



# **MDCS – Mosaic Dataset Configuration Script**

Version 5  
29 January 2013

# Table of Contents

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

## What is MDCS?

MDCS is an acronym for Mosaic Dataset Configuration Script and is the entry point to a collection of *Python classes/libraries that could be consumed by a Python client application to complete a given workflow* for creating a mosaic dataset, populating it with data, and setting all required/desired parameters. MDCS simplifies 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.

## Why create MDCS?

The traditional approach to building a mosaic dataset in ArcGIS requires invoking multiple GP tools one at a time in order to achieve the desired results. This approach works, however can be a time consuming process, and can be difficult to reproduce in cases where multiple similar mosaic datasets need to be created, or for 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 the parameters for each of the commands to be defined in a configuration files. The order of the commands to be executed is also defined by a list of unique command codes. 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.

## Input to MDCS

MDCS library takes an XML file (Configuration file) 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 code as mentioned earlier. Using these predefined codes, it is possible for a client/user to chain together (or omit) required operations to be carried out on an existing mosaic dataset or to create new mosaic datasets.

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.

## Predefined command codes and their corresponding GP functions recognized by MDCS

<i>CM</i>	<i>Create new mosaic dataset</i>
<i>CR</i>	<i>Create new referenced mosaic dataset</i>
<i>AF</i>	<i>Add fields</i>
<i>AR</i>	<i>Add rasters/data to a mosaic dataset</i>
<i>BF</i>	<i>Build footprint</i>
<i>BS</i>	<i>Build seam lines</i>
<i>BP</i>	<i>Build Pyramid</i>
<i>CS</i>	<i>Calculate Statistic</i>
<i>CBMD</i>	<i>Color Balance mosaic dataset</i>
<i>BPS</i>	<i>Build Pyramid and Statistic</i>
<i>ERF</i>	<i>Edit Raster Function</i>
<i>DN</i>	<i>Define no data values</i>
<i>SP</i>	<i>Set mosaic dataset properties</i>
<i>IG</i>	<i>Import mosaic dataset geometry</i>
<i>IF</i>	<i>Import field values/calculate fields</i>
<i>BB</i>	<i>Build boundary</i>
<i>SS</i>	<i>Set statistics for a raster or mosaic dataset</i>
<i>CC</i>	<i>Computes the minimum and maximum cell sizes for the rasters in a mosaic dataset</i>
<i>BO</i>	<i>Defines and generates overviews for a mosaic dataset</i>
<i>DO</i>	<i>Defines the tiling schema and properties of the pre-processed raster datasets</i>
<i>AI</i>	<i>Adds attribute index on the Mosaic Dataset</i>
<i>CV</i>	<i>Calculate values on the Mosaic Dataset</i>
<i>CP</i>	<i>Compact file Geodatabase</i>

## 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, ALL codes contained within the configuration file are ignored, and only those codes listed on the command line are executed.

## 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 the given mosaic dataset mentioned within the file which could take part in one or more workflows to complete a given task. 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 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 many cases the simplest method of 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 also 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 a model builder environment to prompt users to change only a few of the parameters in the configuration file. CFE can be configured as a GPTool to reference only the optional parameters in a configuration file. When opened the dialog will prompt for the values and perform the required verification. When run, CFE sets the defined parameters in the 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.

## 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 logged in the log-file in the `\logs\` directory.

## 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 run 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.

## 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
```

**Following is a list of valid arguments accepted by MDCS at the command line.**

- i: Input configuration file.**
- c: command code(s)**
- m: Mosaic dataset path including GDB and MD name [e.g. c:\WorldElevation.gdb\Portland]**
- s: Source data paths. (As inputs to command (AR))**
- 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 purpose. 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.