DBControl for FTK 2.0 – 4.1

Introduction: DBContol is a command line database administration tool, used to manipulate the database(s) used by AccessData's products. The following explains how to use the versions of DBControl shipped with FTK 2.0 - 4.1. Full documentation for DBControl can be found at http://wiki.accessdata.dev/index.php?title=Dbcontrol, but the commands listed are the ones most commonly used with FTK.

Notes:

- You should always use the version of DBControl shipped with the version of FTK whose data you wish to manipulate
- The DBControl executable is usually located at "C:\Program Files\AccessData\Forensic Toolkit\[version]\bin\ "

Usage:

dbcontrol [param=value ...] [-noprompt] -command [args ...]

Parameters

Parameter	Value	Description
host=	<IP or Host Name>	Specifies the location of database. "Localhost" is assumed if not specified.
port=	<Port>	Specifies the database port. Default ports are assumed if not specified.
sysuser=	<System Username>	Specifies the database system username. Default usernames are assumed if not specified.
syspass=	<System Password>	Specifies the database system password. If one is needed but not specified, you will be prompted to enter it.
pgdb=	<SID>	Specifies the SID for PostgreSQL. This required when using PostgreSQL. The SID for these versions of FTK should always be "FTK2".

Commands

Command	Argument	Description
-noprompt		Hides any prompts.
-getavailableschemas		Lists all valid FTK case schemas existing on the current server.
-createschema	ADMS|<Shared Schema>	Creates the specified schema.
-deleteschema	ADMS|<Shared Schema>|<Case Schema>	Deletes the specified schema.
-reinstall	<Shared Schema>	Deletes then creates the specified schema.
-validate	<Shared Schema>|<Case Schema>	Validates the specified schema, attempting to fix non-critical errors.
-copyschema	See below for arguments and usage.	Migrates a case between different FTK versions. Only works when

		both FTK versions use the same database.
-detach	See below for arguments and usage.	Despite the name, this creates a full backup of an existing case and does not detach the case.
-attach	See below for arguments and usage.	Despite the name, this restores a case from a backup.
-cputime	See below for arguments and usage.	Outputs a log of Oracle's activity.

Usage for –copyschema

-copyschema <From Case Schema> <To Case Schema> <TEMP Path>

Argument	Acceptable Values	Description
<From Case Schema>	<Case Schema>	The original case's schema. This argument is required.
<To Case Schema>	<Next Case Schema>	The schema for the case to be migrated to. This argument is required.
<TEMP Path>	<Folder Path>	Folder to temporarily store data during case migration. Quotes must be used if the path has spaces. This argument is required.

Usage for –detach

-detach <Case Schema> <Output Folder>

Argument	Acceptable Values	Description
<Case Schema>	<Case Schema>	The case's schema. This argument is required.
<Output Folder>	<Folder Path>	Folder to store the case backup. This folder should not already exist. Quotes must be used if the path has spaces. This argument is required.

Usage for –attach

-attach <To Case Schema> <Source Folder>

Argument	Acceptable Values	Description
<To Case Schema>	<Next Case Schema>	The schema to restore the case to. This argument is required.
<Source Folder>	<Folder Path>	Folder containing the backup to be restored. The path should be to the folder that contains the most recent archive's "Detach.xml", which is usually the "DB f-0" folder. Quotes must be used if the path has spaces. This argument is required.

Usage for –cputime

-cputime <Output File> -c <Interval>

Argument	Acceptable Values	Description
<Output File>	<File Path>	File to output data to. This should have the extension HTML and should not already exist. Quotes must be used if the path has spaces. This argument is required.
-c <Interval>	<Seconds>	Specifies the interval, in seconds, at which the output file will be updated. If not used, the output file is updated every second.

Explanation of Schema Parameters

Parameter	Format	Example
<Shared Schema>	FTK 2.0 – 3.3: FTK_[FTK Version] FTK 3.4: FTK_40 FTK 4.0: FTK_41 FTK 4.1: FTK_42	Shared schema for FTK 3.3: FTK_33
<Case Schema>	<Shared Schema>_C<Case ID>	Case schema for case 1001 in FTK 4.1: FTK_42_C1001
<Next Case Schema>	<Shared Schema>_CNEXT	Next case in FTK 3.4: FTK_40_CNEXT

Examples:

Create the FTK 4.0 shared schema, using Oracle:

```
dbcontrol -createschema FTK_41
```

Delete case 1005 from FTK 3.4, using Oracle:

```
dbcontrol -deleteschema FTK_40_C1005
```

Delete and rebuild the FTK 3.4 shared schema, using PostgreSQL:

```
dbcontrol pgdb=FTK2 -reinstall FTK_40
```

Validate case 1010 in FTK 3.3, using Oracle at 192.168.1.150:

```
dbcontrol host=192.168.1.150 -validate FTK_33_C1010
```

Validate the FTK 4.1 shared schema, using PostgreSQL:

```
dbcontrol pgdb=FTK2 -validate FTK_42
```

Migrate case 1005 from FTK 4.0 to FTK 4.1, using Oracle, using C:\Temp for temporary storage:

```
dbcontrol -copyschema FTK_41_C1005 FTK_42_CNEXT C:\Temp
```

Archive case 1003 from FTK 4.1, using PostgreSQL, to C:\Cases\1003\DB f-0:

```
dbcontrol pgdb=FTK2 -detach FTK_42_C1003 “C:\Cases\1003\DB f-0”
```

Attach a case to FTK 4.0, using Oracle, from an archive at C:\Case\1050\DB f-0:

```
dbcontrol -attach FTK_41_CNEXT “C:\Cases\1050\DB f-0”
```

Create a log of Oracle’s activity, updating every 5 seconds, stored at C:\OracleLog.html:

```
dbcontrol -cputime C:\OracleLog.html -c 5
```