ArcGIS® 9.0

Geoprocessing Commands Quick Reference Guide

CONTRIBUTING WRITERS
Melanie Harlow, Ghislain Prince, Catherine Jones, Corey Tucker, Jeff Reinhart

Table of Contents

Introduction

This reference guide is designed to provide an easy and quick reference for those wishing to use the ESRI® command language at the ArcGIS® command line and for those writing scripts.

All commands are referred to as tools, scripts, or models and are maintained in toolsets within the ArcGIS toolboxes.

 A toolbox can contain tools, toolsets, and scripts and is organized according to the collection of geoprocessing commands it contains.

 A toolset can contain tools, toolsets, and scripts and is organized according to the geoprocessing commands it contains.

 A tool is a single geoprocessing command.

 A script is a set of instructions usually stored in a file and interpreted, or compiled, at run time.

 A model consists of one process or, more commonly, multiple processes strung together.

This guide describes the following toolboxes:

Analysis Toolbox
Cartography Toolbox
Conversion Toolbox
Coverage Toolbox
Data Management Toolbox
Geocoding Toolbox
Linear Referencing Toolbox
Spatial Analyst Toolbox
Spatial Statistics Toolbox
3D Analyst™ Toolbox

Each toolbox contains a list of the toolsets and tools as they are organized within ArcToolbox™.

The Index section at the end of this guide contains an alphabetical list of each tool, script, toolset, and toolbox.

All tools are available with the ArcInfo™ license or the extension they are associated with. However, many are available for use with ArcView® or ArcEditor™ (sometimes with limited functionality). Those available with ArcView and ArcEditor are denoted with a ❖, and those available with ArcEditor are denoted with a ◆.

Some tools, such as Clip, exist in multiple toolboxes. Therefore, an alias can be added as a suffix to the tool name when more than one toolbox is available. An example of an alias usage is clip_arc, where clip is the tool and arc is the suffix representing the Coverage Toolbox, or clip_analysis, where the suffix represents the Analysis Toolbox.

The alias list

Analysis Toolbox	_analysis
Cartography Toolbox	_cartography
Conversion Toolbox	_conversion
Coverage Toolbox	_arc
Data Management Toolbox	_management
Geocoding Toolbox	_geocoding
Linear Referencing Toolbox	_lr
Spatial Analyst Toolbox	_sa
Spatial Statistics Toolbox	_stat
3D Analyst Toolbox	_3d

The syntax of an example tool

```
Union_arc <in_cover> <union_cover> <out_cover> {fuzzy_tolerance}
  {JOIN | NO_JOIN}
```

Where

Union_arc is the tool; the following components are the parameters.

<> indicates required parameters.

{ } indicates optional parameters; these do not need to be included. One can be skipped using # if you need to apply only a portion of them.

The | indicates mutually exclusive arguments, and only one of the arguments in the list of options can be specified.

In some commands, there may be an ellipsis between two arguments, such as item1...item4. This indicates that you can give one or more (up to 4 in this example) names or values for that argument.

Example

```
Union_arc Treepolycov Newtreecov Finaltreecov # JOIN
```

Analysis Toolbox

A suite of geoprocessing tools used to solve spatial or statistical problems.

Extract Toolset

Contains tools used to manipulate data into manageable datasets containing only the desired features and attributes.

❖**Clip:** extracts those features from an input feature class that overlap with features from a clip feature class.

```
Clip <in_features> <clip_features> <out_feature_class> {cluster_tolerance}
```

Select: extracts selected features from an input feature class or layer and stores them in the output feature class.

```
Select <in_features> <out_feature_class> {where_clause}
```

Split: clips the input features and stores them in multiple output datasets.

```
Split <in_features> <split_features> <split_field> <out_workspace>
  {cluster_tolerance}
```

Table Select: extracts selected attributes from an input table or table view and stores them in an output table.

```
TableSelect <in_table> <out_table> {where_clause}
```

Overlay Toolset

Contains tools for topological integration of features based on symmetry.

Erase: copies input features falling outside the erase polygon feature boundaries to the output.

```
Erase <in_features> <erase_features> <out_feature_class> {cluster_tolerance}
```

Identity: intersects two feature classes. The output contains the input features as well as those overlapping features of the identity feature class.

```
Identity <in_features> <identity_features> <out_feature_class>
  {ALL | NO_FID | ONLY_FID} {cluster_tolerance}
  {NO_RELATIONSHIPS | KEEP_RELATIONSHIPS}
```

❖**Intersect:** creates an output feature class containing features that fall within the area common to both input datasets.

```
Intersect <features{Ranks};features{Ranks}...> <out_feature_class>
  {ALL | NO_FID | ONLY_FID} {cluster_tolerance} {INPUT | LINE | POINT}
```

Symmetrical Difference: creates an output feature class containing features or portions of features common only to one of the inputs.

```
SymDiff <in_features> <update_features> <out_feature_class>
  {ALL | NO_FID | ONLY_FID} {cluster_tolerance}
```

❖**Union:** creates an output feature class containing all features from both inputs.

```
Union <features {Ranks};features {Ranks}...> <out_feature_class>
  {ALL | NO_FID | ONLY_FID} {cluster_tolerance} {GAPS | NO_GAPS}
```

Update: updates the attributes and geometry of the input using the update feature class or layer they overlap.

```
Update <in_features> <update_features> <out_feature_class>
  {BORDERS | NO_BORDERS} {cluster_tolerance}
```

Proximity Toolset

Contains tools to determine spatial relationships among features, with respect to the distance relationships between features.

❖**Buffer:** creates buffer polygons to a specified distance around the input features.

```
Buffer <in_features> <out_feature_class> <buffer_distance_or_field>
  {FULL | LEFT | RIGHT} {ROUND | FLAT} {NONE | ALL | LIST}
  {dissolve_field;dissolve_field...}
```

❖**Multiple Ring Buffer:** creates a new feature class of buffer features using a set of buffer distances.

```
MultipleRingBuffer <input_features> <output_feature_class>
  <distances;distances...> {DEFAULT | CENTIMETERS | DECIMALDEGREES | FEET |
  INCHES | KILOMETERS | METERS | MILES | MILLIMETERS | NAUTICALMILES | POINTS |
  YARDS} {field_name} {ALL | NONE}
```

Near: computes the distance from each point in the input to the nearest feature in the near feature class or layer.

```
Near <in_features> <near_features> {search_radius} {NO_LOCATION | LOCATION}
  {NO_ANGLE | ANGLE}
```

Point Distance: computes the distance between each point in a feature class or layer to all points in a different feature class or layer.

```
PointDistance <in_features> <near_features> <out_table> {search_radius}
```

Statistics Toolset

Contains tools that perform standard statistical analysis on attribute data.

Frequency: calculates frequency statistics for field(s) in a table.

```
Frequency <in_table> <out_table> <frequency_fields;frequency_fields...>
  {summary_fields;summary_fields...}
```

Summary Statistics: calculates summary statistics for field(s) in a table.

```
Statistics <in_table> <out_table> <field{Statistic Type};field{Statistic
  Type}...> {case_field}
```

Cartography Toolbox

Contains tools designed to produce data for maps to meet specific cartographic standards.

Masking Toolset

Contains tools to construct masking polygons for use with variable depth masking.

Cul-de-Sac Masks: creates a feature class of polygon masks from a symbolized input line layer.

```
CuldeSacMasks <input_layer> <output_feature_class> <reference_scale>
  <spatial_reference> <margin> {ONLY_FID | NO_FID | ALL}
```

Feature Outline Masks: creates mask polygons at a specified distance and shape around the symbolized features in the input layer.

```
FeatureOutlineMasks <input_layers> <output_feature_class> <reference_scale>
  <spatial_reference> <margin> <CONVEX_HULL | BOX | EXACT_SIMPLIFIED | EXACT>
  <ALL_FEATURES | ONLY_PLACED> {ONLY_FID | NO_FID | ALL}
```

Intersecting Layers Masks: creates masking polygons at a specified shape and size at the intersections of symbolized input layers.

```
IntersectingLayersMasks <masking_layer> <masked_layer> <output_feature_class>
  <reference_scale> <spatial_reference> <margin> <CONVEX_HULL | BOX |
  EXACT_SIMPLIFIED | EXACT>  <ALL_FEATURES | ONLY_PLACED>
  {ONLY_FID | NO_FID | ALL}
```

Conversion Toolbox

Contains tools that are used to convert data into various formats.

From Raster Toolset

Contains tools to output raster datasets to other formats.

❖**Raster to ASCII:** converts a raster dataset to an ASCII file representing raster data.

```
RasterToASCII <in_raster> <out_ascii_file>
```

❖**Raster to Float:** converts a raster dataset into a file of binary floating point values representing raster data.

```
RasterToFloat <in_raster> <out_float_file>
```

❖**Raster to Point:** converts a raster dataset to a point feature dataset.

```
RasterToPoint <in_raster> <out_point_features> {raster_field}
```

❖**Raster to Polygon:** converts a raster dataset to a polygon feature dataset.

```
RasterToPolygon <in_raster> <out_polygon_features> {SIMPLIFY| NO_SIMPLIFY}
  {raster_field}
```

❖**Raster to Polyline:** converts a raster dataset to a polyline feature dataset.

```
RasterToPolyline <in_raster> <out_polyline_features> {ZERO | NODATA}
  {minimum_dangle_length} {SIMPLIFY| NO_SIMPLIFY} {raster_field}
```

To CAD Toolset

Contains tools to assist in the mining of CAD drawings for the purpose of populating new or existing geodatabase feature classes.

Add CAD Fields: adds fields to the input table by selecting from groups of CAD-specific fields, which have the appropriate name and type recognized by the Export CAD tool.

```
AddCADFields <input_table> <ADD_ENTITY_PROPERTIES | NO_ENTITY_PROPERTIES>
  {ADD_LAYER_PROPERTIES | NO_LAYER_PROPERTIES} {ADD_TEXT_PROPERTIES |
  NO_TEXT_PROPERTIES} {ADD_DOCUMENT_PROPERTIES | NO_DOCUMENT_PROPERTIES}
  {ADD_XDATA_PROPERTIES | NO_XDATA_PROPERTIES}
```

Create CAD XData: creates a table formatted to be recognized by the Export CAD tool as AutoCAD Extended Entity Data.

```
CreateCADXData <in_table> <fields;fields...> <RegApp> <ADE | TRADITIONAL>
```

Export to CAD: exports features from a feature class to one or more CAD drawings.

```
ExportCAD <in_features;in_features...> <DWG-R2000 | DGN-V8 | DWG-R14 | DXF-R14 |
  DXF-R2000 | DWG-R2004 | DXF-R2004> <Output_File> {USE_FILENAMES_IN_TABLES |
  IGNORE_FILENAMES_IN_TABLES} {OVERWRITE_EXISTING_FILES |
  APPEND_TO_EXISTING_FILES} {Seed_File}
```

Set CAD Alias: renames one or more existing field name aliases by matching columns from the input table with a list of CAD-specific fields of appropriate names, which are recognized by the Export CAD tool.

```
SetCADAlias <input_table> <field_info>
```

To Coverage Toolset

Contains tools to convert any supported feature class format to a coverage.

Feature Class to Coverage: creates a single coverage from one or more input feature classes or layers.

```
FeatureClassToCoverage <features{Type};features{Type}...> <out_cover>
  {cluster_tolerance} {DOUBLE | SINGLE}
```

To dBASE Toolset

Contains tools to convert tables into a dBASE® format.

❖**Table to dBASE:** converts INFO™, OLE DB, or geodatabase tables to dBASE tables.

```
TableToDBASE <input_tables;input_tables...> <output_folder>
```

To Geodatabase Toolset

Contains tools to convert any supported vector or raster data type to a geodatabase.

❖**Feature Class to Feature Class:** copies a feature class into a geodatabase or to a shapefile.

```
FeatureClassToFeatureclass <input_features> <output_location>
  <output_feature_class_name> {expression} {field_info} {SAME_AS_TEMPLATE |
  DISABLED | ENABLED} {SAME_AS_TEMPLATE | DISABLED | ENABLED}
  {configuration_keyword} {first_spatial_grid}
```

❖**Feature Class to Geodatabase (multiple):** copies one or more feature classes or layers to a geodatabase feature class.

```
FeatureclassToGeodatabase <input_features;input_features...>
  <output_geodatabase>
```

❖**Import CAD Annotation:** converts a collection of CAD annotation features into a geodatabase annotation feature class.

```
ImportCADAnnotations <input_features;input_features...>
  <output_feature_class> <reference_scale> {CLASSES_FROM_LEVELS |
  ONE_CLASS_ONLY} {NO_MATCH | MATCH_FIRST_INPUT} {NO_SYMBOL_REQUIRED |
  REQUIRE_SYMBOL} {STANDARD | FEATURE_LINKED} {linked_feature_class}
  {AUTO_CREATE | NO_AUTO_CREATE} {AUTO_UPDATE | NO_AUTO_UPDATE}
```

❖**Import Coverage Annotation:** imports coverage annotations into a geodatabase annotation feature class.

```
ImportCoverageAnnotations <input_features;input_features...>
  <output_feature_class> <reference_scale> {CLASSES_FROM_LEVELS |
  ONE_CLASS_ONLY} {NO_MATCH | MATCH_FIRST_INPUT} {NO_SYMBOL_REQUIRED |
  REQUIRE_SYMBOL} {STANDARD | FEATURE_LINKED} {linked_feature_class}
  {AUTO_CREATE | NO_AUTO_CREATE} {AUTO_UPDATE | NO_AUTO_UPDATE}
```

❖**Import from CAD:** imports from one or more CAD files to a geodatabase.

```
ImportCAD <input_files;input_files...> <out_personal_geodatabase>
  {spatial_reference} {DO_NOT_EXPLODE_COMPLEX | EXPLODE_COMPLEX}
```


❖**Raster to Geodatabase (multiple):** loads multiple raster datasets into a geodatabase or raster catalog.

```
RasterToGeodatabase <input_rasters;input_rasters...> <output_geodatabase>
  {configuration_keyword}
```


❖**Table to Geodatabase (multiple):** converts dBASE, INFO, or OLE DB tables to geodatabase tables and copies tables from one geodatabase to another.

```
TableToGeodatabase <input_table;input_table...> <output_geodatabase>
```


❖**Table to Table:** converts or copies dBASE, INFO, OLE DB, or geodatabase tables to a dBASE or geodatabase table.

```
TableToTable <input_table> <output_location> <output_table_name> {expression}
  {field_info} {configuration_keyword}
```

To Raster Toolset

Contains tools to convert any supported raster format to either a GRID, ERDAS IMAGINE, TIFF, or geodatabase format.

