

Altair® Monarch® v2023.2

COMMAND LINE USER GUIDE

TABLE OF CONTENTS

[1] Overview	1
[2] Classic Command Line Parameters	2
Input Options.....	2
Input Database Options	2
Input Report Options	3
View Options.....	3
Export Options	3
Command Line Examples	4
[2] Launching Quick Help	6
[3] Using Classic Command Lines to Export Data	7
[4] Data Prep Studio Command Line Interface	8
Input Options.....	8
Input Path Options	13
Export Options	15

[1] OVERVIEW

Monarch Complete supports a number of command line parameters that you can use to open files when you start a Monarch session or automate an entire session and produce an export file.

The syntax for a simple Monarch command line is:

```
Monarch.exe /rpt:[Report File] /mod:[Model File] /exp:[Export File] /[View]
```

To open password-protected PDF files, the following syntax is used:

```
Monarch.exe /rpt:[Report File@password] /mod:[Model File] /exp:[Export File]
/[View]
```

NOTE

The executable name is not the same as the interactive Monarch application. It is named Monarch.exe to help with compatibility of existing scripts for users upgrading from previous versions. The command line module does not have a user interface, to increase performance.

For example, to open a report and Model and then export from Table View:

```
Monarch.exe /rpt:"C:\Users\Public\Documents\Altair
Monarch\Reports\classic.prn" /mod:"C:\Users\Public\Documents\Altair
Monarch\Models\lesson9.dmod" /exp:"C:\Users\Public\Documents\Altair
Monarch\Export\classic.xls" /T
```

To open multiple reports, set a runtime field called "Test" to "True," and export from Summary View:

```
Monarch.exe /rpt:"C:\Users\Public\Documents\Altair
Monarch\Reports\classic.prn" /rpt:"C:\Users\Public\Documents\Altair
Monarch\Reports\classfeb.prn" /mod:"C:\Users\Public\Documents\Altair
Monarch\Models\runtime.dmod" /set:"Test"="True"
/exp:"C:\Users\Public\Documents\Altair Monarch\Export\classic.xls" /S
```

[2] CLASSIC COMMAND LINE PARAMETERS

The tables below present a list and descriptions of the Monarch Classic command line parameters.

INPUT OPTIONS

PARAMETER	DESCRIPTION
<code>/prj:[Project File]</code>	Specifies the Project file to use in the Monarch session. When the /prj and /rpt switches are used in tandem, the reports specified in the command line override any report indicated in the Project, i.e., if the Project already contains a report list, it is discarded.
<code>/mod:[Model File]</code>	Specifies the Model file to use in the Monarch session. Note: Omit this parameter if you specified a Project file as the first parameter.
<code>/set:[Field Name]=[Field Value]</code>	Sets a runtime parameter named [Field Name] to the value [Field Value] from the command line, instead of typing the values into a dialog box. If all runtime parameters are fulfilled from the command line, the corresponding dialog is suppressed upon program startup.

INPUT DATABASE OPTIONS

PARAMETER	DESCRIPTION
<code>/datasource:[name]=connection_string</code>	Specifies the connection string for a named data source, where name is either maindb (for the main database) or the name of an external lookup. For example, <code>/datasource:Lookup2="c:\test\MyData.mdb"</code> sets "c:\test\MyData.mdb" as the source for the external lookup named "Lookup2" and overrides whatever source was specified in the model. If the maindb is to be opened, the syntax <code>"/datasource:maindb=[name]=connection_string"</code> must be explicitly used.
<code>/table:[name=]tablename</code>	Specifies the table to open for the named data source, where name is either maindb (for the main database) or the name of an external lookup. If "name=" part is omitted, maindb is implied. For example, <code>/table:Lookup2=Employees</code> sets "Employees" as the table for the external lookup named "Lookup2" and overrides whatever table was specified in the model.
<code>/pwd:[name=]password</code>	Specifies the password to use for the named data source, where name is either maindb (for the main database) or the name of an external lookup. If "name=" part is omitted, maindb is implied. For example, <code>/pwd:fred</code> sets "fred" as the password to be used for the main import table.

INPUT REPORT OPTIONS

PARAMETER	DESCRIPTION
<code>/rpt:[Report File]</code>	<p>Specifies the report file to use in the Monarch session.</p> <p>You can specify /rpt: more than once to open multiple reports.</p> <p>If you specified a Project file and that Project has existing reports, the reports specified in the command line override any report indicated in the Project, i.e., if the Project already contains a report list, it is discarded.</p>

VIEW OPTIONS

PARAMETER	DESCRIPTION
<code>/R</code>	Sets the current view to Report View
<code>/T</code>	Sets the current view to Table View
<code>/S</code>	Sets the current view to Summary View

