as path to input data, path and name for GDB, etc. The intent is for the majority of the mosaic dataset parameters to be predefined (based on “best practices” for each type of imagery), so the user need not be concerned with (or confused by) the full list of all possible parameters.

When opened, the CFE dialog will prompt for the required values and perform verification of the inputs. When run, CFE sets the defined parameters and saves a revised configuration file. In this way GPModels can be created to take a template configuration file, and set new parameters before generating a new mosaic dataset. Note – CFE will overwrite the input file, so a unique copy of an example configuration file should be created and saved BEFORE inputting that file to CFE.

One additional note of importance: the design of MDCS enables massive scalability to support management and sharing of large imagery collections. One key parameter in the configuration file is *MosaicDatasetType* which has one of three possible values: Source, Derived, or Referenced.

The definition and usage of these three different mosaic dataset types is discussed in detail in the Image Management Guidebook in the ArcGIS Help system. For simple mosaic datasets, use Source as the *MosaicDatasetType*. The user should review the Guidebook, and also examine the sample data and scripts provided with MDCS, to understand proper usage of these types. See “Contents of configuration files for MDCS” below for further details on the configuration file.

Predefined command codes and corresponding GP functions recognized by MDCS

The necessary GP tools required to complete a workflow can be chained together by a series of commands. Each GP tool is given a code that is used in the configuration file. Below is a table of the codes and the corresponding GP tool or command.

<i>Code</i>	<i>Tool</i>	<i>Link to Help</i>
<i>CM</i>	<i>Create Mosaic to create new mosaic dataset</i>	link
<i>CR</i>	<i>Create referenced mosaic dataset</i>	link
<i>AF</i>	<i>Add fields</i>	link
<i>AR</i>	<i>Add rasters (image data) into a mosaic dataset</i>	link
<i>BF</i>	<i>Build footprint</i>	link
<i>BS</i>	<i>Build seam lines</i>	link
<i>BP</i>	<i>Build Pyramids</i>	link
<i>CS</i>	<i>Calculate Statistics</i>	link
<i>CBMD</i>	<i>Color Balance mosaic dataset</i>	link
<i>BPS</i>	<i>Build Pyramids and Statistics</i>	link
<i>ERF</i>	<i>Edit Raster Function to add, replace, or remove a raster function template</i>	link
<i>DN</i>	<i>Define no data values</i>	link
<i>SP</i>	<i>Set properties for a mosaic dataset</i>	link
<i>IG</i>	<i>Import geometry (footprint, seamline, or boundary) into a mosaic dataset</i>	link
<i>IF</i>	<i>Import field values/calculate fields</i>	link
<i>BB</i>	<i>Build boundary</i>	link
<i>SS</i>	<i>Set statistics for a raster or mosaic dataset</i>	link
<i>CC</i>	<i>Computes the minimum and maximum cell sizes for the rasters in a mosaic dataset</i>	link
<i>BO</i>	<i>Build Overviews - Defines and generates overviews for a mosaic dataset</i>	link
<i>DO</i>	<i>Define Overviews - Defines the tiling schema and properties of the overviews</i>	link
<i>AI</i>	<i>Adds attribute index on the Mosaic Dataset</i>	link
<i>CV</i>	<i>Runs the "Calculate values" GP tool to set values in the attribute table</i>	link
<i>CP</i>	<i>Compact file Geodatabase</i>	link
<i>SY</i>	<i>Synchronize - Rebuilds or updates each raster item in the mosaic dataset.</i>	link

MDCS command line arguments

MDCS takes in few arguments at the command line to work with as shown below:

```
MDCS.py [Optional predefined codes] <configuration file>
```

IMPORTANT: If any command codes are specified as command line arguments, only those codes listed on the command line are executed. ALL command codes contained within the configuration file are ignored.

How does MDCS work?