❖**ASCII to Raster:** converts an ASCII file representing raster data into a raster dataset.

```
ASCIIToRaster <in_ascii_file> <out_raster> {INTEGER | FLOAT}
```

❖**DEM to Raster:** converts a USGS DEM file into a raster dataset.

```
DEMToRaster <in_dem_file> <out_raster> {FLOAT | INTEGER} {z_factor}
```

❖**Feature to Raster:** converts a feature dataset to a raster dataset.

```
FeatureToRaster <in_features> <field> <out_raster> {cell_size}
```

❖**Float to Raster:** converts a file of binary floating point values representing raster data into a raster dataset.

```
FloatToRaster <in_float_file> <out_raster>
```

❖**Raster to Other Format (multiple):** converts one or more ArcGIS supported raster dataset formats to a GRID, IMAGINE, or TIFF format.

```
RasterToOtherFormat <input_rasters;input_rasters...> <output_workspace>
  {GRID | IMAGINE IMAGE | TIFF}
```

To Shapefile Toolset

Contains tools to create shapefiles from feature classes.

❖**Feature Class to Shapefile (multiple):** exports one or more feature classes to shapefiles in a designated folder.

```
FeatureclassToShapefile <input_feature_class;input_feature_class...>
  <output_folder>
```

Coverage Toolbox

Contains the original ArcInfo Workstation commands used to perform geoprocessing tasks with coverages.

Analysis Toolset

Contains tools and toolsets used for geospatial processing.

Extract Toolset

Contains tools used to select features or parts of features to create a new coverage.

Clip: extracts, using a cookie-cutter method, those features or portions of features from an input coverage that overlap with a clip coverage polygon.

```
Clip <in_cover> <clip_cover> <out_cover> {POLY | LINE | POINT | NET | LINK | RAW}
  {fuzzy_tolerance}
```

Select: extracts map features from the input coverage and stores them in an output coverage, based on logical expressions or by applying the criteria contained in a selection file.

```
Reselect <in_cover> <out_cover> <info_express;info_express...>
  {POLY | LINE | POINT | ANNO.subclass | ROUTE.subclass | SECTION.subclass |
  REGION.subclass} {selection_file} {out_feature_type}
```

Split: clips portions of the input coverage into multiple coverages.

```
Split <in_cover> <split_cover> <split_item> <path> {POLY | LINE | POINT | NET |
  LINK | RAW} {fuzzy_tolerance}
```

Overlay Toolset

Contains tools used to calculate the various options when overlaying two coverages.

Erase: erases the input coverage features or portions of features that overlap with the erase coverage polygons.

```
Erase <in_cover> <erase_cover> <out_cover> {POLY | LINE | POINT | NET | LINK | RAW}
  {fuzzy_tolerance}
```

Identity: computes the geometric intersection of two coverages, where all features of the input coverage and only those overlapping from the identity coverage are preserved.

```
Identity <in_cover> <identity_cover> <out_cover> {POLY | LINE | POINT}
  {fuzzy_tolerance} {JOIN | NO_JOIN}
```

Intersect: computes the geometric intersection of two coverages, where only those features in the area common to both coverages will be preserved.

```
Intersect <in_cover> <intersect_cover> <out_cover> {POLY | LINE | POINT}
  {fuzzy_tolerance} {JOIN | NO_JOIN}
```

Union: computes the geometric intersection of two polygon coverages. All polygons from both coverages will be split at their intersections and preserved in the output coverage.

```
Union <in_cover> <union_cover> <out_cover> {fuzzy_tolerance}
  {JOIN | NO_JOIN}
```

Update: replaces the input coverage areas with the update coverage polygons using a cut-and-paste type of operation.

```
Update <in_cover> <update_cover> <out_cover> {POLY | NET} {fuzzy_tolerance}
  {KEEP_BORDER | DROP_BORDER}
```

Proximity Toolset

Contains tools used in geoprocessing analysis involving distance.

Buffer: creates buffer polygons around specified input coverage features.

```
Buffer <in_cover> <out_cover> {LINE | POLY | POINT | NODE} {buffer_item}
  {buffer_table} {buffer_distance} {fuzzy_tolerance} {ROUND | FLAT} {FULL | LEFT
  | RIGHT}
```

Near: computes the distance from each point in a coverage to the nearest arc, point, or node in another coverage.

```
Near <in_cover> <near_cover> <out_cover> {LINE | POINT | NODE} {search_radius}
  {NO_LOCATION | LOCATION}
```

Point Distance: computes the distances between point features in one coverage to all points in a second coverage that are within the specified search radius.

```
PointDistance <from_cover> <to_cover> <out_info_table> {search_radius}
```

Point Node: transfers attributes from a point feature class to a node feature class.

```
PointNode <point_cover> <node_cover> {search_radius}
```

Thiessen: converts a point coverage to a coverage of Thiessen or proximal polygons.

```
Thiessen <in_cover> <out_cover> {proximal_tolerance}
```

Conversion Toolset

Contains tools and toolsets to convert a coverage to or from another file format.

From Coverage Toolset

Contains tools used to convert a coverage into various file formats.

Export to DLG: converts a coverage to an Optional Digital Line Graph (DLG-3) file format.

```
ArcDLG <in_cover> <out_dlg_file> {in_point_cover} {in_projection_file}
  {x_shift} {y_shift} {in_header_file} {TRANSFORM | NO_TRANSFORM}
```

Export to Interchange File: converts a coverage to an Interchange file.

```
Export <COVER | FONT | GRID | INFO | LINESET | MAP | MARKERSET | PLOT | SHADESET |
  STACK | STACKALL | TEXT | TEXTSET | TIN> <in_dataset> <interchange_file> {NONE |
  PARTIAL | FULL} {max_lines}
```

Export to S57: converts a coverage to an S-57 object format.

```
ArcS57 <in_workspace> <log_file> {out_workspace}
```

Export to SDTS: converts a coverage or grid to an SDTS Topological Vector Profile (TVP) or Point Profile Transfer file.

```
SDTSExport <TVP | POINT | RASTER> <in_dataset> <out_transfer_prefix>
  {in_point_cover} {out_DD_transfer} {conversion_control_file}
```

Export to VPF: converts a coverage into either a VPF coverage or VPF tile.

```
VPFExport <in_cover> <out_file> {tile_name} {control_file} {EXTRA | NO_EXTRA}
  {NO_FIT | FIT}
```

Ungenerate: Creates a text file of xy coordinates from a coverage.

```
Ungenerate <in_cover> <out_generate_file> <LINE | POINT | POLY | TIC | LINK |
  REGION.subclass | ANNO.subclass> {NODES | NO_NODES} {EXPONENTIAL | FIXED}
```

To Coverage Toolset

Contains tools used to convert various file formats into coverages.

Advanced Tiger Conversion: performs the Basic Tiger Conversion, followed by advanced operations including joining, defining a projection, and building topology.

```
TigerTool <in_tiger_file_prefix> <out_cover_prefix> {NO_JOIN | JOIN} {UTM |
  STATE} {zone_number} {1995 | 1997 | 1998 | 1999 | 2000 | 2002} {NO_RESTART |
  RESTART}
```

Basic Tiger Conversion: converts U.S. Bureau of Census TIGER/Line™ files into one or more coverages.

```
TigerArc <in_tiger_file_prefix> <out_cover> {out_point_cover}
  {out_landmark_cover} {1995 | 1997 | 1998 | 1999 | 2000 |2002}
```

Generate: creates a coverage from raw coordinates stored in a text file.

```
Generate <in_file> <out_cover> <LINES | ANNOTATIONS | CIRCLES | CURVES | FISHNET |
  LINKS | POINTS | POLYGONS | TICS>
```

Import from DLG: converts a Standard or Optional formatted Digital Line Graph (DLG) file into a coverage.

```
DLGArc <in_dlg_file> <out_cover> {out_point_cover} {NOFIRST | ALL | ATTRIBUTED}
  {x_shift} {y_shift} {category}
```

Import from Interchange File: converts an interchange file into a coverage.

```
Import <AUTO | COVER | FONT | GRID | INFO | LINESET | PLOT | MAP | MARKERSET |
  SHADESET | STACK | TEXT | TEXTSET | TIN> <interchange_file> <out_dataset>
```

Import from S57: converts data from an S-57 file format to a coverage.

```
S57Arc <in_s57_file> <out_workspace> {CLEAN | NO_CLEAN}
```

Import from SDTS: creates coverages or grids from an SDTS Topological Vector Profile (TVP) or Point Profile Transfer file.

```
SDTSImport <in_transfer_prefix> <output> {out_point_cover} {layer_name}
  {DD | DROP_DD} {PRESERVE | CONVERT}
```

Import from VPF: converts a VPF table into an INFO table or converts a VPF coverage or tile into a coverage.

```
VPFImport <input_vpf> <output> {tile_name} {control_file} {NO_EXTRA | EXTRA}
```

Data Management Toolset

Contains tools and toolsets to manage, manipulate, and maintain coverages and their attribute tables.

Aggregate Toolset

Contains tools used to combine coverages.

Append: combines an unlimited number of coverages into one coverage.

```
Append <in_covers;in_covers...> <out_cover> {FEATURES_ONLY |
  FEATURES_ATTRIBUTES} {POLY | LINE | POINT | NODE | NET | LINK | ANNO.subclass |
  SECTION.subclass | ROUTE.subclass | REGION.subclass}
  {NO | TICS_ONLY | FEATURES_ONLY | FEATURES_TICS}
```

Composite Features Toolset

Contains tools to create or convert regions within a coverage and to convert line features to routes.

Line Coverage to Region: converts arcs to preliminary regions in a new or existing coverage or appends preliminary regions to an existing region subclass.

```
RegionClass <in_cover> {out_cover} <out_subclass> {in_region_item}
  {out_region_item} {selection_file} {MULTIRING | SINGLERING}
```

Line Coverage to Route: creates a route system by creating whole arc sections for each arc in the input coverage. It can also be used to append arcs to an existing route system.

```
ArcRoute <in_cover> <out_route_system> {in_route_item} {out_route_item}
  {measure_item} {UL | UR | LL | LR} {BLANK | NO_BLANK}
```

Polygon Coverage to Region: converts a polygon coverage to a region subclass. Each polygon in the in_cover becomes a region of the output subclass.

```
PolyRegion <in_cover> <out_cover> <out_subclass>
```

Region to Polygon Coverage: converts a region subclass into a polygon coverage and creates an INFO table containing overlapping region information.

```
RegionPoly <in_cover> <out_cover> <in_subclass> {out_table}
```

Generalization Toolset

Contains tools to derive data with less detail and complexity from coverage features.

Aggregate Polygons: combines disjointed and adjacent polygons into new area features based on a distance.

```
AggregatePolygons <in_cover> <out_cover> <cell_size> <distance>
  {NON_ORTHOGONAL | ORTHOGONAL}
```

Collapse Dual Lines to Centerline: derives centerlines (single lines) from dual-line features, such as road casings, based on specified width tolerances.

```
CollapseDualLineToCenterline <in_cover> <out_cover> <maximum_width>
  {minimum_width}
```

Dissolve: merges adjacent polygons, lines, or regions that have the same value for a specified item.

```
Dissolve <in_cover> <out_cover> <dissolve_item>
  {POLY | LINE | NET | REGION.subclass}
```

Eliminate: merges selected polygons with neighboring polygons that have the largest shared border between them or the largest area.

```
Eliminate <in_cover> <out_cover> <info_express;info_express...>
  {NO_KEEP_EDGE | KEEP_EDGE} {POLY | LINE} {selection_file} {BORDER | AREA}
```

Find Conflicts: searches a region coverage for overlapping and closely spaced buildings, based on a specified distance, and records the occurrences.

```
FindConflicts <in_cover> <out_cover> <conflict_distance>
```

Simplify Building: simplifies the boundary or footprint of building polygons while maintaining their essential shape and size.

```
SimplifyBuilding <in_cover> <out_cover> <simplification_tolerance>
  {minimum_area} {selection_file} {NOT_CHECK | CHECK_CONFLICT}
```

Simplify Line or Polygon: removes small fluctuations or extraneous bends from a line or polygon, while preserving its essential shape.

```
SimplifyLineOrPolygon <in_cover> <out_cover> <simplification_tolerance>
  {POINT_REMOVE | BEND_SIMPLIFY} {NO_ERROR_CHECK | ERROR_CHECK}
```

Indexes Toolset

Contains tools to add or remove attribute indexes.

Drop Index: drops an attribute index from the specified item and INFO table.

```
DropIndex <in_info_table> {index_item;index_item...}
```

Index Item: creates an attribute index to increase access speed to the specified item during query operations.

```
IndexItem <in_info_table> <index_item>
```

Items Toolset

Contains tools to add or remove items (fields) in INFO tables.

Add Item: adds a blank or zero item to an INFO table.

```
AddItem <in_info_table> <out_info_table> <item_name> <item_width>
  <output_width> <BINARY | CHARACTER | DATE | FLOATING | INTEGER | NUMERIC>
  {decimal_places} {start_item}
```

Drop Item: deletes one or more items from an INFO table.

```
DropItem <in_info_table> <out_info_table> <drop_item;drop_item...>
```

Joins Toolset

Contains a tool to join INFO tables.

Join Info Tables: joins the item definitions and values of two tables based on a shared item.

```
JoinItem <in_info_table> <join_info_table> <out_info_table> <relate_item>
  {start_item} {LINEAR | ORDERED | LINK}
```

Projections Toolset

Contains tools to set a projection or reproject or transform a coverage.

Define Projection: creates or modifies the coordiante system information (including projection parameters, such as datum and spheroid) stored in the coverage's projection definition file (.prj).

```
DefineProjection <in_cover> <projection_file>
```

Project: changes the coordinate system of a coverage including its datum or spheroid.

```
Project <in_cover> <out_cover> <projection_file>
```

Transform: moves all features in the coverage based on a set of from and to control points.

```
Transform <in_cover> <out_cover> {AFFINE | PROJECTIVE | SIMILARITY}
```

Tables Toolset

Contains tools used for editing the associated attribute tables.