EXPORT OPTIONS

PARAMETER	DESCRIPTION
<code>/exp:[Export Path and File]</code>	<p>Specifies the name of an export file to create. Include this parameter if you want to automate a Monarch session.</p> <p>The export file extension determines the file type produced. Valid extensions are: DBF, MDB, ACCDB, TXT, PDF, XLS, XLSX, HTM, HTML, XML or any other extension to create a delimited text file.</p>
<code>/expfileopt:[option]</code>	<p>Sets the file option for the command line export. Possible values are:</p> <ul style="list-style-type: none"><input type="checkbox"/> Overwrite – create table, overwrite if the table already exists<input type="checkbox"/> Add – add (or append) data to an existing table. Create the table first if it does not exist.<input type="checkbox"/> Skip – create the table only if it does not exist.
<code>/exptable:[Tablename]</code>	Sets the name of the table to be used for the command line export
<code>/exptableopt:[option]</code>	Sets the table option for the command line export. Possible values are overwrite, append, and skip.
<code>/px:[jobname]</code>	Runs a Project export jobname if it exists. Multiple jobs may be run using multiple /px switches
<code>/pxall</code>	Specifies that all exports defined within the Project should be executed.

These switches may appear in any order and may appear multiple times (with different name portions).

NOTE

The executable name is not the same as the interactive Monarch application. It is named **Monarch.exe** to help with compatibility of existing scripts for users upgrading from previous versions. Also, starting from Monarch v12, the command line module does not have a user interface, to increase performance.

While these switches are primarily used to override values from a Project or Model file, they can be used to specify a main import table even in the absence of an associated database Project.

For example:

```
Monarch.exe /datasource:"c:\Data\June.mdb" /table:Sales /pwd:xyz  
/mod:"c:\Models\Sales.dmod" /exp:"c:\Exports\JuneSales.xls" /S
```

This command launches Monarch using data from table "Sales" in the Access database "c:\Data\June.mdb" (which requires password "xyz").

The data is loaded and interpreted via Model "c:\Models\Sales.dmod.". The current summary (notice the "/S" switch) is then exported to "c:\Exports\JuneSales.xls."

COMMAND LINE EXAMPLES

The following commands assume that you installed Monarch to the following folder:

```
c:\Program Files\Altair Monarch 2022\Monarch\
```

The following command launches Monarch and appends data to an existing file named classic.txt from Summary View:

```
"c:\Program Files\Altair Monarch 2022\Monarch.exe"  
/rpt:"C:\Users\Public\Documents\Altair Monarch\Reports\classic.prn"  
/mod:"C:\Users\Public\Documents\Altair Monarch\Models\lesson11.dmod"  
/expfileopt:add /exp:"C:\Users\Public\Documents\Altair  
Monarch\Export\classic.txt" /S
```

The following command launches Monarch and exports classic.xls from Table View, overwriting an existing file:

```
"c:\Program Files\Altair Monarch 2022\Monarch.exe"  
/rpt:"C:\Users\Public\Documents\Altair Monarch\Reports\classic.prn"  
/mod:"C:\Users\Public\Documents\Altair Monarch\Models\lesson11.dmod"  
/expfileopt:overwrite /exp:"C:\Users\Public\Documents\Altair  
Monarch\Export\classic.xls" /T
```

The following command launches Monarch and appends data from Table View to an existing named range or sheet called "Test" within an existing file called classic.xls.

```
"c:\Program Files\Altair Monarch 2022\Monarch.exe"  
/rpt:"C:\Users\Public\Documents\Altair Monarch\Reports\classic.prn"  
/mod:"C:\Users\Public\Documents\Altair Monarch\Models\lesson11.dmod"  
/expfileopt:add /exptableopt:append /exptable: "Test"  
/exp:"C:\Users\Public\Documents\Altair Monarch\Export\classic.xls" /T
```

The following command launches Monarch and export data from Table View, overwriting an existing named range or sheet called "Test" within an existing file called classic.xls.

```
"c:\Program Files\Altair Monarch 2022\Monarch.exe"  
/rpt:"C:\Users\Public\Documents\Altair Monarch\Reports\classic.prn"  
/mod:"C:\Users\Public\Documents\Altair Monarch\Models\lesson11.dmod"  
/expfileopt:add /exptableopt:overwrite /exptable:"Test"  
/exp:"C:\Users\Public\Documents\Altair Monarch\Export\classic.xls" /T
```

The following command launches Monarch, opens the password-protected file data.pdf and exports data.xls from Table View.

```
"c:\Program Files\Altair Monarch 2022\Monarch.exe"  
/rpt:"C:\Users\Public\Documents\Altair Monarch\Reports\data.pdf@password"  
/mod:"C:\Users\Public\Documents\Altair Monarch\Models\modelFile.dmod"  
/exp:"C:\Users\Public\Documents\Altair Monarch\Export\data.xls" /T
```

You may use any of the following methods to pass a command line to Monarch:

- Select **Start > Run**, type the command line in the *Open* box, and then choose **OK**. Use this method if you will need to use the command line only once.
- Create an icon that includes the command line. Use this method if you want to run the same Monarch session in the future, such as when you receive a new instance of a report file.
- Enter one or more command lines into a batch file (.bat or .cmd) and execute the batch file.

