DAx Command Line Parameters

A number of command line parameters are available for DAx. To use these parameters, go to the **Run** option in the Start Menu, and enter:

DAx32 <command-line-options>

Command line parameters can also be used when DAx is run from a batch file, for instance for the unsupervised analysis of vast numbers of files.

The command line options are defined as follows:

DAx [/a] [/c] [/aa:codeletters] [/al:filename] [/dd:dirname] [/df:filename] [/od:dirname] [/op:operatorname] [/pd:dirname] [/r+/-/1..9] [/wth] [/z:00,11] [/zz:00,11] [/z-] [[filename]]

The **-** (hyphen) can be used instead of the **/** (slash).

/a — performs analysis (constructing baseline, finding peaks) on data that have been read. To set up what kind of analysis will be performed, use the ***/aa*** option. The setup can also be performed inside DAx, using the **Peaks | Config auto-analysis** menu option (when a data file has been loaded).

/c — closes DAx when all files have been read and analysed. This option only makes sense when combined with the ***/a*** option, or when certain non-data files are read. Insert this option at the start of the command line parameters to cause analysed data files to be saved.

The ***/a*** and ***/c*** options should be specified at the start of the command line.

/aa: — define analysis that is to be performed. Settings specified here also define the AutoAnalysis settings for files that are subsequently opened with the File | Open dialog in DAx. See below for available analysis options.

/al:filename	starts using **filename** as an analysis log file. This option can be repeated, defining a different log file name for subsequent data files.
/bs:filename	write the current binning sheet to the specified filename. This option should be included *after* one or more file names are specified, so that the binning sheet has been created from these files. If a file with the specified filename already exists, it is overwritten. If the file name has an extension of .NEX, the file is written in PAUP/NEXUS format. An extension of .PHY causes the file to be written in sequential PHYLIP format.
/bs-	empty the binning sheet.
/dd	makes DAx store analysed data files in the Measurement Data Directory. Original filenames will be used. This option only makes sense used in conjunction with the /a parameter.
/dd:pathname	sets a new Measurement Data Directory. The directory must already exist. Also stores analysed data files in that directory. This option can be repeated. See also the **/od** and **/pd** options.
/dd-	stops storing analysed data files in the Measurement Data Directory. Files will be stored in the original data directory, unless an **/od:** option has been specified.
/df:filename	**Parameter Definition File** name. Refer to the /r command line options and to the next sections.
/fx+	fixates data after they have been analysed. This prevents the data being re-analysed when new analysis parameters are loaded. Saving the data also causes fixation.
/fx-	stops fixating data after they have been analysed.
/gs:filename	write the current global comparison sheet to the specified filename. This option should be included *after* one or more file names are specified, so that the global sheet has been created from these files. If a file with the

	specified filename already exists, it is overwritten.
/gs-	empty the global comparison sheet.
/hide	hides DAx until the command line analyses have been completed.
/n+	creates new filenames for data files and peak list files. Existing files will not be overwritten. An exception occurs when DAx data files are being opened and analysed. When the data are saved, existing files will be overwritten. This is safe to do, because this operation corresponds to a **File | Save** operation (not a **File | Save as** operation).
/n-	Overwrites existing data files and peak files, as needed.
/od:pathname	defines a directory where analysed data will be stored. If the path is specified as a relative path (i.e. it does not have a drive letter specification, and it does not start with a backslash \), the path will be relative to the directory a data file is found in. If needed, the directory will be newly created (only a single directory level can be specified in that case).
/od+	stores analysed data in the same file the data were loaded from. If the data were loaded from a non-DAx file, they are saved in a DAx file.
/od-	stops storing analysed data.
/op:operatorname	logs in as **operatorname**. If this option is not used, the log-in dialog will be displayed. If DAx is to be run as part of a batch, the /op option is therefore required.
/pd:pathname	defines a directory where peak lists will be stored. If the path is specified as a relative path (i.e. it does not have a drive letter specification, and it does not start with a backslash \), the path will be relative to the directory a data file is found in. If needed, the directory will be newly created (only a single directory level can be specified in that case). Specifying the **/pd:** option will cause peak list files to be written (i.e. it implies /aa:d+) If no **/pd:** option is specified, but a **/od:** option