Add XY Coordinates: calculates and adds xy coordinates of labels or points to the coverage PAT or xy coordinates of nodes to the coverage NAT.

```
AddXY <in_cover> {POINT | NODE}
```

Renumber Nodes: updates arc–node topology by renumbering nodes for coverage arcs and identifies arcs that share the same node locations.

```
Renode <in_cover> {from_item} {to_item}
```

Update IDs: updates User-IDs in a coverage after they have been modified in a feature attribute table.

```
IDEdit <in_cover> <POLY | LINE | POINT | ANNO.subclass>
```

Tolerances Toolset

Contains a tool to adjust coverage-associated tolerances.

Tolerance: sets the tolerances associated with a coverage.

```
Tolerance <in_cover> {FUZZY | DANGLE | TIC_MATCH | EDIT | NODESNAP | WEED | GRAIN |
  SNAP} {tolerance_value}
```

Topology Toolset

Contains tools used to develop the topologic relationship within a coverage.

Build: creates or updates feature attribute tables and polygon topology.

```
Build <in_cover> <POINT | LINE | POLY | NODE | ANNO> {anno_subclass}
```

Clean: generates a coverage with correct polygon or arc–node topology.

```
Clean <in_cover> {out_cover} {dangle_length} {fuzzy_tolerance} {POLY | LINE}
```

Create Labels: creates label points for polygons that have no labels and assigns each a User-ID.

```
CreateLabels <in_cover> {id_base}
```

VPF Tile Topology: creates cross-tile topology for all tiled coverages in a Vector Product Format (VPF) database library or creates topology for an individual tile in a VPF library.

```
VPFTile <VPF_library> {sig_digits} {93 | 96} {ALL | VPF_cover}
```

Workspace Management Toolset

Contains a tool to manage coverages within a workspace.

Create Coverage: creates a new, empty coverage.

```
Create <out_cover> {template_cover}
```

Data Management Toolbox

Contains the tools to develop, manage, and manipulate feature classes, datasets, and layers.

Database Toolset

Contains tools that improve database performance.

Compact: reduces the size and optimizes the performance of a personal geodatabase.

```
Compact <in_workspace>
```

Compress: removes all unreferenced database states and redundant rows in a version.

```
Compress <in_workspace>
```

Disconnected Editing Toolset

Contains tools to check out or update data from a shared data system.

Check In: checks datasets back in to the parent ArcSDE® geodatabase from a checked out ArcSDE or personal geodatabase.

```
Checkin <in_workspace> <dest_workspace> {NON_RECONCILE | RECONCILE}
```

Check In from Delta: checks in a geodatabase from a delta geodatabase or an XML file. A delta geodatabase contains only the changes exported from a check-out geodatabase.

```
CheckinDelta <in_delta_database> <dest_workspace>
  {NON_RECONCILE | RECONCILE}
```

Check Out: checks out data from an ArcSDE geodatabase to an ArcSDE or personal geodatabase for offline editing.

```
CheckOut <in_data;in_data...> <out_workspace> <out_name>
  <DATA | SCHEMA_ONLY> <NO_REUSE | REUSE> <RELATED | NO_RELATED>
```

Export to Delta: exports changes in a check-out geodatabase to a delta database or XML file. A delta geodatabase contains changes exported from a check-out geodatabase.

```
ExporttoDelta <in_workspace> <dest_delta_database>
```

Domains Toolset

Contains tools for the management of domains, both coded and attribute, within a workspace.

Add Coded Value to Domain: adds a new value to a domain's coded value list.

```
AddCodedValueToDomain <in_workspace> <domain_name> <code> <code_description>
```

Assign Domain to Field: sets the domain for a particular field and optionally for a subtype.

```
AssignDomainToField <in_table> <field_name> <domain_name>
  {subtype_code;subtype_code...}
```

Create Domain: creates an attribute domain in the specified workspace.

```
CreateDomain <in_workspace> <domain_name> <domain_description>
  <SHORT | LONG | FLOAT | DOUBLE | TEXT | DATE> {CODED | RANGE}
  {DEFAULT | DUPLICATE | GEOMETRY_RATIO} {DEFAULT | SUM_VALUES | AREA_WEIGHTED}
```

Delete Coded Value from Domain: removes a value from a coded value domain.

```
DeleteCodedValueFromDomain <in_workspace> <domain_name> <code;code...>
```

Delete Domain: deletes a domain from a workspace.

```
DeleteDomain <in_workspace> <domain_name>
```

Domain to Table: creates a table from an attribute domain.

```
DomainToTable <in_workspace> <domain_name> <out_table> <code_field>
  <description_field> {configuration_keyword}
```

Remove Domain from Field: removes an attribute domain association from a feature class field.

```
RemoveDomainFromField <in_table> <field_name> {subtype_code;subtype_code...}
```

Set Value for Range Domain: sets the minimum and maximum values for an existing range domain.

```
SetValueForRangeDomain <in_workspace> <domain_name> <min_value> <max_value>
```

Table to Domain: creates or updates a coded value domain from a table.

```
TableToDomain <in_table> <code_field> <description_field> <in_workspace>
  <domain_name> <domain_description> <APPEND | REPLACE>
```

Feature Class Toolset

Contains tools for basic feature class management including appending and integrating multiple feature classes.

❖**Append Annotation Feature Classes:** combines annotation from multiple input feature classes to create a new annotation feature class.

```
AppendAnnotation <input_features;input_features...> <output_feature_class>
  <reference_scale> {CREATE_CLASSES | ONE_CLASS_ONLY}
  {NO_SYMBOL_REQUIRED | REQUIRE_SYMBOL} {AUTO_CREATE | NO_AUTO_CREATE}
  {AUTO_UPDATE | NO_AUTO_UPDATE}
```

Calculate Default Cluster Tolerance: calculates a default cluster tolerance value.

```
CalculateDefaultClusterTolerance <in_features>
```

Calculate Default Spatial Grid Index: calculates a spatial grid value, used to quickly locate features in a dataset that match the criteria of a spatial search.

```
CalculateDefaultGridIndex <in_features>
```

❖**Create Feature Class:** creates a new feature class.

```
CreateFeatureClass <out_path> <out_name> <POLYGON | POINT | MULTIPOINT |
  POLYLINE> {template;template...} {DISABLED | SAME_AS_TEMPLATE | ENABLED}
  {DISABLED | SAME_AS_TEMPLATE | ENABLED} {spatial_reference}
  {configuration_keyword} {spatial_grid_1} {spatial_grid_2} {spatial_grid_3}
```

Integrate: compares one or more feature classes and makes any lines, points, or vertices within a specified distance range identical or coincident.

```
Integrate <features{Ranks};features{Ranks}...> {cluster_tolerance}
```

❖**Update Annotation Feature Class:** updates the input annotation feature class with text attribute fields and optionally populates the value of each new field for every feature in the feature class.

```
UpdateAnnotation <in_features> {POPULATE | DO_NOT_POPULATE}
```

Features Toolset

Contains tools to manage and enrich feature classes, such as inspecting and correcting potential errors with data and creating different geometries.

❖**Add XY Coordinates:** adds the field POINT_X and POINT_Y for labels, points, tics, or nodes to the input feature class.

```
AddXY <in_features>
```

❖**Check Geometry:** checks the validity of the geometries of features.

```
CheckGeometry <in_features;in_features...> <out_table>
```

❖**Copy Features:** copies the selected features to a new feature class.

```
CopyFeatures <in_features> <out_feature_class> {configuration_keyword}
  {spatial_grid_1} {spatial_grid_2} {spatial_grid_3}
```

Delete Features: deletes features from the input features.

```
DeleteFeatures <in_features>
```

Feature Envelope to Polygon: creates polygons from the envelopes of each feature in the input features.

```
FeatureEnvelopeToPolygon <in_features> <out_feature_class>
  {SINGLEPART | MULTIPART}
```

Feature to Line: creates an output line feature class from input polygon features.

```
FeatureToLine <in_features;in_features...> <out_feature_class>
  {cluster_tolerance} {ATTRIBUTES | NO_ATTRIBUTES}
```

Feature to Point: creates a point feature class from the centroids or midpoints of the input features.

```
FeatureToPoint <in_features> <out_feature_class> {CENTROID | INSIDE}
```

Feature to Polygon: creates a new polygon feature class from input line features.

```
FeatureToPolygon <in_features;in_features...> <out_feature_class>
  {cluster_tolerance} {ATTRIBUTES | NO_ATTRIBUTES} {label_features}
```

Feature Vertices to Points: creates points from the vertex locations of the input features.

```
FeatureVerticesToPoints <in_features> <out_feature_class>
  {ALL | MID | START | END | BOTHENDS}
```

❖**Multipart to Singlepart:** breaks any multipart features into single features.

```
MultipartToSinglepart <in_features> <out_feature_class>
```

Polygon to Line: generates a new line feature class from input polygon features.

```
PolygonToLine <in_features> <out_feature_class>
```

❖**Repair Geometry:** repairs geometry problems in a feature class or layer.

```
RepairGeometry <in_features>
```

Split Line at Vertices: splits a feature class at any intersection contained in that feature.

```
SplitLine <in_features> <out_feature_class>
```

Fields Toolset

Contains tools to add and make changes to the fields in the tables of a feature class.

❖**Add Field:** adds a field to the table of a feature class, layer or raster catalog.

```
AddField <in_table> <field_name> <LONG | TEXT | FLOAT | DOUBLE | SHORT | DATE |
  BLOB> {field_precision} {field_scale} {field_length} {field_alias}
  {NULLABLE | NON_NULLABLE} {NON_REQUIRED | REQUIRED} {field_domain}
```

Assign Default to Field: creates a default value for a specified field and automatically applies a user-determined value to a certain field for every row added to the table or feature class.

```
AssignDefaultToField <in_table> <field_name> <default_value>
  {subtype_code;subtype_code...}
```

❖**Calculate Field:** calculates the value of a field using an expression.

```
CalculateField <in_table> <field> <expression>
```

❖**Delete Field:** deletes one or more fields from a table or feature class.

```
DeleteField <in_table> <drop_field;drop_field...>
```

General Toolset

Contains tools allowing for some simple dataset changes.

❖**Append:** combines many feature classes into one feature class.

```
Append <inputs;inputs...> <target> {TEST | NO_TEST}
```

❖**Copy:** copies feature datasets, feature classes, or tables and pastes them to another location.

```
Copy <in_data> <out_data> {data_type}
```

❖**Delete:** deletes feature datasets, feature classes, rasters, or tables.

```
Delete <in_data> {data_type}
```

❖**Rename:** changes the name of data, such as feature datasets, feature classes, rasters, tables, or toolboxes.

```
Rename <in_data> <out_data> {data_type}
```

❖**Select Data:** selects any data type on disk as input. This tool is intended for use within ModelBuilder™, not at the command line.

```
SelectData <in_data_element> <out_data_element>
```

Generalization Toolset

Contains tools to derive data with less detail and complexity from a dataset.

❖**Dissolve:** aggregates features based on one or more specified attributes.

```
Dissolve <in_features> <out_feature_class> {dissolve_field;dissolve_field...}
  {field{Statistics_Type}; field{Statistics_Type}...}
```

Eliminate: eliminates features.

```
Eliminate <in_features> <out_feature_class> {LENGTH | AREA}
```

Simplify Line: removes small fluctuations or extraneous bends from a line in a feature class.

```
SimplifyLine <in_features> <out_feature_class> <POINT_REMOVE | BEND_SIMPLIFY>
  <tolerance> {FLAG_ERRORS | RESOLVE_ERRORS} {KEEP_COLLAPSED_POINTS | NO_KEEP}
```

Smooth Line: reduces the number of segments used to represent a line in a feature class.

```
SmoothLine <in_features> <out_feature_class> <PAEK | BEZIER_INTERPOLATION>
  <tolerance> {FIXED_CLOSED_ENDPOINT | NO_FIXED}
```

Indexes Toolset

Contains tools to create, alter, and remove indexes.

Add Attribute Index: adds an index to an existing table, feature class, shapefile, coverage, or attributed relationship class.

```
AddIndex <in_table> <fields;fields...> <index_name> {NON_UNIQUE | UNIQUE}
  {NON_ASCENDING | ASCENDING}
```

Add Spatial Index: creates a new spatial index for a feature class in an ArcSDE geodatabase.

```
AddSpatialIndex <in_features> <grid_size_1> {grid_size_2} {grid_size_3}
```

Remove Attribute Index: deletes an index from an existing table, feature class, shapefile, coverage, or attributed relationship class.

```
RemoveIndex <in_table> <index_name;index_name...>
```

Remove Spatial Index: deletes the spatial index for a feature class in an ArcSDE geodatabase.

```
RemoveSpatialIndex <in_features>
```

Joins Toolset

Contains tools to add or remove a table join.

Add Join: links a layer to a table (or a table to a table) based on a common field.

```
AddJoin <in_layer_or_view> <in_field> <join_table> <join_field> {KEEP_ALL|
  KEEP_COMMON}
```

Remove Join: removes the link between two tables.

```
RemoveJoin <in_layer_or_view> <join_name>
```

Layers and Table Views Toolset

Contains tools for creating and manipuating layers, layer files, and table views.

❖**Make Feature Layer:** creates a temporary layer.

```
MakeFeatureLayer <in_features> <out_layer> {where_clause} {workspace}
  {field_info}
```

❖**Make Query Table:** represents the results of a SQL query to a database in a layer or table view.

```
MakeQueryTable <in_table;in_table...> <out_table> {USE_KEY_FIELDS |
  ADD_VIRTUAL_KEY_FIELD | NO_KEY_FIELD} {in_key_field;in_key_field...}
  <field{Alias};field{Alias}...> {where_clause}
```

❖**Make Raster Catalog Layer:** makes a temporary raster catalog layer.

```
MakeRasterCatalogLayer <in_raster_catalog> <layer_name> {where_clause}
  {workspace} {field_info}
```

❖**Make Raster Layer:** makes a temporary raster layer.

```
MakeRasterLayer <in_raster> <out_raster_layer> {where_clause} {envelope}
```

❖**Make Table View:** creates a temporary table.

```
MakeTableView <in_table> <out_name> {where_clause} {workspace} {field_info}
```

❖**Make XY Event Layer:** creates a temporary event layer with xy coordinates.

```
MakeXYEventLayer <table> <in_x_field> <in_y_field> <out_layer>
  {spatial_reference}
```

Save to Layer File: creates a layer file on disk.

```
SaveToLayerFile <in_layer> <out_layer>
```

Select Layer by Attribute: creates, updates, or removes the selection on a layer or table view using an attribute query.

```
SelectLayerByAttribute <in_layer_or_view> {NEW_SELECTION | ADD_TO_SELECTION |
  REMOVE_FROM_SELECTION | SUBSET_SELECTION | SWITCH_SELECTION |
  CLEAR_SELECTION} {where_clause}
```

Select Layer by Location: creates, updates, or removes the selection of features in a layer based on a spatial relationship.

```
SelectLayerByLocation <in_layer> {INTERSECT | WITHIN_A_DISTANCE |
  COMPLETELY_CONTAINS | COMPLETELY_WITHIN | HAVE_THEIR_CENTER_IN |
  SHARE_A_LINE_SEGMENT_WITH | BOUNDARY_TOUCHES | ARE_IDENTICAL_TO |
  CROSSED_BY_THE_OUTLINE_OF | CONTAINS | CONTAINED_BY} {select_features}
  {search_distance} {NEW_SELECTION | ADD_TO_SELECTION | REMOVE_FROM_SELECTION |
  SUBSET_SELECTION | SWITCH_SELECTION}
```

Projections and Transformations Toolset

Contains tools to set the projection as well as reproject or transform a dataset.