[2] LAUNCHING QUICK HELP

You can run quick help from the command line to view the available arguments.

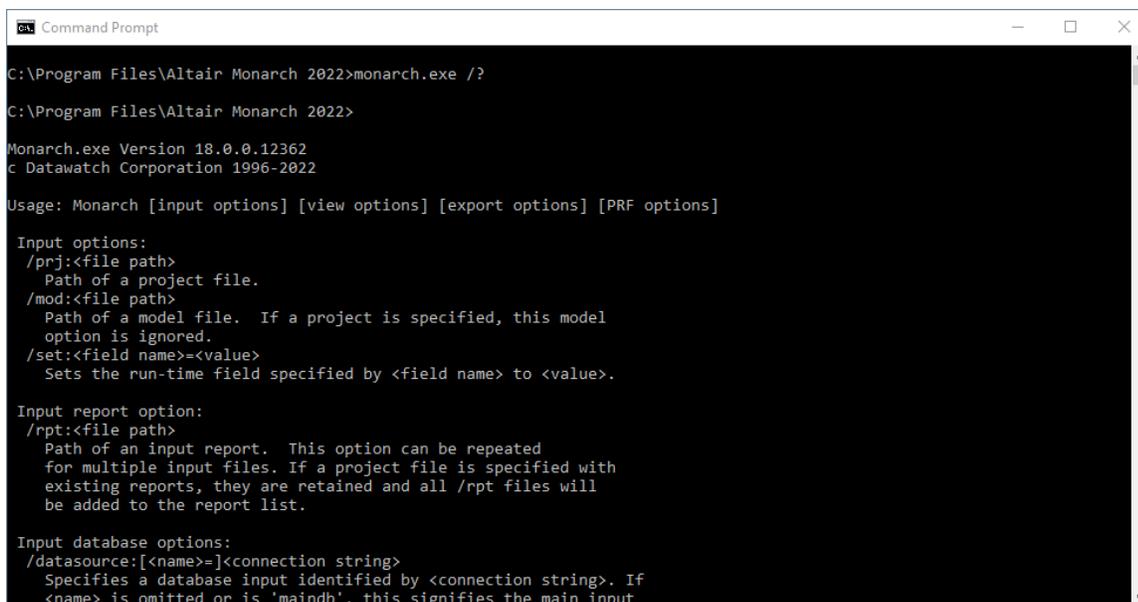
To view quick help, run the following from the command line: `monarch.exe /?`

NOTE

The command above assumes that Monarch is installed in:

`c:\Program Files\Altair Monarch 2023\`

Invoking quick help displays several usage options on screen:



```
Command Prompt
C:\Program Files\Altair Monarch 2022>monarch.exe /?
C:\Program Files\Altair Monarch 2022>
Monarch.exe Version 18.0.0.12362
c Datawatch Corporation 1996-2022

Usage: Monarch [input options] [view options] [export options] [PRF options]

Input options:
/prj:<file path>
  Path of a project file.
/mod:<file path>
  Path of a model file. If a project is specified, this model
  option is ignored.
/set:<field name>=<value>
  Sets the run-time field specified by <field name> to <value>.

Input report option:
/rpt:<file path>
  Path of an input report. This option can be repeated
  for multiple input files. If a project file is specified with
  existing reports, they are retained and all /rpt files will
  be added to the report list.

Input database options:
/datasource:[<name>=]<connection string>
  Specifies a database input identified by <connection string>. If
  <name> is omitted or is 'maindb', this signifies the main input
```

[3] USING CLASSIC COMMAND LINES TO EXPORT DATA

Once you have created a Model file for a Monarch session, you can use a Monarch command line to by-pass the Monarch menus and automatically launch a Monarch session and export data to any supported file format. By creating an icon that includes the command line, you can easily run the same Monarch session in the future. The command line syntax for exporting Monarch data is:

```
Monarch.exe /rpt:[Report File] /mod:[Model File] /exp:[Export File]
```

where [Export File] is the path of the export file you want to create.

For password-protected PDF files, the following syntax is used:

```
Monarch.exe /rpt:[Report File@Password] /mod:[Model File] /exp:[Export File]
```

NOTE

You may replace the /rpt and /mod parameters with a /prj parameter for a Project file that you want to use as input in the session. If the view parameter is omitted, Monarch will export data from Table View.

Example:

The following command line loads the Classic.prn report and the Lesson6.dmod Model file and then creates a text file (Classic.txt) from the data in Table View:

```
Monarch.exe /rpt:"C:\Users\Public\Documents\Altair  
Monarch\Reports\classic.prn" /mod:"C:\Users\Public\Documents\Altair  
Monarch\Models\lesson6.dmod" /exp:"classic.txt" /T
```

The command above assumes, for simplicity, that your report and Model folders are located in C:\Users\Public\Documents\Altair Monarch\Reports\ and C:\Users\Public\Documents\Altair Monarch\Models\, respectively.