MDCS reads in the configuration file specified at the command line input and, based on the GP command codes defined in the configuration file, goes through each one of them in the order they are defined. The order is

important, because in some cases changing the order will change the resulting output, including a possible failure of one or more processes (e.g. attempting to *AddRasters* before execution of *CreateMosaic*).

As noted above, GP codes defined within the configuration file can also be overridden by specifying the codes at the command prompt as input to MDCS. This enables you to run just parts of the complete script to verify the intermediate outputs.

Once MDCS verifies each command to be valid, each command gets processed sequentially. Corresponding information/arguments for the matching GP operation is read from the configuration file to run the desired GP tool. Any errors/warnings, if found, while each process is being run are recorded in the log-file in the \logs\ directory.

Calling MDCS from Batch files

MDCS can be called by using batch files with commands of the following form.

```
python.exe MDCS.py -i:ConfigFileName.xml
```

A configuration file is the only mandatory input, as shown above. The following list describes optional valid arguments accepted by MDCS at the command line.

- c: command code(s)** (as noted above, if any command codes are included on the command line, all command codes in the input configuration.xml file are ignored)
- m: Mosaic dataset path including GDB and MD name** [e.g. c:\WorldElevation.gdb\Portland]
- s: Source data paths.** (As inputs to command)
- l: Log file to write to disk** [path+file name]

In this way multiple calls to MDCS can be chained together for more complex workflows involving the creation of multiple mosaic datasets.

Calling MDCS from Model Builder

MDCS and all its relevant library modules can be imported as standard python objects using code in an ArcGIS python toolbox. Once MDCS gets properly imported without any errors, all classes/tools defined within the toolbox can make calls to MDCS as a program object to access its functions. MDCS can also be run as an external process with valid command line arguments if required. This is akin to running MDCS from batch files with correct arguments.

ArcGIS users who would like to leverage MDCS within Model Builder to help build workflows could create Tools within a python tool box to associate MDCS configuration file content with relevant Model Builder UI elements on screen for editing. One or more editing tools can be created to edit the same configuration file but at different places in the same file. This way, a tool can exist to edit Mosaic Dataset Properties in the configuration file while another could exist to edit process information.

MDCS is provided with a Configuration File Editor tool (CFE) to help edit configuration files interactively through Model Builder rather than resort to XML editors that could potentially introduce editing errors. The CFE tool exposes commonly used/edited entries in the configuration file to a Model Builder UI to allow a user to edit an existing configuration file template interactively and have the changes saved to a new configuration file.

Return Codes

All command operations or functions within MDCS system return either true or false to show the result of the last run operation. The client application can keep track of any operations that failed using the status codes returned. Further details on any errors/warning are available in the log files generated in the log folder where the standard GP

error codes and descriptions can be found. Based on these predefined and standard error codes, a user can refer to the standard ArcGIS desktop Help to find an acceptable solution to fix any errors found.

Reporting

MDCS includes a reporting system. For each execution of MDCS, a report file with a name of the form **ConfigFileNameYYYYMMDDT####.xml** is written to the \logs\ directory. The filename includes date and time, so a log file will never overwrite the result of a previous process.

Contents of configuration files for MDCS

As noted above, the recommended method for creating configuration files is to copy an existing file for a similar workflow and make edits appropriate for your new data. However, for developers and data managers seeking an in-depth understanding of the configuration files, this section will provide further detail.

The first step to creating a new configuration file for input to MDCS is to understand the master.xml file. This file is a template that contains the choices and defaults for each parameter. Below is a guide to understanding these parameters and choices.

Understanding the Master.xml template.

The master xml consists of several nodes and sub nodes. The main node name is *Application* which encompasses all the other sub nodes. The nodes correspond to the commands listed above. Whenever a command is required it is imperative to make sure that the corresponding node exists in the configuration file and is correctly populated. The list of major node names below is followed by detailed explanations of each.