❖**Define Projection:** sets the projection information for a dataset.

```
DefineProjection <in_dataset> <coordinate_system>
```

Feature (Projections and Transformations) Toolset

Contains tools to convert a geographic dataset from one coordinate system to another.

❖**Batch Project:** changes the coordinate system of one or more feature classes including the datum or spheroid.

```
BatchProject <input_feature_class_or_dataset;
  input_feature_class_or_dataset...> <output_workspace>
  {output_coordinate_system} {template_dataset} {transformation}
```

❖**Create Spatial Reference:** creates spatial reference and domains.

```
CreateSpatialReference {spatial_reference} {spatial_reference_template}
  {xy_domain} {z_domain} {m_domain} {template;template...} {expand_ratio}
```

❖**Project:** changes the coordinate system of a feature class including its datum or spheroid.

```
Project <in_dataset> <out_dataset> <out_coordinate_system>
  {transform_method;transform_method...}
```

Raster (Projections and Transformations) Toolset

Contains tools to set the projection, reproject, reorient, or relocate a raster dataset.

❖**Flip:** flips a raster dataset along a horizontal axis.

```
Flip <in_raster> <out_raster>
```

❖**Mirror:** fllips a raster dataset along the vertical axis.

```
Mirror <in_raster> <out_raster>
```

❖**Project Raster:** converts a raster dataset between two coordinate systems.

```
ProjectRaster <in_raster> <out_raster> <out_coordinate_system>
  {NEAREST | BILINEAR | CUBIC} {cell_size}
```

❖**Rescale:** scales a raster by the specified x and y scale factors.

```
Rescale <in_raster> <out_raster> <x_scale> <y_scale>
```

❖**Rotate:** rotates a raster dataset around a specified point by a specified angle.

```
Rotate <in_raster> <out_raster> <angle> {pivot_point} {NEAREST | BILINEAR |
  CUBIC}
```

❖**Shift:** shifts a raster by the specified x and y shift values.

```
Shift <in_raster> <out_raster> <x_value> <y_value> {in_snap_raster}
```

❖**Warp:** transforms or rubber sheets a raster dataset along a set of links using a polynomial transformation.

```
Warp <in_raster> <source_control_points;source_control_points...>
  <target_control_points;target_control_points...> <out_raster>  {POLYORDER1 |
  POLYORDER2 | POLYORDER3} {NEAREST | BILINEAR | CUBIC}
```

Raster Toolset

Contains tools to create and manage raster datasets and raster catalogs.

❖**Batch Build Pyramids:** allows you to build pyramids on multiple raster datasets.

```
BatchBuildPyramids <input_raster_datasets;input_raster_datasets...>
```

❖**Batch Calculate Statistics:** allows you to calculate statistics on multiple raster datasets.

```
BatchCalculateStatistics <input_raster_datasets;input_raster_datasets...>
  {number_of_columns_to_skip} {number_of_rows_to_skip}
  {ignore_values;ignore_values...}
```

❖**Build Pyramids:** builds raster pyramids for a raster dataset.

```
BuildPyramids <in_raster_dataset>
```

❖**Calculate Statistics:** calculates statistics for a raster dataset.

```
CalculateStatistics <in_raster_dataset> {x_skip_factor} {y_skip_fator}
  {ignore_values;ignore_values...}
```

❖**Clip:** creates a rectangular spatial subset of a raster dataset.

```
Clip <in_raster> <rectangle> <out_raster>
```

❖**Composite Bands:** creates a multiband raster dataset from one or more raster datasets.

```
CompositeBands <in_rasters;in_rasters...> <out_raster>
```

❖**Copy Raster:** converts a raster dataset into a GRID, TIFF, IMAGINE, or geodatabase raster dataset or loads raster datasets into a raster catalog.

```
CopyRaster <in_raster> <out_raster_dataset> {configuration_keyword}
  {background_value} {nodata_value} {NONE | ONEBITTO8BIT} {NONE | COLORMAPTORGB}
```

❖**Copy Raster Catalog Items:** makes a copy of a raster catalog including all its contents.

```
CopyRasterCatalogItems <in_raster_catalog> <out_raster_catalog>
  {configuration_keyword} {spatial_grid_1} {spatial_grid_2} {spatial_grid_3}
```

❖**Create Raster Catalog:** creates an empty raster catalog in a geodatabase.

```
CreateRasterCatalog <out_path> <out_name> {raster_spatial_reference}
  {spatial_reference} {configuration_keyword} {spatial_grid_1}
  {spatial_grid_2} {spatial_grid_3} {MANAGED | UNMANAGED}
```

❖**Create Raster Dataset:** creates an empty raster dataset in a geodatabase.

```
CreateRasterDataset <out_path> <out_name> {cellsize} {8_BIT_UNSIGNED | 1_BIT |
  2_BIT | 4_BIT | 8_BIT_SIGNED | 16_BIT_UNSIGNED | 16_BIT_SIGNED |
  32_BIT_UNSIGNED | 32_BIT_SIGNED | 32_BIT_FLOAT | 64_BIT}
  {raster_spatial_reference} {number_of_bands} {configuration_keyword}
  {pryamids} {tile_size} {compression} {pyramid_origin}
```

❖**Delete Raster Catalog Items:** deletes the raster catalogs contained in a raster catalog.

```
DeleteRasterCatalogItems <in_raster_catalog>
```

❖**Mosaic:** mosaics multiple rasters into a single raster.

```
Mosaic <inputs;inputs...> <target> {FIRST | LAST | BLEND | MEAN | MINIMUM |
  MAXIMUM} {REJECT | FIRST | LAST | MATCH} {background_value} {nodata_value}
  {NONE | ONEBITTO8BIT} {mosaicking_tolerance}
```

❖**Mosaic to New Raster:** mosaics multiple rasters into a new, single raster dataset.

```
MosaicToNewRaster <input_rasters;input_rasters...> <output_location>
  <raster_dataset_name_with_extension> {cell_size} {8_BIT_UNSIGNED | 1_BIT |
  2_BIT | 4_BIT | 8_BIT_SIGNED | 16_BIT_UNSIGNED | 16_BIT_SIGNED |
  32_BIT_UNSIGNED | 32_BIT_SIGNED | 32_BIT_FLOAT | 64_BIT}
  {coordinate_system_for_the_raster} {number_of_bands} {FIRST | LAST | BLEND |
  MEAN | MINIMUM | MAXIMUM} {REJECT | FIRST | LAST | MATCH}
```

❖**Resample:** changes the cell size of a grid.

```
Resample <in_raster> <out_raster> <cell_size> {NEAREST | BILINEAR | CUBIC}
```

Relationship Classes Toolset

Contains tools to create associations between feature classes as well as feature classes and tables.

Create Relationship Class: creates a relationship class to store an association between features in two feature classes or tables.

```
CreateRelationshipClass <origin_table> <destination_table>
  <out_relationship_class> <SIMPLE | COMPOSITE> <forward_label>
  <backward_label> <NONE | FORWARD | BACKWARD | BOTH> <ONE_TO_ONE | ONE_TO_MANY |
  MANY_TO_MANY> <NONE | ATTRIBUTED> <origin_primary_key> <origin_foreign_key>
  {destination_primary_key} {<destination_foreign_key}
```

Table To Relationship Class: creates an attributed relationship class from the Origin, Destination and Relationship tables.

```
TableToRelationshipClass <origin_table> <destination_table>
  <out_relationship_class> <SIMPLE | COMPOSITE> <forward_label>
  <backward_label> <NONE | FORWARD | BACKWARD | BOTH> <ONE_TO_ONE | ONE_TO_MANY |
  MANY_TO_MANY> <relationship_table> <attribute_fields;attribute_fields...>
  <origin_primary_key> <origin_foreign_key> <destination_primary_key>
  <destination_foreign_key>
```

Subtypes Toolset

Contains tools to manage the subtypes of a feature class or a table.

Add Subtype: adds a new subtype to the set of subtypes in a feature class or table.

```
AddSubtype <in_table> <subtype_code> <subtype_description>
```

Remove Subtype: deletes a subtype from the set of associated subtypes in a feature class or table.

```
RemoveSubtype <in_table> <subtype_code;subtype_code...>
```

Set Default Subtype: sets the default subtype value.

```
SetDefaultSubtype <in_table> <subtype_code>
```

Set Subtype Field: defines the field in the feature class or table that stores the subtype codes.

```
SetSubtypeField <in_table> <field>
```

Table Toolset

Contains tools to help you create and evaluate tabular data from a variety of sources.

Analyze: updates relational database management system statistics for a dataset.

```
Analyze <in_dataset> <BUSINESS | FEATURE | RASTER | ADDS | DELETES>
```

Change Privileges: changes the user privileges associated with a dataset.

```
ChangePrivileges <in_dataset> <user> <SELECT | UPDATE | INSERT | DELETE>
```

❖**Copy Rows:** copies rows from a feature class, layer, table, or table view.

```
CopyRows <in_rows> <out_table> {configuration_keyword}
```

❖**Create Table:** creates an empty geodatabase or dBASE table.

```
CreateTable <out_path> <out_name> {template;template...}
  {configuration_keyword}
```

❖**Delete Rows:** deletes rows from a feature class, layer, table, or table view.

```
DeleteRows <in_rows>
```

Get Count: returns the number of rows in the feature class, layer, table, or table view.

```
GetCount <in_rows>
```

Pivot Table: sorts and summarizes table fields, based on selected fields to reduce redundancy.

```
PivotTable <in_table> <fields;fields...> <pivot_field> <value_field>
  <out_table>
```

Topology Toolset

Contains tools to establish and manage topological relationships between features.

Add Feature Class to Topology: adds a new feature class to a topological relationship.

```
AddFeatureClassToTopology <in_topology> <in_feature_class> <xy_rank> <z_rank>
```

Add Rule to Topology: adds a rule to the management of the topological relationship within a feature dataset.

```
AddRuleToTopology <in_topology> <Must Not Have Gaps (Area) | Must Not Overlap (Area) | Must Be Covered By Feature Class Of (Area-Area) | Must Cover Each Other (Area-Area) | Must Be Covered By (Area-Area) | Must Not Overlap With (Area-Area) | Must Be Covered By Boundary Of (Line-Area) | Must Be Covered By Boundary Of (Point-Area) | Must Be Properly Inside (Point-Area) | Must Not Overlap (Line) | Must Not Intersect (Line) | Must Not Have Dangles (Line) | Must Not Have Pseudo-Nodes (Line) | Must Be Covered By Feature Class Of (Line-Line) | Must Not Overlap With (Line-Line) | Must Be Covered By (Point-Line) | Must Be Covered By Endpoint Of (Point-Line) | Boundary Must Be Covered By (Area-Line) | Boundary Must Be Covered By Boundary Of (Area-Area) | Must Not Self-Overlap (Line) | Must Not Self-Intersect (Line) | Must Not Intersect Or Touch Interior (Line) | Endpoint Must Be Covered By (Line-Point) | Contains Point (Area-Point) | Must Be Single Part (Line)> <in_feature_class> {subtype} {in_feature_class2} {subtype2}
```

Create Topology: creates a topology within a feature dataset.

```
CreateTopology <in_dataset> <out_name> <in_cluster_tolerance>
```

Remove Feature Class from Topology: removes a feature class from the topological relationship.

```
RemoveFeatureClassFromTopology <in_topology> <in_feature_class>
```

Remove Rule from Topology: removes a rule from a topological relationship.

```
RemoveRuleFromTopology <in_topology> <in_rule>
```

Set Cluster Tolerance: alters the cluster tolerance in a topological relationship.

```
SetClusterTolerance <in_topology> <cluster_tolerance>
```

Validate Topology: evaluates the features against the rules and finds any new errors related to new rules or feature classes.

```
ValidateTopology <in_topology> {FULL_EXTENT | VISIBLE_EXTENT}
```

Versions Toolset

Contains tools to make adjustments to the versions of the data.

Alter Version: alters the properties of any of the versions of the dataset including name, description, and access permissions.

```
AlterVersion <in_workspace> <in_version> {name} {description} {PRIVATE | PUBLIC | PROTECTED}
```

Create Version: creates a new version of the specified database.

```
CreateVersion <in_workspace> <parent_version> <version_name>
```

Delete Version: deletes the specified version from the input workspace.

```
DeleteVersion <in_workspace> <version_name>
```

Post Version: applies the currect edit session to the reconciled target version during versioned geodatabase editing.

```
PostVersion <in_workspace> <version_name>
```

Reconcile Version: reconciles an ArcSDE version against a parent version in its lineage.

```
ReconcileVersion <in_workspace> <version_name> <target_name>
```

Register as Versioned: registers an ArcSDE dataset as versioned in ArcCatalog™.

```
RegisterAsVersioned <in_dataset>
```

Unregister as Versioned: unregisters a dataset as versioned in ArcCatalog.

```
UnregisterAsVersioned <in_dataset> {KEEP_EDIT | NO_KEEP_EDIT}
```

Workspace Toolset

Contains tools to create the storage models used with ArcGIS.

Create ArcInfo Workspace: creates a workspace with an INFO subdirectory.

```
CreateArcInfoWorkspace <out_folder_path> <out_name>
```

Create Feature Dataset: creates an empty feature dataset within an existing geodatabase.

```
CreateFeatureDataset <out_dataset_path> <out_name> {spatial_reference}
```

Create Folder: creates a new folder.

```
CreateFolder <in_folder_path> <out_name>
```

Create Personal GDB: creates a new personal geodatabase.

```
CreatePersonalGDB <out_folder_path> <out_name>
```

Geocoding Toolbox

Contains tools used to manage a geocoding service and run geocoding actions.