In most practical cases, your report and Model files will likely be located elsewhere.

[4] DATA PREP STUDIO COMMAND LINE INTERFACE

The Command Line Interface in Data Prep Studio is of two types:

□ **Interactive**

- Can execute via the main executable file
- Launches an instance of the application
- No additional license required

□ **Non-interactive**

- Requires a separate executable file to run
- Does not launch the application; runs silently from the command line
- Additional license required

The tables below describe the command line parameters available in Data Prep Studio.

INPUT OPTIONS

PARAMETER	DESCRIPTION	TYPE
<code>/rpt:<File Path[:p=password]></code>	Brings up an interactive session in Report Design with the list of report files (text or PDF) being modeled together. Can be set multiple times. Optional parameter: :p	Interactive only
<code>/delim:<File Path></code>	Opens a new delimited text load plan using the file path provided. Can be set multiple times.	Both
<code>/excel:<File Path[:t=table][:p=password]></code>	Opens a new Excel load plan using the file path provided. If the table is not specified and only one table is available, this table will be loaded. If the table is not specified and multiple tables are available, the table explorer will be launched. The table and password can be specified in any order. Can be set multiple times.	Both*

PARAMETER	DESCRIPTION	TYPE
	<p>Optional parameters: :t, :p :t is required in the non-interactive type.</p>	
/access:<File Path[:t=table] [:p=password]>	<p>Opens a new Access load plan using the file path provided.</p> <p>If the table is not specified and only one table is available, this table will be loaded.</p> <p>If the table is not specified and multiple tables are available, the table explorer will be launched.</p> <p>The table and password can be specified in any order.</p> <p>Can be set multiple times.</p> <p>Optional parameters: :t, :p :t is required in the non-interactive type.</p>	Both*
/html<File Path[:t=table]>	<p>Opens a new html file load plan using the file path provided.</p> <p>If the table is not specified and multiple tables are available, the table explorer will be launched.</p> <p>Can be set multiple times.</p> <p>Optional parameter: :t is required in the non-interactive type.</p>	Both
/ftp:<URL> [:u=username] [:p=password]	<p>Opens a table from an FTP source using the file path provided.</p>	
/http:<URL> [:u=username] [:p=password]	<p>Opens a table from an HTTP source using the file path provided.</p>	
/sastransportv5:<File Path>	<p>Opens a new SAS Transport V5 file load plan using the file path provided.</p> <p>Can be set multiple times.</p>	Both
/awss3:<URL [:t=table] [:p=password] [:akid=accessKeyId] [:skid=secretKey]>	<p>Opens a new file-based load plan using the URL to file on an AWS S3 instance. This URL must begin with "awss3://" in order to be recognized as a valid S3 input.</p> <p>In non-interactive mode, only delimited text, Excel, Access, HTML, and SAS Transport V5 files are currently supported through this method. The import method is defined by the file extension in the URL. These sources follow identical optional parameters to their corresponding input commands (/delim, /excel, /access, /sastransportv5).</p>	Both*

PARAMETER	DESCRIPTION	TYPE
	<p>:akid – the Access Key ID for the Amazon S3 account/instance being exported to.</p> <p>:skid – the account-specific Amazon S3 secret key.</p> <p>If :akid and :skid are not provided, registry-stored credential use will be attempted.</p> <p>Similar parameter requirements to source file type import function.</p> <p>Can be set multiple times.</p>	
<pre>/adl:<URL [:t=table][:p=password]> [:an=accountName] [:ak=accountKey]</pre>	<p>Opens a new file-based load plan using the URL to file on an Azure Data Lake Gen2 instance. This URL must begin with "adl://" in order to be recognized as a valid Azure Data Lake Gen2 input.</p> <p>In non-interactive mode, only delimited text, Excel, Access, HTML, and SAS Transport V5 files are currently supported through this method. The import method is defined by the file extension in the URL. These sources follow identical optional parameters to their corresponding input commands (/delim, /excel, /access, /sastransportv5).</p> <p>:an – the account name for the Azure Data Lake Gen2 instance being exported to.</p> <p>:ak – the (secret) account key for the corresponding account provided by ":an".</p> <p>If :an and :ak are not provided, registry-stored credential use will be attempted.</p> <p>Similar parameter requirements to source file type import function.</p> <p>Can be set multiple times.</p>	Both*
<pre>/abs:<URL [:t=table][:p=password]> [:an=accountName] [:ak=accountKey]</pre>	<p>Opens a new file-based load plan using the URL to file on an Azure Blob Storage instance. This URL must begin with "abs://" in order to be recognized as a valid Azure Blob Storage input.</p> <p>In non-interactive mode, only delimited text, Excel, Access, HTML, and SAS Transport V5 files are currently supported through this method. The import method is defined by the file extension in the URL. These sources follow identical optional parameters to their corresponding input commands (/delim, /excel, /access, /sastransportv5).</p> <p>:an – the account name for the Azure Blob Storage instance being exported to.</p>	Both*