	is specified, peaks will be stored in the path specified in the latter.
/r+, /r-, /r1 .. /r9	recurse subdirectories when opening files. /r+ will recurse a maximum of 10 levels, /r5 will recurse no deeper than 5 levels. /r- stops recursing subdirectories. Filenames may be specified with a wild card, but it is also possible to specify a fully qualified name. DAx32 /r5 test5.dax will open all test5.dax files that lie in the current directory, no deeper than 5 subdirectories deep. Refer to the next section, **Parameter Definition Files**, for information about defining parameters for each subdirectory.
/wtv	tiles data windows vertically at the end of the command line sequence.
/wth	tiles data windows horizontally at the end of the command line sequence.
/z:aa,bb	zooms any data windows that are opened to the specified interval, aa to bb units. It is not possible to specify time coordinates in minutes, they must be specified in seconds. The vertical scale is optimised. This option can be repeated.
/zz:aa,bb	identical to /z:aa,bb; however, the data are analysed first, and then zoomed. This is significant particularly when the analysis performs an axis conversion (e.g. from time to calibrated).
/z:aa,bb,cc,dd	identical to /z:aa,bb, but the vertical scale is set to cc - dd. aa and bb can be omitted (e.g. /z:,,0,1)
/zz:aa,bb,cc,dd	identical to /zz:aa,bb, but the vertical scale is set to cc - dd. aa and bb can be omitted.
/z-	stops zooming in on an interval. Both horizontal and vertical scale of any data windows that are opened are optimised.
[[filename]]	one or more filenames with data that is to be read. All files that can be loaded from the File Open menu option in DAx can be read. This includes parameter and list files. Wild cards may be used to load multiple files

(e.g. DAx32 test*.dax).
Files are recognised by their extension. This means that if a data file has the extension **.lst**, it cannot be read as data - it would be read as a list file.
Two types of files that are not in the File Open dialog box can also be read: **.tap** files are read into *both the ABI and MegaBACE™ trace analysis parameters*. **.tdb** files are read into the MegaBACE™ colour separation matrix.
Files are read in the order they are listed on the command line. This means that the command line

DAx32 /a matrix1.tdb data1.esd matrix2.tdb data2.esd

will read and analyse data1.esd using matrix1, and will analyse data2.esd with matrix2.

The **/aa:**, **/al:**, **/dd:**, **/fx**, **/n**, **/od**, **/pd**, **/z:** and **/zz:** options can be repeated. In this case, any [[filename]] that are specified are analysed with the parameters that precede it. For instance,

```
DAx32 /a /op:pp  /al:c:\dax\log1.log  c:\dax\test1.dax
              /al:c:\dax\log2.log  c:\dax\test2.dax
```

will store analysis log values for test1.dax in log1.log, and results for test2.dax in log2.log.

Analysis Settings (specified with the /aa: option)

The **/aa:** option takes the form /aa:[[[[codeletter]]{+/-}]]

For instance /aa:fbp+a-d+n-
turns options f, b, p and d on, turns options a and n off.

Available codeletters are:

F	filter data
H	hide (do not plot) raw data after filtered data have been created. Use only in conjunction with the F option
B	construct baseline
U	subtract baseline from data, then remove baseline

P	find peaks
O	overlay (normalise) data (using the most recently used setting in the Data Overlay dialog box)
R	print a report
A	print data plot (reports are preferred, use the R option)
I	print data info
L	print a peak list
N	only list named peaks
D	store a peak list in a file
C	close data window after analysis
X	close peak list windows after analysis
S	convert horizontal axis to calibrated units after analysis
T	convert horizontal axis to molecular weight after analysis
V	add data to binning sheet when analysis is complete
W	add data to global comparison sheet when analysis is complete
E	exclude data from binning sheet and global comparison sheet if data are a calibration trace
G	derive molecular weight calibration from data
Q	derive calibration from data

Example command line

```
dax32   /a /od:mulout /pd:peakout /fx+ /r1 /op:peter d:\bio7\bio7.tap
        d:\bio7\*.rsd
```

This command line

- performs automatic analysis
- stores analysed data in a subdirectory called mulout
- stores peak lists in a subdirectory called peakout
- fixates data after analysis
- recurses subdirectories one level deep
- logs in as peter (no password is required)

First, the trace analysis parameter file d:\bio7\bio7.tap is loaded. After that, all RSD files in d:\bio7 and its subdirectories are loaded and analysed.