❖**Automate Geocoding Indexes:** creates an automatically updating relationship between the reference data and the geocoding index(es) of an address locator.

```
AutomateGeocodingIndexes <in_address_locator>
```

❖**Create Address Locator:** creates a new address locator.

```
CreateAddressLocator <in_address_locator_style>
  <reference_data{Role};reference_data{Role}...> <in_field_map>
  <out_address_locator>
```

❖**Deautomate Geocoding Indexes:** removes the automatically updating relationship between the reference data and geocoding index of an address locator.

```
DeautomateGeocodingIndexes <in_address_locator>
```

❖**Delete Address Locator:** deletes an address locator.

```
DeleteAddressLocator <in_address_locator>
```

❖**Geocode Addresses:** creates a point feature class from a table of addresses.

```
GeocodeAddresses <in_table> <address_locator> <in_address_fields>
  <out_feature_class> {STATIC | DYNAMIC}
```

❖**Rebuild Geocoding Indexes:** rebuilds the indexes of an address locator.

```
RebuildGeocodingIndex <in_address_locator>
```

❖**Standardize Addresses:** standardizes the address information in a table or feature class.

```
StandardizeAddresses <in_address_data>
  <in_input_address_fields;in_input_address_fields...>
  <in_address_locator_style>
  <in_output_address_fields;in_output_address_fields...> <out_address_data>
  {STATIC | DYNAMIC}
```

Linear Referencing Toolbox

Contains tools to model relative locations along linear features and associate multiple sets of attributes to portions of linear features.

◆**Calibrate Routes:** adjusts route measures by reading measure information stored as an attribute in a point feature class.

```
CalibrateRoutes <in_route_features> <route_id_field> <in_point_features>
  <point_id_field> <measure_field> <out_feature_class> {DISTANCE | MEASURES}
  {search_radius} {BETWEEN | NO_BETWEEN} {BEFORE | NO_BEFORE} {AFTER | NO_AFTER}
  {IGNORE | NO_IGNORE} {KEEP | NO_KEEP} {INDEX | NO_INDEX}
```

◆**Create Routes:** creates routes from existing lines.

```
CreateRoutes <in_line_features> <route_id_field> <out_feature_class>
  <LENGTH | ONE_FIELD | TWO_FIELDS> <from_measure_field> <to_measure_field>
  {UPPER_LEFT | LOWER_LEFT | UPPER_RIGHT | LOWER_RIGHT} {measure_factor}
  {measure_offset} {IGNORE | NO_IGNORE} {INDEX | NO_INDEX}
```

Dissolve Route Events: removes redundant information from event tables or separates event tables having more than one descriptive attribute into separate tables.

```
DissolveRouteEvents <in_events> <in_event_properties> <dissolve_field;
  dissolve_field...> <out_table> <out_event_properties>
  {DISSOLVE | CONCATENATE} {INDEX | NO_INDEX}
```

Locate Features Along Routes: computes the intersection of input features (point, line, or polygon) and route features and writes the route and measure information to a new event table.

```
LocateFeaturesAlongRoutes <in_features> <in_routes> <route_id_field>
  <radius_or_tolerance> <out_table> <out_event_properties> {FIRST | ALL}
  {DISTANCE | NO_DISTANCE} {ZERO | NO_ZERO} {FIELDS | NO_FIELDS}
```

❖**Make Route Event Layer:** creates a temporary route event layer.

```
MakeRouteEventLayer <in_routes> <route_id_field> <in_table>
  <in_event_properties> <out_layer> {offset_field} {NO_ERROR_FIELD |
  ERROR_FIELD} {NO_ANGLE_FIELD | ANGLE_FIELD} {NORMAL | TANGENT}
  {ANGLE | COMPLEMENT} {LEFT | RIGHT} {POINT | MULTIPOINT}
```

Overlay Route Events: combines two input event tables to create a single output event table, using either a union or intersection operation.

```
OverlayRouteEvents <in_table> <in_event_properties> <overlay_table>
  <overlay_event_properties> <INTERSECT | UNION> <out_table>
  <out_event_properties> {ZERO | NO_ZERO} {FIELDS | NO_FIELDS} {INDEX | NO_INDEX}
```

Transform Route Events: transforms the measures of events from one route reference to another and writes them to a new event table.

```
TransformRouteEvents <in_table> <in_event_properties> <in_routes>
  <route_id_field> <target_routes> <target_route_id_field> <out_table>
  <out_event_properties> <cluster_tolerance> {FIELDS | NO_FIELDS}
```

Spatial Analyst Toolbox

This toolbox provides tools for performing cell-based (raster) spatial analysis.

Conditional Toolset

Contains tools to control the output values based on the conditions placed on the input values.

Con: performs one or more conditional if/else evaluations on a cell-by-cell basis.

```
Con <in_conditional_raster> <in_true_raster_or_constant> <out_raster>
  {in_false_raster_or_constant} {where_clause}
```

Pick: uses the values of the cell to determine which expression in a list will be used to compute the output raster dataset.

```
Pick <in_position_raster> <in_rasters_or_constants;
  in_rasters_or_constants...> <out_raster>
```

Set Null: defines the null cells within a raster dataset.

```
SetNull <in_conditional_raster> <in_false_raster_or_constant> <out_raster>
  {where_clause}
```

Density Toolset

Contains tools used to calculate the spread of values over a surface.

Kernel Density: calculates a magnitude per unit area from point or polyline feature using a kernel function to fit a smoothly tapered surface to each point or polyline.

```
KernelDensity <in_features> <population_field> <out_raster> {cell_size}
  {search_radius} {SQUARE_MAP_UNITS | SQUARE_MILES | SQUARE_KILOMETERS | ACRES |
  HECTARES | SQUARE_YARDS | SQUARE_FEET | SQUARE_INCHES | SQUARE_METERS |
  SQUARE_CENTIMETERS | SQUARE_MILLIMETERS}
```

Line Density: calculates a magnitude per unit area from polyline features that fall within a radius around each cell.

```
LineDensity <in_polyline_features> <population_field> <out_raster> {cell_size}
  {search_radius} {SQUARE_MAP_UNITS | SQUARE_MILES | SQUARE_KILOMETERS | ACRES |
  HECTARES | SQUARE_YARDS | SQUARE_FEET | SQUARE_INCHES | SQUARE_METERS |
  SQUARE_CENTIMETERS | SQUARE_MILLIMETERS}
```

Point Density: calculates a magnitude per unit area from point features that fall within a neighborhood around each cell.

```
PointDensity <in_point_features> <population_field> <out_raster> {cell_size}
  {neighborhood} {SQUARE_MAP_UNITS | SQUARE_MILES | SQUARE_KILOMETERS | ACRES |
  HECTARES | SQUARE_YARDS | SQUARE_FEET | SQUARE_INCHES | SQUARE_METERS |
  SQUARE_CENTIMETERS | SQUARE_MILLIMETERS}
```

Distance Toolset

Contains tools used to compute distance across a raster dataset with respect to cost or along a path.

Corridor: computes the sum of two accumulative cost raster datasets.

```
Corridor <in_distance_raster1> <in_distance_raster2> <out_raster>
```

Cost Allocation: identifies for each cell the zone of each source cell that could be reached with the least accumulative cost.

```
CostAllocation <in_source_data> <in_cost_raster> <out_allocation_raster>
  {maximum_distance} {in_value_raster} {source_field} {out_distance_raster}
  {out_backlink_raster}
```

Cost Back Link: defines the neighbor cell on the least accumulative cost path from a cell to a set of source cells.

```
CostBackLink <in_source_data> <in_cost_raster> <out_backlink_raster>
  {maximum_distance} {out_distance_raster}
```

Cost Distance: calculates the least accumulative cost distance over a cost surface.

```
CostDistance <in_source_data> <in_cost_raster> <out_distance_raster>
  {maximum_distance} {out_backlink_raster}
```

Cost Path: calculates the least cost paths from a source to a destination over a surface.

```
CostPath <in_destination_data> <in_cost_distance_raster>
  <in_cost_backlink_raster> <out_raster>
  {EACH_CELL | EACH_ZONE | BEST_SINGLE} {destination_field}
```

Euclidean Allocation: assigns each cell the value of the sources to which it is closest.

```
EucAllocation <in_source_data> <out_allocation_raster> {maximum_distance}
  {in_value_raster} {cell_size} {source_field} {out_distance_raster}
  {out_direction_raster}
```

Euclidean Direction: computes for each cell the direction to the nearest source, measured in degrees.

```
EucDirection <in_source_data> <out_direction_raster> {maximum_distance}
  {cell_size} {out_distance_raster}
```

Euclidean Distance: calculates for each cell the straight-line distance from each cell to the closest source.

```
EucDistance <in_source_data> <out_distance_raster> {maximum_distance}
  {cell_size} {out_direction_raster}
```

Path Distance: calculates for each cell the least accumulative cost distance over a cost surface from a source cell or a set of source cells while accounting for surface distance and horizontal and vertical cost factors.

```
PathDistance <in_source_data> <out_distance_raster> {in_cost_raster}
  {in_surface_raster} {in_horizontal_raster} {horizontal_factor}
  {in_vertical_raster} {vertical_factor} {maximum_distance}
  {out_backlink_raster}
```

Path Distance Allocation: calculates for each cell its nearest source based on the least accumulative cost over a cost surface, while accounting for surface distance, horizontal cost factors, and vertical cost factors.

```
PathAllocation <in_source_data> <out_allocation_raster> {in_cost_raster}
  {in_surface_raster} {in_horizontal_raster} {horizontal_factor}
  {in_vertical_raster} {vertical_factor} {maximum_distance} {in_value_raster}
  {source_field} {out_distance_raster} {out_backlink_raster}
```

Path Distance Back Link: defines the neighbor that is the next cell on the least accumulative cost path to the nearest source, while accounting for surface distance, horizontal cost factors, and vertical cost factors.

```
PathBackLink <in_source_data> <out_backlink_raster> {in_cost_raster}
  {in_surface_raster} {in_horizontal_raster} {horizontal_factor}
  {in_vertical_raster} {vertical_factor} {maximum_distance}
  {out_distance_raster}
```

Extraction Toolset

Contains tools to extract a subset of cells either by the attributes or spatial location of each cell.

Extract by Attributes: extracts the cells of a raster dataset based on a logical query.

```
ExtractByAttributes <in_raster> <where_clause> <out_raster>
```

Extract by Circle: extracts the cell values of a raster dataset based on the boundaries of a circle.

```
ExtractByCircle <in_raster> <center_point> <radius> <out_raster>
  {INSIDE | OUTSIDE}
```

Extract by Mask: extracts the raster dataset cell values based on a mask dataset.

```
ExtractByMask <in_raster> <in_mask_data> <out_raster>
```

Extract by Points: extracts the cells of a raster dataset based on a set of points.

```
ExtractByPoints <in_raster> <points;points...> <out_raster>
  {INSIDE | OUTSIDE}
```

Extract by Polygon: extracts the cells of a raster dataset based on the boundaries within a polygon feature.

```
ExtractByPolygon <in_raster> <polygon;polygon...> <out_raster>
  {INSIDE | OUTSIDE}
```

Extract by Rectangle: extracts the cells of a raster dataset based on the boundaries of a rectangle.

```
ExtractByRectangle <in_raster> <rectangle> <out_raster> {INSIDE | OUTSIDE}
```

Extract Values to Points: extracts the cell values from a raster at the locations of points in a feature class.

```
ExtractValuesToPoints <in_point_features> <in_raster> <out_point_features>
  {NONE | INTERPOLATE} {VALUE_ONLY | ALL}
```

Sample: writes a sample of cell values from a group of rasters to a table.

```
Sample <in_rasters;in_rasters...> <in_location_data> <out_table>
  {NEAREST | BILINEAR | CUBIC}
```

Generalization Toolset

Contains tools to remove or reduce erroneous or irrelevant data within a raster through aggregation, edge smoothing, intelligent noise removal, and so forth.

Aggregate: generates a reduced resolution raster dataset.

```
Aggregate <in_raster> <out_raster> <cell_factor>
  {SUM | MAXIMUM | MEAN | MEDIAN | MINIMUM} {EXPAND | TRUNCATE}
  {DATA | NODATA}
```

Boundary Clean: smoothes the boundary between zones by expanding and shrinking the boundary.

```
BoundaryClean <in_raster> <out_raster> {NO_SORT | DESCEND | ASCEND}
  {TWO_WAY | ONE_WAY}
```

Expand: expands the selected zones of a raster dataset by a specified number of cells.

```
Expand <in_raster> <out_raster> <number_cells> <zone_values;zone_values...>
```

Majority Filter: replaces cell values within a raster dataset based upon the majority of their contiguous neighboring cells.

```
MajorityFilter <in_raster> <out_raster> {FOUR | EIGHT} {MAJORITY | HALF}
```

Nibble: replaces cell values in a raster dataset corresponding to a mask with the values of the nearest neighbors.

```
Nibble <in_raster> <in_mask_raster> <out_raster> {ALL_VALUES | DATA_ONLY}
```

Region Group: records for each cell of a raster the identity of the connected region to which it belongs. A unique number is assigned to each region.

```
RegionGroup <in_raster> <out_raster> {FOUR | EIGHT} {WITHIN | CROSS} {ADD_LINK |
  NO_LINK} {excluded_value}
```

Shrink: shrinks the selected zones by a specified number of cells.

```
Shrink <in_raster> <out_raster> <number_cells> <zone_values;zone_values...>
```

Thin: thins rasterized linear features in a raster dataset.

```
Thin <in_raster> <out_raster> {ZERO | NODATA} {NO_FILTER | FILTER}
  {ROUND | SHARP} {maximum_thickness}
```

Groundwater Toolset

Contains tools used to measure hydrodynamic movement within or along a surface.