PARAMETER	DESCRIPTION	TYPE
	<p>:ak – the (secret) account key for the corresponding account provided by :an. If :an and :ak are not provided, registry-stored credential use will be attempted.</p> <p>Similar parameter requirements to source file type import function. Can be set multiple times.</p>	
<pre>/oci:<URL [:t=table] [:p=password]> [:akid=accessKeyId] [:skid=secretKey]</pre>	<p>Opens a new file-based load plan using the URL to file on an Oracle Cloud Infrastructure (OCI) instance. This URL must begin with "oci://" in order to be recognized as a valid OCI input.</p> <p>In non-interactive mode, only delimited text, Excel, Access, HTML, and SAS Transport V5 files are currently supported through this method. The import method is defined by the file extension in the URL. These sources follow identical optional parameters to their corresponding input commands (/delim, /excel, /access, /sastransportv5).</p> <p>:akid – the Access Key ID for the OCI account/instance being exported to. :skid – the account-specific OCI secret key. If :akid and :skid are not provided, registry-stored credential use will be attempted.</p> <p>Similar parameter requirements to source file type import function. Can be set multiple times.</p>	Both*
<pre>/gcs:<URL [:t=table] [:p=password]> [:akid=accessKeyId] [:skid=secretKey]</pre>	<p>Opens a new file-based load plan using the URL to file on a Google Cloud Storage (GCS) instance. This URL must begin with "gcs://" in order to be recognized as a valid GCS input.</p> <p>In non-interactive mode, only delimited text, Excel, Access, HTML, and SAS Transport V5 files are currently supported through this method. The import method is defined by the file extension in the URL. These sources follow identical optional parameters to their corresponding input commands (/delim, /excel, /access, /sastransportv5).</p> <p>:akid – the Access Key ID for the GCS account/instance being exported to. :skid – the individual account or domain-specific GCS secret key. If :akid and :skid are not provided, registry-stored credential use will be attempted.</p>	Both*

PARAMETER	DESCRIPTION	TYPE
	<p>Similar parameter requirements to source file type import function.</p> <p>Can be set multiple times.</p>	
<pre>/append[:type=Type] [:t=tableName]</pre>	<p>Appends the list of input tables.</p> <p>Type can be: {name, order}.</p> <p>Only one append is allowed as part of the command string; any that are seen after the first will be discarded.</p> <p>This flag will append any eligible tables that were part of the command: delimited tables and Excel/Access with a single table or table specified.</p> <p>The desired name for the append table produced can be supplied with the “:t=” option AFTER a type is applied, if any.</p> <p>If no table name is specified the default name 'Append' is used and disambiguated accordingly.</p> <p>Can be set only once.</p> <p>Optional parameters: :type, :t</p>	Both
<pre>/dpwx:<File Path[:p=password]></pre>	<p>Opens the specified Workspace file.</p> <p>If this flag is used, any files that are opened with the other flags will be opened in the context of this Workspace.</p> <p>A password can be entered in the case of encrypted Workspaces.</p> <p>Can be set only once.</p> <p>Optional parameter: :p</p>	Both
<pre>/setrtip:<Field Name>=<Field Value>[:scope=Scope] [:type=Type]</pre>	<p>Sets a runtime parameter named [Field Name] to the value [Field Value] from the command line, instead of typing the values into a dialog box.</p> <p>If all runtime parameters are fulfilled from the command line, the corresponding dialog is suppressed upon program startup.</p> <p>[:Scope] {a, application, w, workspace} for application and workspace settings.</p> <p>Scope is optional; if it is not specified, then duplicates will be resolved in the usage of the application version of the parameter.</p> <p>Type {t, text, n, numeric, d, date} is optional and only necessary in the presence of RTP values of the same scope with the same name and different types (DPS allows this scenario, unlike Classic).</p>	Both

PARAMETER	DESCRIPTION	TYPE
	<p>If this type of duplication is present and Type is not specified, the first type found is set.</p> <p>Can be set multiple times.</p> <p>Optional parameters: :s, :type</p>	

Both* – In non-interactive mode, the **/excel** and **/access** flags are not valid unless a **:t=** modifier is present.