- 1) Name
- 2) Command
- 3) Workspace
 - a) WorkspacePath
 - b) Geodatabase
 - c) MosaicDataset
 - d) AddRasters
 - e) DefaultProperties
 - f) Table
 - g) Functions
 - h) Processes

1) Name

This corresponds to the name of the project. Edit this parameter to reflect the name of the project you are going to use the configuration file for. This helps in logging the actions of MDCS.

2) Command

This node takes in the predefined command codes specified above. The command codes correspond to specific GP functions. A list of the command codes are listed above in [“Predefined command codes and their corresponding GP functions recognized by MDCS”](#)

3) Workspace

The workspace defines parameters related to creating, populating and setting properties of the mosaic dataset.

- a) **WorkspacePath** : This is the path to the folder where the GDB will be stored.

For example : c:\Image_Mgmt_Workflows\browseimagery\MD

- b) **Geodatabase:** This is the name of the geodatabase. Typically it is the name of the geodatabase in which you would like to create the mosaic dataset. For example : BllImage.gdb
- c) **MosaicDataset:** The mosaic dataset node contains properties required to create the mosaic dataset. Here you can specify the Mosaic Dataset Type, Name, SRS, Number of Bands and the Pixel Type. In the Master XML the nodes are filled with either a dummy name or valid choices. The *Mosaic Dataset Type* is a parameter that is required by MDCS, with three valid values: Source, Derived, or Referenced. For type=Referenced, a different GP tool (Create Reference Mosaic Dataset) must be used than for the other types. For type=Derived the dataset_ID field will not be updated, to preserve values entered for type=Source. Refer to the Image Management Guidebook in the ArcGIS Help system for further detailed discussion regarding these three types. More information regarding the other parameters can be found [here](#).
- d) **AddRaster :** The add raster node contains information regarding the data source that needs to be added to the mosaic dataset. Replace the values that are in these nodes to point to the dataset that needs to be added. More help on this topic can be found [here](#). **NOTE:** regarding the Raster Type, if the data will be loaded using the default type, "Raster Dataset" and all default settings, no further detail is required, but if the Raster Type or any properties must change, the user must create an *.art.xml file using ArcGIS Desktop, and import that to define the Raster Type. See help [here](#).
- e) **DefaultProperties:** Default properties control how the mosaic dataset can be used when it is being displayed or published. Edit the values appropriately; keep only the appropriate choices where multiple choices are defined.
For example : <resampling_type>**NEAREST;BILINEAR;CUBIC;MAJORITY**</resampling_type>, should contain only one value after editing, e.g. <resampling_type>**BILINEAR**</resampling_type>
For more help on this topic please refer to this [link](#).
- f) **Table:** This node defines the additional fields that need to be added to the mosaic dataset attribute table. In the master XML file you can include multiple fields. An example of how to that is given below. You will need to copy and paste additional nodes to define more than one field.

Add Single Field	Add Multiple Fields
<pre> <Fields> <Field> <name> SampleFld</name> <type>TEXT</type> <length>32</length> </Field> </Fields> </pre>	<pre> <Fields> <Field> <name>SampleFld</name> <type>TEXT</type> <length>32</length> </Field> <Field> <name>SampleFld2</name> <Type>SHORT</Type> <Length>4</Length> </Field> </Fields> </pre>

For help regarding supported field types look [here](#).

- g) **Functions:** Enter the path to the raster function template file. The raster functions allow you to define processing operations to the rasters in the mosaic dataset.

To know what functions are used by mosaic dataset go [here](#).

For information on how to create a Raster Function Template file click [here](#).

- h) **Processes:** The processes defined in this node correspond to the various commands specified using the command line or the commands node in the configuration file. The parameters are only accessed by MDCS if the command is specified. Only edit the parameters of the nodes of the commands that are specified.

For example: Commands specified are CS+BPS. The configuration nodes that are read in by MDCS are *CalculateStatistics* and *BuildPyramidsAndStatistics*. Other configuration nodes may be populated in the configuration file, but they will be ignored.

For help regarding the individual parameters specified in this section please refer to the table of command codes above.