Darcy Flow: calculates the groundwater volume balance residual and other outputs for steady flow in an aquifer.

```
DarcyFlow <in_head_raster> <in_porosity_raster> <in_thickness_raster>
  <in_transmissivity_raster> <out_volume_raster> {out_direction_raster}
  {out_magnitude_raster}
```

Darcy Velocity: calculates the groundwater seepage velocity vector (direction and magnitude) for steady flow in an aquifer.

```
DarcyVelocity <in_head_raster> <in_porosity_raster> <in_thickness_raster>
  <in_transmissivity_raster> <out_direction_raster> <out_magnitude_raster>
```

Particle Track: calculates the path of a particle through a velocity field.

```
ParticleTrack <in_direction_raster> <in_magnitude_raster> <source_point>
  <out_track_file> {step_length} {tracking_time} {out_track_polyline_features}
```

Porous Puff: calculates the hydrodynamic dispersion of an instantaneous point release of a constituent as it is advected along the flow path.

```
PorousPuff <in_track_file> <in_porosity_raster> <in_thickness_raster>
  <out_raster> <mass> {dispersion_time} {longitudinal_dispersivity}
  {dispersivity_ratio} {retardation_factor} {decay_coefficient}
```

Hydrology Toolset

Contains tools providing hydrology functions to simulate the flow of water over an elevation surface and creates either a stream network or a watershed.

Basin: creates a raster dataset delineating all drainage basins.

```
Basin <in_flow_direction_raster> <out_raster>
```

Fill: fills sinks in a surface raster to remove small imperfections in the data.

```
Fill <in_surface_raster> <out_surface_raster> {z_limit}
```

Flow Accumulation: creates a raster dataset of accumulated flow to each cell by accumulating the weight for all cells that flow into each downslope cell.

```
FlowAccumulation <in_flow_direction_raster> <out_accumulation_raster>
  {in_weight_raster}
```

Flow Direction: creates a raster dataset of flow direction from each cell to its steepest downslope neighbor.

```
FlowDirection <in_surface_raster> <out_flow_direction_raster>
  {NORMAL | FORCE} {out_drop_raster}
```

Flow Length: calculates upstream or downstream distance or weighted distance along a flow path for each cell.

```
FlowLength <in_flow_direction_raster> <out_raster> {DOWNSTREAM | UPSTREAM}
  {in_weight_raster}
```

Sink: creates a raster dataset identifying all sinks or areas of internal drainage.

```
Sink <in_flow_direction_raster> <out_raster>
```

Snap Pour Point: snaps pour points to the cell of highest flow accumulation within a specified distance.

```
SnapPourPoint <in_pour_point_data> <in_accumulation_raster> <out_raster>
  <snap_distance> {pour_point_field}
```

Stream Link: assigns unique values to sections of a raster linear network between intersections.

```
StreamLink <in_stream_raster> <in_flow_direction_raster> <out_raster>
```

Stream Order: assigns a numeric order to segments of a raster dataset representing branches of a linear network.

```
StreamOrder <in_stream_raster> <in_flow_direction_raster> <out_raster>
  {STRAHLER | SHREVE}
```

Stream to Feature: converts a raster dataset representing a linear network to features representing the linear network.

```
StreamToFeature <in_stream_raster> <in_flow_direction_raster>
  <out_polyline_features> {SIMPLIFY| NO_SIMPLIFY}
```

Watershed: determines the contributing area above a set of cells in a raster dataset.

```
Watershed <in_flow_direction_raster> <in_pour_point_data> <out_raster>
  {pour_point_field}
```

Interpolation Toolset

Contains tools to create a raster surface from point features.

IDW: interpolates a surface from points using an inverse distance weighted (IDW) technique.

```
IDW <in_point_features> <z_field> <out_raster> {cell_size} {power}
  {search_radius} {in_barrier_polyline_features}
```

Krige: interpolates a raster dataset from a set of points using kriging.

```
Kriging_sa <in_point_features> <z_field> <out_surface_raster>
  <semiVariogram_props> {cell_size} {search_radius}
  {out_variance_prediction_raster}
```

Natural Neighbor: interpolates a surface from points using a natural neighbor technique.

```
NaturalNeighbor <in_point_features> <z_field> <out_raster> {cell_size}
```

Spline: interpolates a surface from points using a minimum curvature spline technique.

```
Spline <in_point_features> <z_field> <out_raster> {cell_size}
  {REGULARIZED | TENSION} {weight} {number_points}
```

Topo to Raster: generates a hydrologically correct raster dataset of elevation.

```
TopoToRaster <feature_layer{Field} {Type};feature_layer{Field} {Type}...>
  <out_surface_raster> {cell_size} {extent} {Margin} {minimum_z_value}
  {maximum_z_value} {ENFORCE | NO_ENFORCE | ENFORCE_WITH_SINK}
  {CONTOUR | SPOT} {maximum_iterations} {roughness_penalty}
  {discrete_error_factor} {vertical_standard_error} {tolerance_1}
  {tolerance_2} {out_stream_features} {out_sink_features}
  {out_diagnostic_file} {out_parameter_file}
```

Topo to Raster by File: generates a hydrologically correct raster dataset of elevation.

```
TopoToRasterByFile <in_parameter_file> <out_surface raster>
  {out_stream_features} {out_sink_features}
```

Trend: interpolates a surface from points using a trend technique.

```
Trend <in_point_features> <z_field> <out_raster> {cell_size} {order}
  {LINEAR | LOGISTIC}
```

Local Toolset

Contains tools to compute an output raster dataset where the output value at each location is a function of the value associated with that location on one or more raster datasets.

Cell Statistics: calculates a per cell statistic from multiple raster datasets.

```
CellStatistics <in_rasters_or_constants;in_rasters_or_constants...>
  <out_raster> {MEAN | MAJORITY | MAXIMUM | MEDIAN | MINIMUM | MINORITY | RANGE |
  STD | SUM | VARIETY}
```

Combine: combines multiple raster datasets on a cell-by-cell basis.

```
Combine <in_rasters;in_rasters...> <out_raster>
```

Equal to Frequency: evaluates the number of times the input raster dataset values are equal to a specified value on a cell-by-cell basis.

```
EqualToFrequency <in_value_raster> <in_rasters;in_rasters...> <out_raster>
```

Greater Than Frequency: evaluates the number of times the input raster dataset values are greater than a specified value on a cell-by-cell basis.

```
GreaterThanFrequency <in_value_raster> <in_rasters;in_rasters...> <out_raster>
```

Highest Position: determines the position of a raster dataset with the maximum value in a set of raster datasets.

```
HighestPosition <in_rasters_or_constants;in_rasters_or_constants...>
  <out_raster>
```

Less Than Frequency: evaluates the number of times the input raster dataset values are less than a specified value on a cell-by-cell basis.

```
LessThanFrequency <in_value_raster> <in_rasters;in_rasters...> <out_raster>
```

Lowest Position: determines the position for each cell of the input raster dataset with the minimum value in the argument list.

```
LowestPosition <in_rasters_or_constants;in_rasters_or_constants...>
  <out_raster>
```

Popularity: determines the value that is at a specified level of popularity on a cell-by-cell basis.

```
Popularity <in_popularity_raster_or_constant> <in_rasters;in_rasters...>
  <out_raster>
```

Rank: returns the value of a set of raster datasets based on a rank level specified by another raster dataset on a cell-by-cell basis.

```
Rank <in_rank_raster_or_constant> <in_rasters;in_rasters...> <out_raster>
```

Map Algebra Toolset

Contains the tool to create expressions using any of the Map Algebra statements. Map Algebra is the analysis language from Spatial Analyst.

Multi Output Map Algebra: executes GRID's map algebra statements.

```
MultiOutputMapAlgebra <expression_string>
```

Single Output Map Algebra: executes GRID's map algebra statement to produce a raster dataset.

```
SingleOutputMapAlgebra <expression_string> <out_raster> {in_data;in_data...}
```

Math Toolset

Contains tools to implement math functions, which apply a specified mathematical operation or function to each cell location on an input raster or series of raster datasets.

Abs: calculates the absolute value of the input raster dataset on a cell-by-cell basis.

```
Abs <in_raster_or_constant> <out_raster>
```

Divide: divides the values of two input raster datasets on a cell-by-cell basis.

```
Divide <in_raster_or_constant1> <in_raster_or_constant2> <out_raster>
```

Exp: calculates the base e exponential of cells in a raster dataset.

```
Exp <in_raster_or_constant> <out_raster>
```

Exp10: calculates the base 10 exponential of cells in a raster dataset.

```
Exp10 <in_raster_or_constant> <out_raster>
```

Exp2: calculates the base 2 exponential of cells in a raster dataset.

```
Exp2 <in_raster_or_constant> <out_raster>
```

Float: converts each cell value in a raster dataset to floating-point values.

```
Float <in_raster_or_constant> <out_raster>
```

Int: converts each cell value in a raster dataset to an integer by truncation.

```
Int <in_raster_or_constant> <out_raster>
```

Ln: calculates the natural logarithm (base e) of cells in a raster dataset.

```
Ln <in_raster_or_constant> <out_raster>
```

Log10: calculates the base 10 logarithm of cells in a raster dataset.

```
Log10 <in_raster_or_constant> <out_raster>
```

Log2: calculates the base 2 logarithm of cells in a raster dataset.

```
Log2 <in_raster_or_constant> <out_raster>
```

Minus: subtracts the values of the second input from the values of the first input on a cell-by-cell basis.

```
Minus <in_raster_or_constant1> <in_raster_or_constant2> <out_raster>
```

Mod: divides the values of the first input by the values of the second input and returns the remainder.

```
Mod <in_raster_or_constant1> <in_raster_or_constant2> <out_raster>
```

Negate: changes the sign of the cell values of the input raster dataset (multiplies by -1).

```
Negate <in_raster_or_constant> <out_raster>
```

Plus: adds the values of two raster datasets on a cell-by-cell basis.

```
Plus <in_raster_or_constant1> <in_raster_or_constant2> <out_raster>
```

Power: calculates the “nth” power of the input raster or number on a cell-by-cell basis.

```
Power <in_raster_or_constant1> <in_raster_or_constant2> <out_raster>
```

Round Down: returns the next lower whole number value for each cell in a raster dataset.

```
RoundDown <in_raster_or_constant> <out_raster>
```

Round Up: returns the next highest whole number value that is greater than or equal to the input value for each cell in a raster dataset.

```
RoundUp <in_raster_or_constant> <out_raster>
```

Square: calculates the square of cell values in a raster dataset.

```
Square <in_raster_or_constant> <out_raster>
```

Square Root: calculates the square root of the input grid or number for each cell.

```
SquareRoot <in_raster_or_constant> <out_raster>
```

Times: multiplies the values of two raster datasets on a cell-by-cell basis.

```
Times <in_raster_or_constant1> <in_raster_or_constant2> <out_raster>
```

Bitwise (Math) Toolset

Contains tools to implement bitwise operators, which treat the operands as bits (binary representations) and calculate the output by applying the logical operation (e.g., 3 BITWISEAND 5 = 0011 && 0101 = 0001 = 1).

Bitwise And: performs the bitwise AND operation on the binary values of two inputs on a cell-by-cell basis (e.g., 0011 && 0101 = 0001).

```
BitwiseAnd <in_raster_or_constant1> <in_raster_or_constant2> <out_raster>
```

Bitwise Left Shift: shifts the bits to the left using the number specified (e.g., 1<<2 = 0001 << 2 = 0100 = 4).

```
BitwiseLeftShift <in_raster_or_constant1> <in_raster_or_constant2>
  <out_raster>
```

Bitwise Not: performs the bitwise COMPLEMENT operation on the binary values of two inputs on a cell-by-cell basis (flips the bits; e.g., 5 = 0101 ~ 1010 = 10).

```
BitwiseNot <in_raster_or_constant> <out_raster>
```

Bitwise Or: performs a bitwise OR operation on the binary values of two inputs on a cell-by-cell basis (e.g., 0101 || 1100 = 1101).

```
BitwiseOr <in_raster_or_constant1> <in_raster_or_constant2> <out_raster>
```

Bitwise Right Shift: shifts the bits to the right using the number specified (e.g., 6>>1 = 0110 >> 1 = 0011 = 3).

```
BitwiseRightShift <in_raster_or_constant1> <in_raster_or_constant2>
  <out_raster>
```

Bitwise XOr: performs a bitwise exclusive OR operation on the binary values of two inputs on a cell-by-cell basis (e.g., 0101 !! 1100 = 1001).

```
BitwiseXOr <in_raster_or_constant1> <in_raster_or_constant2> <out_raster>
```

Logical (Math) Toolset

Contains tools to evaluate the values of an input raster or rasters relative to a conditional statement, the values in another raster, a constant value, or a specific value. Also contains tools that can produce an output that tracks the unique combinations of the input values between two rasters or constants.

Boolean And: performs the Boolean AND operator on the cell values of two input raster datasets.

```
BooleanAnd <in_raster_or_constant1> <in_raster_or_constant2> <out_raster>
```

Boolean Not: performs the Boolean COMPLEMENT operator on the cell values of two input raster datasets.

```
BooleanNot <in_raster_or_constant> <out_raster>
```

Boolean Or: performs the Boolean OR operator on the cell values of two input raster datasets.

```
BooleanOr <in_raster_or_constant1> <in_raster_or_constant2> <out_raster>
```

Boolean XOr: performs the Boolean exclusive OR operator on the cell values of two input raster datasets.

```
BooleanXOr <in_raster_or_constant1> <in_raster_or_constant2> <out_raster>
```

Combinatorial And: performs a combinatorial AND operation on two input raster datasets.

```
CombinatorialAnd <in_raster_or_constant1> <in_raster_or_constant2>
  <out_raster>
```

Combinatorial Or: performs a combinatorial OR operation on the cell values of two input raster datasets.

```
CombinatorialOr <in_raster_or_constant1> <in_raster_or_constant2> <out_raster>
```

Combinatorial XOr: performs a combinatorial exclusive OR operation on the cell values of two input raster datasets.

```
CombinatorialXOr <in_raster_or_constant1> <in_raster_or_constant2>
  <out_raster>
```

Equal To: returns 1 for cells where the first raster equals the second raster and 0 if it does not.

```
EqualTo <in_raster_or_constant1> <in_raster_or_constant2> <out_raster>
```

Greater Than: returns 1 for cells where the first raster dataset is greater than the second raster dataset and returns 0 where it is not.

```
GreaterThan <in_raster_or_constant1> <in_raster_or_constant2> <out_raster>
```

Greater Than Equal: returns 1 for cells where the first raster dataset is greater than or equal to the second raster dataset and returns 0 where it is not.

```
GreaterThanEqual <in_raster_or_constant1> <in_raster_or_constant2>
  <out_raster>
```

Is Null: returns 1 for cells in the input raster dataset that have a value of NoData and returns 0 where they do not.

```
IsNull <in_raster> <out_raster>
```

Less Than: returns 1 for cells where the first raster dataset is less than the second raster dataset and returns 0 where it is not.

```
LessThan <in_raster_or_constant1> <in_raster_or_constant2> <out_raster>
```

Less Than Equal: returns 1 for cells where the first raster dataset is less than or equal to the second raster dataset and returns 0 where it is not.

```
LessThanEqual <in_raster_or_constant1> <in_raster_or_constant2> <out_raster>
```

Not Equal: returns 1 for cells where the first raster dataset is not equal to the second raster dataset and returns 0 where it is not.

```
NotEqual <in_raster_or_constant1> <in_raster_or_constant2> <out_raster>
```

Test: returns 1 for cells that evaluate to true based on a logical expression and returns 0 for cells that evaluate to false.

```
Test <in_raster> <where_clause> <out_raster>
```

Trigonometric (Math) Toolset

Contains tools for the trigonometric functions that are applied on a per cell basis to an input raster dataset.