INPUT PATH OPTIONS

PARAMETER	DESCRIPTION	TYPE
<pre>/src:<File Path or URL[:p=password][:cloud=cloud service][:an=azure account name][:ak=azure account key][:akid=S3/S3 compatible access key id][:skid=S3/S3 compatible secret key id]<:t=table>></pre>	<p>Sets the path for the indicated file-based load plan.</p> <p>Setting this property will overwrite the path previously specified in the Workspace.</p> <p>A password can be set if required, but can be omitted if not.</p> <p>:t – the table to be re-pathed.</p> <p>To access child tables, delimit with the @ character. If a table name is using the @ character, the @ in the command can be “escaped” by adding an additional @. If two sub-tables have the same name in the path, an error is returned.</p> <p>Can be set multiple times for a multi-file load plan.</p> <p>May accept URLs to cloud sources as well as https, http, and ftp URLs.</p> <p>Optional parameters: :p, :cloud</p> <p>Cloud service values are: gcs, oci, awss3, abs, and adi, corresponding to Google Cloud Storage, Oracle Cloud Infrastructure, Amazon S3, Azure Blob Storage, and Azure Data Lake Gen 2, respectively.</p> <p>Account Key and Account Name can ONLY be validly used for Azure sources.</p>	Non-interactive only

PARAMETER	DESCRIPTION	TYPE
	Access Key ID and Secret Key ID can ONLY be validly used for S3, OCI, and GCS.	

EXPORT OPTIONS

PARAMETER	DESCRIPTION	TYPE
<pre>/exp:<File Path<:t=table> <:fo=fileOption> [:to=tableOption] [:tn=exportTableName]></pre>	<p>Executes an export of the specified target table to the specified path.</p> <p>:t – the table to be exported.</p> <p>To access child tables, delimit with the @ character. If a table name uses the @ character, the @ in the command can be “escaped” by adding an additional @. If two sub-tables have the same name in the path, an error is returned.</p> <p>The export file extension determines the file type produced. Valid extensions are: XLS, XLSX, ACCDB, MDB, CSV, TSV, TAB, JSON, XML, XPT, KDD, TDE, and QVX. Any unrecognized extension will result in a delimited text export.</p> <p>Can be set multiple times for a multi-file load plan.</p> <p>Optional parameters: :to (required for Excel and Access), :tn</p>	Non-interactive only
<pre>/expawss3:<URL <:t=table><:fo=fileOption> [:to=tableOption] [:tn=exportTableName]> [:akid=accessKeyId] [:sk=secretKey]</pre>	<p>Executes an export of the specified table (:t) to the provided S3 url. This url must begin with "awss3://" in order to be recognized as a valid S3 export.</p> <p>:t – the table to be exported. :akid – the Access Key ID for the Amazon S3 account/instance being exported to. :sk – the account-specific Amazon S3 secret key.</p> <p>If :akid and :sk are not provided, registry-stored credential use will be attempted.</p> <p>The export file extension determines the file type produced. Valid extensions are: XLS, XLSX, ACCDB, MDB, CSV, TSV, TAB, JSON, XML, XPT, KDD, TDE, and QVX. Any unrecognized extensions will result in a delimited text export.</p> <p>Can be set multiple times for a multi-file load plan.</p> <p>Optional parameters: :to (Required for Access and Excel), :tn</p>	Non-interactive only
<pre>/expadl:<URL <:t=table> <:fo=fileOption></pre>	<p>Executes an export of the specified table (:t) to the provided Azure Data Lake Gen2</p>	Non-interactive only

PARAMETER	DESCRIPTION	TYPE
<pre>[:to=tableOption] [:tn=exportTableName]> [:an=accountName] [:ak=accountKey]</pre>	<p>URL. This URL must begin with "https://" in order to be recognized as a valid Azure Data Lake Gen2 export.</p> <p>:t – the table to be exported. :an – the account name for the Azure Data Lake Gen2 instance being exported to. :ak – the (secret) account key for the corresponding account provided by ":an". If :an and :ak are not provided, registry-stored credential use will be attempted.</p> <p>The export file extension determines the file type produced. Valid extensions are: XLS, XLSX, ACCDB, MDB, CSV, TSV, TAB, JSON, XML, XPT, KDD, TDE, and QVX. Any unrecognized extension will result in a delimited text export.</p> <p>Can be set multiple times for a multi-file load plan.</p> <p>Optional parameters: :to (Required for Access and Excel), :tn</p>	
<pre>/expabs:<URL <:t=table><:fo=fileOption> [:to=tableOption] [:tn=exportTableName]> [:an=accountName] [:ak=accountKey]</pre>	<p>Executes an export of the specified table (:t) to the provided Azure Blob Storage URL. This URL must begin with "https://" in order to be recognized as a valid Azure Blob Storage export.</p> <p>:t – the table to be exported. :an – the account name for the Azure Blog Storage instance being exported to. :ak – the (secret) account key for the corresponding account provided by ":an". If :an and :ak are not provided, registry-stored credential use will be attempted.</p> <p>The export file extension determines the file type produced. Valid extensions are: XLS, XLSX, ACCDB, MDB, CSV, TSV, TAB, JSON, XML, XPT, KDD, TDE, and QVX. Any unrecognized extension will result in a delimited text export.</p> <p>Can be set multiple times for a multi-file load plan.</p> <p>Optional parameters: :to (Required for Access and Excel), :tn</p>	Non-interactive only
<pre>/expoci:<URL <:t=table> <:fo=fileOption> [:to=tableOption] [:tn=exportTableName]></pre>	<p>Executes an export of the specified table (:t) to the provided OCI URL. This URL must begin with "https://" in order to be recognized as a valid OCI export.</p>	Non-interactive only