This command line presupposes that automatic analysis has been set up. This could also have been done by including an option such as /aa:bp+ on the command line.

When and Where are Analysed Data Stored?

Whether or not analysed data are stored, and where, is determined by the **/c**, **/dd** and **/od** options.

- if **/c** is specified, data are stored, unless the **/od-** options is specified. Data are normally stored in their own directory. If the **/dd** or **/dd:** option is specified, data are stored in the default data directory. If the **/od:** option is specified, data are stored in the output data directory.
- if **/c** is not specified, data are not stored, unless the **/dd**, **/dd:**, **/od+** or **/od:** option is specified. If the data are analysed but not stored, the user will be asked if the data need to be saved before the data window can be closed.

Parameter Definition Files

When the /r command line parameter is used to open data in subdirectories, a parameter definition file may be used to specify **/aa:**, **/al:**, **/dd**, **/fx**, **/n**, **/od**, **/pd**, **/z:** and **/zz:** options, as well as parameter files, for each subdirectory.

To do this, create a file **dirs.def** in the current directory (an alternate file name may be specified using the **/df:** command line option). The file should contain a number of lines, each consisting of a directory name, followed by a number of parameters.
The parameters can contain **/aa:**, **/al:**, **/dd**, **/fx**, **/n**, **/od**, **/pd**, **/z:** and **/zz:** command line options. You may also specify parameter file names (analysis procedures, GPC / CE / HPLC / Calibration parameter files, trace analysis parameter files, matrix definition files). Data file names will be ignored.
The lines in dirs.def should be in alphabetical order. Whenever DAx is about to enter a subdirectory, it will scan the dirs.def file. The line that specifies a directory name that is equal to or smaller than the directory being entered will be executed.
Lines will only be executed for the first level of subdirectories. If the subdirectory contains another subdirectory, no additional parameters will be retrieved from dirs.def.

Do not assume that a line in the definition file will be executed if

the directory exists. At least one data file must be loaded from the directory for the line to be executed.
Do not assume that subdirectories will be traversed in alphabetical order. This depends on your operating system.

Iterating over Parameter Files

DAx has the ability to do iterative analysis, where first multiple parameter files are each used to perform a temporary analysis of certain data files, after which the parameter file that gave the highest quality analysis is automatically used for the permanent analysis.

NB make sure analysis assays are calculated! After all, they are used to determine which are the best parameters.

Use a command line like this one:
DAx32 /op:pp /a [par1.par par2.par par3.par] file.rsd all*.rsd [] regular.rsd

The files between [and] specify which parameter files to iterate through. In this case, **file.rsd** and all **all*.rsd** will be iterated.
regular.rsd will be loaded normally (using whichever of **par1.par**, **par2.par** and **par3.par** was loaded last). Note the use of an empty collection of parameter files to stop iterative analysis.

When **file.rsd** is analysed, it is first analysed with **par1.par**, **par2.par**, **par3.par**. Each of these analyses gives a quality. NB no output is generated at this stage (no printing, no peak files, no file saving).
When the three analyses are done, the file that gave the best quality is used again, this time "for real" (with printing, peak file, file saving).

All analyses, including the temporary ones, are logged in the analysis log. The temporary ones are marked as "temporary".

If two analysis parameter files give the same quality, the first one is used. In other words, in the [...] you should first specify the file that you would most like to use, followed by the second best, etc.

You can specify any kind of parameter file in between [...]. They can be .PAR files, or .TAP files, or matrix files, etc.