ACos: calculates the inverse cosine of the input raster dataset or number on a cell-by-cell basis.

```
ACos <in_raster_or_constant> <out_raster>
```

ACosH: calculates the inverse hyperbolic cosine of cells in an input raster dataset.

```
ACosH <in_raster_or_constant> <out_raster>
```

ASin: calculates the inverse sine of cells in an input raster dataset.

```
ASin <in_raster_or_constant> <out_raster>
```

ASinH: calculates the inverse hyperbolic sine of cells in an input raster dataset.

```
ASinH <in_raster_or_constant> <out_raster>
```

ATan: calculates the inverse tangent of cells in an input raster dataset.

```
ATan <in_raster_or_constant> <out_raster>
```

ATan2: calculates the inverse tangent (based on y/x) of cells in an input raster dataset.

```
ATan2 <in_raster_or_constant1> <in_raster_or_constant2> <out_raster>
```

ATanH: calculates the inverse hyperbolic tangent of cells in an input raster dataset.

```
ATanH <in_raster_or_constant> <out_raster>
```

Cos: calculates the cosine of cells in an input raster dataset.

```
Cos <in_raster_or_constant> <out_raster>
```

CosH: calculates the hyperbolic cosine of cells in an input raster dataset.

```
CosH <in_raster_or_constant> <out_raster>
```

Sin: calculates the sine of cells in an input raster dataset.

```
Sin <in_raster_or_constant> <out_raster>
```

SinH: calculates the hyperbolic sine of cells in an input raster dataset.

```
SinH <in_raster_or_constant> <out_raster>
```

Tan: calculates the tangent of cells in an input raster dataset.

```
Tan <in_raster_or_constant> <out_raster>
```

TanH: calculates the hyperbolic tangent of cells in an input raster dataset.

```
TanH <in_raster_or_constant> <out_raster>
```

Multivariate Toolset

Contains tools to allow the statistical analysis of a series of raster datasets (independent variables) that, based on the values within the rasters, produces a predictable result for some phenomena (the dependent variable).

Band Collection Statistics: calculates the statistics for a set of raster bands.

```
BandCollectionStats <in_raster_bands;in_raster_bands...> <out_stat_file>
  {BRIEF | DETAILED}
```

Class Probability: creates probability layers for each class in a signature file.

```
ClassProbability <in_raster_bands;in_raster_bands...> <in_signature_file>
  <out_multiband_raster> {maximum_output_value} {EQUAL | SAMPLE | FILE}
  {in_a_priori_file}
```

Create Signatures: creates an ASCII signature file of classes defined by input sample data and a set of raster bands.

```
CreateSignatures <in_raster_bands;in_raster_bands...> <in_sample_data>
  <out_signature_file> {COVARIANCE | MEAN_ONLY} {sample_field}
```

Dendrogram: constructs a tree diagram showing attribute distances between sequentially merged classes in a signature file.

```
Dendrogram <in_signature_file> <out_dendrogram_file> {VARIANCE | MEAN_ONLY}
  {line_width}
```

Edit Signatures: edits and updates a signature file by merging, renumbering, and deleting class signatures.

```
EditSignatures <in_raster_bands;in_raster_bands...> <in_signature_file>
  <in_signature_remap_file> <out_signature_file> {sample_interval}
```

Iso Cluster: uses isodata clustering to determine the characteristics of natural cell groupings in multidimension attribute space.

```
IsoCluster <in_raster_bands;in_raster_bands...> <out_signature_file>
  <number_classes> {number_iterations} {min_class_size} {sample_interval}
```

Maximum Likelihood Classification: performs a maximum likelihood classification on a set of raster dataset bands.

```
MLClassify <in_raster_bands;in_raster_bands...> <in_signature_file>
  <out_classified_raster> {0.0 | 0.005 | 0.01 | 0.025 | 0.05 | 0.1 | 0.25 | 0.5 |
  0.75 | 0.9 | 0.95 | 0.975 | 0.99 | 0.995} {EQUAL | SAMPLE | FILE}
  {in_a_priori_file} {out_confidence_raster}
```

Principal Components: performs principal components analysis on a set of raster bands.

```
PrincipalComponents <in_raster_bands;in_raster_bands...>
  <out_multiband_raster> {number_components} {out_data_file}
```

Neighborhood Toolset

Contains tools to calculate a statistic or value based on the values at each processing cell and the values of the cells within an identified neighborhood.

Block Statistics: calculates statistics for a nonoverlapping neighborhood.

```
BlockStatistics <in_raster> <out_raster> {neighborhood} {MEAN | MAJORITY |
  MAXIMUM | MEDIAN | MINIMUM | MINORITY | RANGE | STD | SUM | VARIETY}
  {DATA | NODATA}
```

Filter: performs a preset focal filter on a raster.

```
Filter <in_raster> <out_raster> {LOW | HIGH} {DATA | NODATA}
```

Focal Flow: determines the flow of the values in the surface raster dataset within each cell's immediate neighborhood.

```
FocalFlow <in_surface_raster> <out_raster> {threshold_value}
```

Focal Statistics: calculates a statistic for each raster dataset cell value within a specified neighborhood.

```
FocalStatistics <in_raster> <out_raster> {neighborhood} {MEAN | MAJORITY |
  MAXIMUM | MEDIAN | MINIMUM | MINORITY | RANGE | STD | SUM | VARIETY}
  {DATA | NODATA}
```

Line Statistics: calculates a statistic on the attributes of lines in a circular neighborhood around each output cell in a raster dataset.

```
LineStatistics <in_polyline_features> <field> <out_raster> {cell_size}
  {search_radius} {MEAN | MAJORITY | MAXIMUM | MEDIAN | MINIMUM | MINORITY | RANGE
  | STD | SUM | VARIETY}
```

Point Statistics: calculates a statistic on the points in a neighborhood outputting a raster dataset.

```
PointStatistics <in_point_features> <field> <out_raster> {cell_size}
  {neighborhood} {MEAN | MAJORITY | MAXIMUM | MEDIAN | MINIMUM | MINORITY | RANGE |
  STD | SUM | VARIETY}
```

Overlay Toolset

Contains the tool to create an output surface by adding a series of weighted input raster datasets together.

Weighted Overlay: overlays several rasters using a common scale and weighing each according to its importance.

```
WeightedOverlay <raster{influence} {field} {remap};raster{influence} {field}
  {remap}...> <out_raster>
```

Raster Creation Toolset

Contains tools to create raster datasets based on a constant, random values, or a normal distribution using the existing cell size, extent, and other analysis properties.

Create Constant Raster: creates a raster dataset from a constant value.

```
CreateConstantRaster <out_raster> <constant_value> {INTEGER | FLOAT}
  {cell_size} {extent}
```

Create Normal Raster: creates a raster dataset of random values from a normal distribution.

```
CreateNormalRaster <out_raster> {cell_size} {extent}
```

Create Random Raster: creates a raster dataset of random numbers between 0 and 1.

```
CreateRandomRaster <out_raster> {seed_value} {cell_size} {extent}
```

Reclass Toolset

Contains tools to change the values assigned to cells in a thematic raster dataset.

Lookup: creates a new raster dataset by looking up values found in another field in the table of the input raster dataset.

```
Lookup <in_raster> <lookup_field> <out_raster>
```

Reclass by ASCII File: reclassifies (or changes) the values of the input cells of a raster dataset by using an ASCII remap file.

```
ReclassByASCIIFile <in_raster> <in_remap_file> <out_raster> {DATA | NODATA}
```

Reclass by Table: reclassifies (or changes) the values of the input cells of a raster dataset by using a remap table.

```
ReclassByTable <in_raster> <in_remap_table> <from_value_field>
  <to_value_field> <output_value_field> <out_raster> {DATA | NODATA}
```

Reclassify: reclassifies (or changes) the value of the cells in a raster dataset.

```
Reclassify <in_raster> <reclass_field> <remap> <out_raster> {DATA | NODATA}
```

Slice: slices a range of values of the input cells by zones of equal area or equal interval.

```
Slice <in_raster> <out_raster> <number_zones> {EQUAL_INTERVAL | EQUAL_AREA}
  {base_output_zone}
```

Surface Toolset

Contains tools to analyze the surface of the shapes represented by the raster values.

Aspect: identifies the direction of the maximum rate of change in z-value from each cell.

```
Aspect <in_raster> <out_raster>
```

Contour: creates contours or isolines from a raster dataset surface.

```
Contour <in_raster> <out_polyline_features> <contour_interval> {base_contour}
  {z_factor}
```

Contour List: creates contours or isolines based on a list of contour values.

```
ContourList <in_raster> <out_polyline_features>
  <contour_values;contour_values...>
```

Curvature: calculates the curvature of a surface at each cell center.

```
Curvature <in_raster> <out_curvature_raster> {z_factor}
  {out_profile_curve_raster} {out_plan_curve_raster}
```

Cut/Fill: calculates cut and fill areas.

```
CutFill <in_before_surface> <in_after_surface> <out_raster> {z_factor}
```

Hillshade: creates a shaded relief raster by considering the illumination angle and shadows.

```
HillShade <in_raster> <out_raster> {azimuth} {altitude}
  {NO_SHADOWS | SHADOWS} {z_factor}
```

Observer Points: identifies exactly which observer points are visible from each surface location.

```
ObserverPoints <in_raster> <in_observer_point_features> <out_raster>
  {z_factor} {FLAT_EARTH | CURVED_EARTH} {refractivity_coefficient}
```

Slope: identifies the rate of maximum change in z-value from each cell.

```
Slope <in_raster> <out_raster> {DEGREE | PERCENT_RISE} {z_factor}
```

Viewshed: derives the raster dataset surface locations visible to a set of observation points.

```
Viewshed <in_raster> <in_observer_features> <out_raster> {z_factor} {FLAT_EARTH
  | CURVED_EARTH} {refractivity_coefficient}
```

Zonal Toolset

Contains tools that can be applied to the zones within the raster dataset.

Tabulate Area: calculates cross-tabulated areas between two datasets.

```
TabulateArea <in_zone_data> <zone_field> <in_class_data> <class_field>
  <out_table> {processing_cell_size}
```

Zonal Fill: fills zones in a dataset using the minimum cell value from another raster dataset along the zone boundary.

```
ZonalFill <in_zone_raster> <in_weight_raster> <out_raster>
```

Zonal Geometry: calculates area, perimeter, thickness, or the characteristics of ellipse and records the information as a raster dataset.

```
ZonalGeometry <in_zone_data> <zone_field> <out_raster>
  {AREA | PERIMETER | THICKNESS | CENTROID} {cell_size}
```

Zonal Geometry as Table: calculates area, perimeter, thickness, or the characteristics of ellipse and records the information in an output table.

```
ZonalGeometryAsTable <in_zone_data> <zone_field> <out_table>
  {processing_cell_size}
```

Zonal Statistics: calculates a statistic on values of a raster within the zones of another dataset.

```
ZonalStatistics <in_zone_data> <zone_field> <in_value_raster> <out_raster>
  {MEAN | MAJORITY | MAXIMUM | MEDIAN | MINIMUM | MINORITY | RANGE | STD | SUM |
  VARIETY} {DATA | NODATA}
```

Zonal Statistics as Table: summarizes values of a raster dataset within the zones of another dataset and reports the results to a table.

```
ZonalStatisticsAsTable <in_zone_data> <zone_field> <in_value_raster>
  <out_table> {DATA | NODATA}
```

Spatial Statistics Toolbox

Contains statistical tools for analyzing the distribution of geographic features.

Analyzing Patterns Toolset

Contains tools to calculate statistical values used to quantify patterns.

❖**Average Nearest Neighbor:** calculates a nearest neighbor index based on the average distance from each feature to its nearest neighboring feature.

```
AverageNearestNeighbor <input_feature_class> <EUCLIDEAN DISTANCE | MANHATTAN
  DISTANCE> {FALSE | TRUE} {area}
```

❖**High/Low Clustering (Getis–Ord General G):** measures concentrations of high or low values for a study area.

```
HighLowClustering <input_feature_class> <input_field> {FALSE | TRUE} <INVERSE
  DISTANCE | INVERSE DISTANCE SQUARED | FIXED DISTANCE BAND | ZONE OF INDIFFERENCE
  | GET SPATIAL WEIGHTS FROM FILE> <EUCLIDEAN DISTANCE | MANHATTAN DISTANCE> <NONE
  | ROW | GLOBAL> <distance_band_or_threshold_distance> {weights_matrix_file}
```

❖**Spatial Autocorrelation (Morans I):** measures spatial autocorrelation based on feature locations and attribute values.

```
SpatialAutocorrelation <input_feature_class> <input_field> {FALSE | TRUE}
  <INVERSE DISTANCE | INVERSE DISTANCE SQUARED | FIXED DISTANCE BAND  | ZONE OF
  INDIFFERENCE | GET SPATIAL WEIGHTS FROM FILE> <EUCLIDEAN DISTANCE | MANHATTAN
  DISTANCE> <NONE | ROW | GLOBAL> <distance_band_or_threshold_distance>
  {weights_matrix_file}
```

Mapping Clusters Toolset

Contains tools for cluster analysis, such as identifying the locations of statistically significant hot spots or areas of significant diversity.