PARAMETER	DESCRIPTION	TYPE
<pre>[:akid=accessKeyId] [:skid=secretKey]</pre>	<p>:t – the table to be exported.</p> <p>:akid – the Access Key ID for the OCI account/instance being exported to.</p> <p>:skid – the account-specific OCI secret key.</p> <p>If :akid and :skid are not provided, registry-stored credential use will be attempted.</p> <p>The export file extension determines the file type produced. Valid extensions are: XLS, XLSX, ACCDB, MDB, CSV, TSV, TAB, JSON, XML, XPT, KDD, TDE, and QVX. Any unrecognized extension will result in a delimited text export.</p> <p>Can be set multiple times for a multi-file load plan.</p> <p>Optional parameters: :to (Required for Access and Excel), :tn</p>	
<pre>/expgcs:<URL <:t=table> <:fo=fileOption> [:to=tableOption] [:tn=exportTableName]> [:akid=accessKeyId] [:skid=secretKey]</pre>	<p>Executes an export of the specified table (:t) to the provided GCS URL. This URL must begin with "https://" in order to be recognized as a valid GCS export.</p> <p>:t – the table to be exported.</p> <p>:akid – the Access Key ID for the GCS account/instance being exported to.</p> <p>:skid – the account-specific GCS secret key.</p> <p>If :akid and :skid are not provided, registry-stored credential use will be attempted.</p> <p>The export file extension determines the file type produced. Valid extensions are: XLS, XLSX, ACCDB, MDB, CSV, TSV, TAB, JSON, XML, XPT, KDD, TDE, and QVX. Any unrecognized extension will result in a delimited text export.</p> <p>Can be set multiple times for a multi-file load plan.</p> <p>Optional parameters: :to (Required for Access and Excel), :tn</p>	Non-interactive only
<pre>:fo=fileOption</pre>	<p>Sets the file option for the command line export. Possible values are:</p> <ul style="list-style-type: none"> <input type="checkbox"/> Overwrite – create table, overwrite if the table already exists 	Non-interactive only

PARAMETER	DESCRIPTION	TYPE
	<ul style="list-style-type: none"> <input type="checkbox"/> Add – add (or append) data to an existing table. Create the table first if it does not exist. <input type="checkbox"/> Skip – create the table only if it does not exist. <p>Note: Parameter for the /exp flag family</p>	
<code>:to=tableOption</code>	<p>Sets the table option for the command line export. Possible values:</p> <ul style="list-style-type: none"> <input type="checkbox"/> Overwrite – create table, overwrite if the table already exists <input type="checkbox"/> Add – add (or append) data to an existing table. Create the table first if it does not exist. <input type="checkbox"/> Skip – create the table only if it does not exist. <p>:to is required for Access and Excel. The export will fail if not specified</p> <p>Note: Parameter for the /exp flag family</p>	Non-interactive only
<code>:tn=exportTableName</code>	<p>Sets the name of the table to be used for the command line export.</p> <p>If :tn is not set, "Untitled" will be used.</p> <p>Note: Parameter for the /exp flag family</p>	Non-interactive only
<code>/expnamed:[export name] or /px:[jobname]</code>	<p>Runs a named export if it exists.</p> <p>Multiple exports may be run using multiple /expnamed switches.</p> <p>Can be set multiple times.</p>	Non-interactive only
<code>/expall or /pxall</code>	<p>Runs all exports defined within the workspace</p> <p>Can be set only once (subsequent flags seen will be discarded after the first flag is successfully performed without failing)</p>	Non-interactive only

The `/px` export commands are included to aid users of the Classic CLI. These switches may appear in any order and may appear multiple times (with different name portions).

Behavior in the event of failures for non-interactive CLI

If any flag is malformed or fails to execute due to an incorrect path or otherwise, the entire command will immediately fail and output error text to the command window. Note that this includes exports, but if an export fails, previous exports that completed successfully cannot be undone.

[5] INSTALLATION COMMAND LINE PARAMETERS

The tables below present a list and descriptions of command line parameters for Monarch installation.

USAGE

PARAMETER	DESCRIPTION
-quiet	Quiet mode
-passive	Unattended mode (this installation shows only a progress bar)
-n, -productName	Product Name
-uiCulture	License Manager UI Culture
-log	Log file location
-deactivation	Deactivate flag
-installFolder	Install folder location
-sampleFiles	Install folder location for sample files (for quiet mode only)
-noSampleFiles	Do not install samples files (for quiet mode only)
-noAlmUtils	Do not install ALM Utils (for quiet mode only)
-noADE	Do not install Access Database Engine 2010 prerequisite
-sqlLocalDbVersion	Version of SQL Server LocalDB to use for the Windows cache (e.g., "11.0," "13.0;" the default setting is "11.0")
-excelDefaultImport Engine	Default Excel engine to use when creating new projects and models (Modern or Legacy)
-excelConvertMetadata ToModern	Convert existing projects and models to use the Modern Excel engine
-uninstall	Uninstall the application
-repair	Repair the application
-activationType	Activation type (nalpeiron , licenseServer , licenseFile , or altairLicenseServer)
-displayLegacyLicensing	Show legacy licensing methods (Nalpeiron Activation Code, Volume License Key File)
-x86	Force 32-bit installation (may affect MS Access import/export)
-x64	Force 64-bit installation (may affect MS Access import/export)
-help, /?	Display this help screen