❖**Cluster and Outlier Analysis (Anselin Local Morans I):** identifies those clusters of points with values similar in magnitude and clusters of points with very heterogeneous values, within a set of weighted data points.

```
ClustersOutliers <input_feature_class> <input_field> <output_feature_class>
  <INVERSE DISTANCE | INVERSE DISTANCE SQUARED | FIXED DISTANCE BAND | ZONE OF
  INDIFFERENCE | GET SPATIAL WEIGHTS FROM FILE> <EUCLIDEAN DISTANCE | MANHATTAN
  DISTANCE> <NONE | ROW | GLOBAL> <distance_band_or_threshold_distance>
  {weights_matrix_file}
```

❖**Cluster/Outlier Analysis with Rendering:** identifies clusters of points with values similar in magnitude and clusters of points with very heterogeneous values, within a given set of weighted data points and applies a “cold to hot” type of rendering.

```
ClustersOutliersRendered <input_feature_class> <input_field>
  <output_layer_file> <output_feature_class>
```

Hot Spot Analysis (Getis–Ord Gi*): identifies spatial clusters of statistically significant high or low attribute values by calculating the Getis–Ord Gi* statistics.

```
HotSpot <input_feature_class> <input_field> <output_feature_class> <INVERSE
  DISTANCE | INVERSE DISTANCE SQUARED | FIXED DISTANCE BAND | ZONE OF INDIFFERENCE
  | GET SPATIAL WEIGHTS FROM FILE> <EUCLIDEAN DISTANCE | MANHATTAN DISTANCE> <NONE
  | ROW | GLOBAL> <distance_band_or_threshold_distance> {self_potential_field}
  {weights_matrix_file}
```

Hot Spot Analysis with Rendering: calculates Gi* statistics and applies a "cold to hot" type of rendering to the output Z scores.

```
HotSpotsRendered <input_feature_class> <input_field> <output_feature_class>
  <output_layer_file>
```

Measuring Geographic Distributions Toolset

Contains tools to calculate a value that represents a characteristic of the distribution of a set of features, such as the center, compactness, or orientation.

Central Feature: identifies the most centrally located feature in a point, line, or polygon feature.

```
CentralFeature <input_feature_class> <output_feature_class> <EUCLIDEAN
  DISTANCE | MANHATTAN DISTANCE> {weight_field} {self_potential_weight_field}
```

Directional Distribution (Standard Deviational Ellipse): measures whether a distribution of features exhibits a directional trend.

```
DirectionalDistribution <input_feature_class> <output_ellipse_feature_class>
  <1 STANDARD DEVIATION | 2 STANDARD DEVIATIONS | 3 STANDARD DEVIATIONS>
  {weight_field} {case_field}
```

Linear Directional Mean: identifies the general (mean) direction for a set of vectors.

```
DirectionalMean <input_feature_class> <output_feature_class>
  <orientation_only> {case_field}
```

Mean Center: identifies the geographic center (or the center of concentration) for a set of features.

```
MeanCenter <input_feature_class> <output_feature_class> {weight_field}
  {case_field} {dimension_field}
```

Standard Distance: measures the degree to which features are concentrated or dispersed around the points (or feature centroids) in an input feature class.

```
StandardDistance <input_feature_class>
  <ouput_standard_distance_feature_class> <1 STANDARD DEVIATION | 2 STANDARD
  DEVIATIONS | 3 STANDARD DEVIATIONS> {weight_field} {case_field}
```

Utilities Toolset

Contains tools to perform a variety of data rendering tasks that can be used in conjunction with other tools in the Spatial Statistics toolbox.

❖**Calculate Areas:** Calculates AREA values for each feature in a polygon feature class.

```
CalculateAreas <input_feature_class> <output_feature_class>
```

❖**Collect Events:** collects event data into weighted point data.

```
CollectEvents <input_incident_features> <output_weighted_point_feature_class>
```

❖**Collect Events with Rendering:** collects event data into weighted point data and applies a graduated circle rendering to the count field.

```
CollectEventsRendered <input_incident_features> <output_layer_file>
  <output_weighted_point_feature_class>
```

❖**Count Rendering:** applies graduated circle rendering to a count type field of a point feature class.

```
CountRenderer <input_feature_class> <field_to_render> <output_layer_file>
  <number_of_classes> <MANGO | BRIGHT_RED | DARK_GREEN | GREEN | DARK_BLUE |
  BRIGHT_PINK | LIGHT_YELLOW| SKY_BLUE> {maximum_field_value}
```

❖**Export Feature Attribute to ASCII:** Exports feature class coordinates and attribute values to a space, comma, or semicolon delimited ASCII text file.

```
ExportXYv <input_feature_class> <value_field> <SPACE | COMMA | SEMI-COLON>
  <output_ascii_file>
```

❖**Z Score Rendering:** applies a “cold or hot” graduated color rendering to a field of Z scores.

```
ZRenderer <input_feature_class> <field_to_render> <output_layer_file>
```

3D Analyst Toolbox

Contains tools to create and modify TIN and raster surfaces, then extract information and features from them.

Conversion Toolset

Contains tools used to convert to and from a TIN.

Raster to TIN: creates a TIN from a raster dataset.

```
RasterTin <in_raster> <out_tin> {z_tolerance} {max_points} {z_factor}
```

TIN Domain: extracts the interpolation zone from an input TIN into an output feature class.

```
TinDomain <in_tin> <out_feature_class> <LINE | POLYGON>
```

TIN Edge: extracts the triangle edge from an input TIN into an output feature class.

```
TinEdge <in_tin> <out_feature_class> {DATA | SOFT | HARD | ENFORCED | REGULAR |
  OUTSIDE | ALL}
```

TIN Node: extracts nodes from an input TIN into an output feature class.

```
TinNode <in_tin> <out_feature_class> {spot_field} {tag_field}
```

TIN Polygon Tag: extracts polygon tag information from an input TIN into an output feature class.

```
TinPolygonTag <in_tin> <out_feature_class> {tag_field}
```

TIN to Raster: converts a TIN to a raster.

```
TinRaster <in_tin> <out_raster> {FLOAT | INT} {LINEAR | NATURAL_NEIGHBORS}
  {OBSERVATIONS | CELLSIZE} {z_factor}
```

TIN Triangle: extracts triangles as polygons from an input TIN into an output feature class.

```
TinTriangle <in_tin> <out_feature_class> {PERCENT | DEGREE} {z_factor}
  {HILLSHADE} {tag_field}
```

Functional Surface Toolset

Contains tools to produce output providing knowledge about height information that is contained in surfaces.

Interpolate Shape: interpolates z-values for a feature class based on an underlying surface.

```
InterpolateShape <in_surface> <in_feature_class> <out_feature_class>
  {sample_distance} {z_factor}
```

Line of Sight: calculates the visibility across a surface between points.

```
LineOfSight <in_surface> <in_line_feature_class> <out_los_feature_class>
  {out_obstruction_feature_class} {use_curvature} {use_refraction}
  {refraction_factor}
```

Surface Length: calculates the surface length of each line in a feature class based on a functional surface.

```
SurfaceLength <in_surface> <in_feature_class> {out_length_field}
  {sample_distance} {z_factor}
```

Surface Spot: calculates surface values for each point of a point feature class by interpolating from a functional surface.

```
SurfaceSpot <in_surface> <in_feature_class> {out_spot_field} {z_factor}
```

Surface Volume: calculates the area and volume of a functional surface above or below a given reference plane.

```
SurfaceVolume <in_surface> {out_text_file} {ABOVE | BELOW} {base_z} {z_factor}
```

Raster Interpolation Toolset

Contains tools to create a raster surface from point features.

IDW: interpolates a surface from points using an inverse distance weighted technique.

```
IDW <in_point_features> <z_field> <out_raster> {cell_size} {power}
  {search_radius} {in_barrier_polyline_features}
```

Krige: interpolates a raster dataset from a set of points using kriging.

```
Kriging <in_point_features> <z_field> <out_surface_raster>
  <semiVariogram_props> {cell_size} {search_radius}
  {out_variance_prediction_raster}
```

Natural Neighbor: interpolates a surface from points using a natural neighbor technique.

```
NaturalNeighbor <in_point_features> <z_field> <out_raster> {cell_size}
```

Spline: interpolates a surface from points using a minimum curvature spline technique.

```
Spline <in_point_features> <z_field> <out_raster> {cell_size}
  {REGULARIZED | TENSION} {weight} {number_points}
```

Topo to Raster: generates a hydrologically correct raster dataset of elevation.

```
TopoToRaster <feature_layer{Field} {Type};feature_layer{Field} {Type}...>
  <out_surface_raster> {cell_size} {extent} {Margin} {minimum_z_value}
  {maximum_z_value} {ENFORCE | NO_ENFORCE | ENFORCE_WITH_SINK}
  {CONTOUR | SPOT} {maximum_iterations} {roughness_penalty}
  {discrete_error_factor} {vertical_standard_error} {tolerance_1}
  {tolerance_2} {out_stream_features} {out_sink_features}
  {out_diagnostic_file} {out_parameter_file}
```

Topo to Raster by File: generates a hydrologically correct raster dataset of elevation.

```
TopoToRasterByFile <in_parameter_file> <out_surface_raster>
  {out_stream_features} {out_sink_features}
```

Trend: interpolates a surface from points using a trend technique.

```
Trend <in_point_features> <z_field> <out_raster> {cell_size} {order}
  {LINEAR | LOGISTIC}
```

Raster Math Toolset

Contains the tools used to perform mathematics with raster datasets.

Divide: divides the values of two inputs on a cell-by-cell basis.

```
Divide <in_raster_or_constant1> <in_raster_or_constant2> <out_raster>
```

Float: converts each cell value of a raster dataset into a floating-point value.

```
Float <in_raster_or_constant> <out_raster>
```

Int: converts each cell value of a raster dataset into an integer value through truncation.

```
Int <in_raster_or_constant> <out_raster>
```

Minus: subtracts the values of the second input from the values of the first input.

```
Minus <in_raster_or_constant1> <in_raster_or_constant2> <out_raster>
```

Plus: adds the values of two raster datasets on a cell-by-cell basis.

```
Plus <in_raster_or_constant1> <in_raster_or_constant2> <out_raster>
```

Times: multiplies the values in raster datasets on a cell-by-cell basis.

```
Times <in_raster_or_constant1> <in_raster_or_constant2> <out_raster>
```

Raster Reclass Toolset

Contains tools to alter the classes within a raster dataset.

Lookup: creates a new raster dataset by looking up values found in another field in the table of the input raster dataset.

```
Lookup <in_raster> <lookup_field> <out_raster>
```

Reclass by ASCII File: reclassifies (or changes) the values of the input cells of a raster dataset by using an ASCII remap file.

```
ReclassByASCIIFile <in_raster> <in_remap_file> <out_raster> {DATA | NODATA}
```

Reclass by Table: reclassifies (or changes) the values of the input cells of a raster dataset by using a remap table.

```
ReclassByTable <in_raster> <in_remap_table> <from_value_field>
  <to_value_field> <output_value_field> <out_raster> {DATA | NODATA}
```

Reclassify: reclassifies (or changes) the value in a raster dataset.

```
Reclassify <in_raster> <reclass_field> <remap> <out_raster> {DATA | NODATA}
```

Slice: slices a range of values of the input cells by zones of equal area or equal interval.

```
Slice <in_raster> <out_raster> <number_zones> {EQUAL_INTERVAL | EQUAL_AREA}
  {base_output_zone}
```

Raster Surface Toolset

Contains tools to analyze the surface of a raster dataset.

Aspect: identifies the direction of the maximum rate of change in z-value from each cell.

```
Aspect <in_raster> <out_raster>
```

Contour: creates contours or isolines from a raster dataset surface.

```
Contour <in_raster> <out_polyline_features> <contour_interval> {base_contour}
  {z_factor}
```

Contour List: creates contours or isolines from a list of contour values.

```
ContourList <in_raster> <out_polyline_features>
  <contour_values;contour_values...>
```

Curvature: calculates the curvature of a surface at each cell center.

```
Curvature <in_raster> <out_curvature_raster> {z_factor}
  {out_profile_curve_raster} {out_plan_curve_raster}
```

Cut/Fill: calculates cut and fill areas.

```
CutFill <in_before_surface> <in_after_surface> <out_raster> {z_factor}
```

Hillshade: creates a shaded relief raster dataset by considering the illumination angle and shadows.

```
HillShade <in_raster> <out_raster> {azimuth} {altitude}
  {NO_SHADOWS | SHADOWS} {z_factor}
```

Observer Points: identifies exactly which observer points are visible from each surface location within the raster dataset.

```
ObserverPoints <in_raster> <in_observer_point_features> <out_raster>
  {z_factor} {FLAT_EARTH | CURVED_EARTH} {refractivity_coefficient}
```

Slope: identifies the rate of maximum change in z-value from each cell.

```
Slope <in_raster> <out_raster> {DEGREE | PERCENT_RISE} {z_factor}
```

Viewshed: determines the raster dataset surface locations visible to a set of observer features.

```
Viewshed <in_raster> <in_observer_features> <out_raster> {z_factor} {FLAT_EARTH
  | CURVED_EARTH} {refractivity_coefficient}
```

TIN Creation Toolset

Contains tools to create and edit TINs.

Create TIN: creates an empty TIN with an extent based on an input geodataset or coordinates.

```
CreateTin <out_tin> <spatial_reference>
```

Edit TIN: edits the TIN geoprocessing function.

```
EditTin <in_tin> <in_feature_class {Height_Field} {Tag_Field} {SF_Type}
  {Use_Z};in_feature_class {Height_Field} {Tag_Field} {SF_Type} {Use_Z}...>
  {out_tin}
```

TIN Surface Toolset

Contains tools to analyze the surface of a TIN dataset.

Interpolate Polygon to Multipatch: converts a polygon feature class to a multipatch feature class whose heights are based on a surface.

```
InterpolatePolyToPatch <in_tin> <in_feature_class> <out_feature_class>
  {max_strip_size} {z_factor}
```

TIN Aspect: calculates aspect for the surface using each triangle in a TIN.

```
TinAspect <in_tin> <out_feature_class> {class_breaks_table} {aspect_field}
```

TIN Contour: derives contour lines for a surface from a TIN.

```
TinContour <in_tin> <out_feature_class> <interval> {base_contour}
  {contour_field} {contour_field_precision} {index_interval}
  {index_interval_field} {z_factor}
```

TIN Slope: calculates the slope of the surface as the maximum rate of change in elevation across each triangle.

```
TinSlope <in_tin> <out_feature_class> {PERCENT | DEGREE} {class_breaks_table}
  {slope_field} {z_factor}
```

Index

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

Z