NALPEIRON

PARAMETER	DESCRIPTION
-activationCode	Nalpeiron activation code
-proxyServer	Nalpeiron proxy server
-proxyPort	Nalpeiron proxy server port
-proxyUser	Nalpeiron proxy server user
-proxyUserPassword	Nalpeiron proxy server user password
-proxyAutoDetectDisabled	Disable Nalpeiron proxy autodetection
-sync	Synchronize current activation with Nalpeiron server

LICENSE FILE

PARAMETER	DESCRIPTION
-licenseFilePath	License File location

MONARCH LICENSE SERVER

PARAMETER	DESCRIPTION
-licenseServer	Monarch License Server location
-instanceName	Monarch License Server instance name
-userName	Monarch License Server user
-password	Monarch License Server user password
-domain	Monarch License Server domain name
-authentication	Monarch License Server authentication type (basic or Windows)
-isSecuredHttp	Monarch License Server secured http

ALTAIR LICENSE SERVER

PARAMETER	DESCRIPTION
-licenseServer	Altair License Server location
-licenseFilePath	Altair License File location
-activationKey	Altair License activation key
-alasStore	Specifies where on your machine the license file will be stored. The options are user and system . If unspecified, the default setting is user .
-overwrite	Overwrites an existing license with the same activation key in the license store if it already exists.
-licenseDataOutput FilePath	Export current license data to specified file

COMMAND LINE EXAMPLES

The following commands assume that the installer is located in:

```
c:\Program Files\Altair Monarch 2022\LicenseManager\
```

Nalpeiron Activation

```
C:\Program Files\Altair Monarch
2022\LicenseManager\Datawatch.Licensing.Manager.exe" --productName="Datawatch
Monarch 18" --activationType=nalpeiron --activationCode=111111111111111111 --
proxyAutoDetectDisabled --proxyServer=127.0.0.1 --proxyPort=8080 --
proxyUser=proxyUser --proxyUserPassword=proxyUserPwd
```

Nalperion Synchronization

```
"C:\Program Files\Altair Monarch
2022\LicenseManager\Datawatch.Licensing.Manager.exe" --productName="Datawatch
Monarch 18" --sync
```

Nalperion Deactivation

```
"C:\Program Files\Altair Monarch
2022\LicenseManager\Datawatch.Licensing.Manager.exe" --productName="Datawatch
Monarch 18" --deactivation
```

License File Activation

```
"C:\Program Files\Altair Monarch
2022\LicenseManager\Datawatch.Licensing.Manager.exe" --productName="Datawatch
Monarch 18" --activationType=licenseFile --
licenseFilePath="C:\Licenses\license.lic"
```

Monarch License Server Activation

```
"C:\Program Files\Altair Monarch
2022\LicenseManager\Datawatch.Licensing.Manager.exe" --productName="Datawatch
```

```
Monarch 18" --activationType=licenseServer --licenseServer=localhost:51111 --instanceName="lsInstance"
```

Altair License Server Activation

```
"C:\Program Files\Altair Monarch  
2022\LicenseManager\Datawatch.Licensing.Manager.exe" --productName="Datawatch  
Monarch 18" --activationType=altairLicenseServer --  
licenseServer=6200@localhost
```

Altair License File Activation

```
"C:\Program Files\Altair Monarch  
2022\LicenseManager\Datawatch.Licensing.Manager.exe" --productName="Datawatch  
Monarch 18" --activationType=altairLicenseServer --  
licenseFilePath="C:\Licenses\alf.dat"
```

Altair License Key activation (ALAS)

```
"C:\Program Files\Altair Monarch  
2022\LicenseManager\Datawatch.Licensing.Manager.exe" --productName="Datawatch  
Monarch 18" --activationType=altairLicenseServer --activationKey="LXXXX-  
XXXXX-XXXXX-XXXXX" --alasStore=[user|system] --overwrite
```

Altair License Key Deactivation (ALAS)

```
"C:\Program Files\Altair Monarch  
2022\LicenseManager\Datawatch.Licensing.Manager.exe" --productName="Datawatch  
Monarch 18" --deactivation --activationKey="LXXXX-XXXXX-XXXXX-XXXXX" --  
alasStore=user
```

Current 'License Data' Export

```
"C:\Program Files\Altair Monarch  
2022\LicenseManager\Datawatch.Licensing.Manager.exe" --licenseDataOutput
```