

# DZbatcher

## **Installation & User Guide**

© Datazone 2010

DZbatcher@datazone.com

[Please read the license agreement on the next page  
before using this software]

## License Agreement and Conditions of Use

You are hereby licensed to use this software program, entitled *DZbatcher*, with any licensed version of Adobe FrameMaker™ version 5.5.6 or above which you have purchased and installed on a Microsoft Windows NT 4, Windows 2000, or Solaris system, **without charge**, subject to your accepting all the terms and conditions contained in this License Agreement.

**THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR ANY PURPOSE.**

**IN NO EVENT SHALL DATAZONE OR ADOBE BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.**

You are *not* entitled to use this software in any other way than described above. You may not transfer this software to any third party. This license agreement does not cover any future upgrades or new versions of this software that may be produced. You are not entitled to any support or maintenance services, or bug-fixes, as a result of using this free software under this License Agreement. If you or others within your organization, or outside of it, require additional copies of DZBatcher, each copy must be downloaded from this site: [www.miramo.com](http://www.miramo.com).

All users of this software are advised to make backup copies of their documents prior to running this software, in order to minimize the risk of data loss. This software has not been subjected to commercial-grade quality testing.

Your comments and criticisms concerning this software will always be welcomed. Please send these to:

[DZbatcher@datazone.com](mailto:DZbatcher@datazone.com)

Trademarks: *Adobe, the Adobe logo, FrameMaker, PostScript, Acrobat* are all trademarks of Adobe Systems Incorporated. *Microsoft Windows and Windows NT* are trademarks of Microsoft Corporation. *Datazone, Miramo, MergeMaker* are trademarks of Datazone Ltd. *Solaris and SunOS* are trademarks of Sun Microsystems.

For full details of the *Miramo* high-end database publishing toolkit, check out [www.miramo.com](http://www.miramo.com) or send an email message to [miramo@datazone.com](mailto:miramo@datazone.com)

## License Agreement and Conditions of Use

## CONTENTS

<b>What is DZbatcher?</b>	<b>5</b>
<b>Installation</b>	<b>7</b>
Installation overview	
Windows	7
Unix	7
Environment variables	8
<b>User Guide</b>	<b>11</b>
Command line options	12
Interactive operation	13
Batch operation	13
Summary of commands	14
Command details	16
Close	16
DeleteFormats	17
echo	18
Exit	18
help or ?	19
ImportFile	20

# Contents

ImportFormats	21
ListOpen	22
Open	22
OpenTextFile	23
ping	23
Print	23
Save	25
SaveAs	26
System	27
Update	27
Example of DZbatcher command file	28
Using DOS .bat files	29
<b>Appendix</b>	<b>32</b>
DZbatcher bugs	32

# *What is DZbatcher?*

*DZbatcher* is a command-driven console application which allows a user to enter simple, line-oriented commands in a ‘DOS prompt’ window or Unix terminal/Xterm, enabling single, or batches of, FrameMaker or FrameMaker+SGML documents to be opened, updated, printed and saved without being displayed on-screen.

*DZbatcher* may be used with a *DZbatcher* command file to perform batch operations on a set of FrameMaker or FrameMaker+SGML books or documents, without further user intervention.

You can use *DZbatcher* to perform the following batch operations:

- | Print documents using a ‘print settings’ document to control how the documents will be printed. For example: with registrations marks, as separations, with Acrobat settings, first sheet last or first, to printer, or to file, etc.
- | Batch convert Frame files to Acrobat PDF or RTF format.
- | Convert files from Frame document (binary) format to Frame MIF (Maker Interchange Format) or *vice versa*. You can also batch-convert files to FrameViewer or FrameMaker locked format.

## Summary

- | Import some or all formats from one document into many other documents.
- | Automate the updating of variables, cross-references and text insets across a large collection of documents.

Be sure to read the section describing **DZbatcher bugs** before using *DZbatcher*.

# Installation

This chapter describes how to install *DZbatcher* on a system running Windows NT 4, Windows 2000, or Solaris 2.6, and FrameMaker or FrameMaker+SGML version 5.5.6 or greater.

Note that to run *DZbatcher* on a Unix system, the Unix host must have access to an X-display, denoted using the `$DISPLAY` environment variable.

## Installation [Windows NT]

Log in as the system administrator and download and run the appropriate `dzbatcher-setup.exe` file.

Setup will guide you through the installation process.

The default installation folder is `C:\Program Files\dzbatcher`, but you may select a different target folder during installation. If the target installation folder does not exist, setup will create it for you.

### 'Setup' will perform the following steps:

- 1 Selects or creates the folder and extracts the files
- 1 Sets the 'PATH' environment variable to include the *DZbatcher* 'bin' folder.

## Installation [UNIX]

If you are downloading from the WEB, download the ‘dzbatcher.tar’ file to a directory on your local drive.

Use the command

```
tar xvf installdir/dzbatcher.tar
```

to extract the ‘bin’, ‘samples’ and ‘documentation’ subdirectories into the current directory. If necessary, update the ‘PATH’ environment variable to include the ‘bin’ directory (which contains the dzbatcher binary).

## Moving files to other locations

There may be no need to move any of the *DZbatcher* files, however you may do so if you wish. The only consideration is that the files in the *installdir/bin* folder must be on *DZbatcher* users’ paths, changed by setting the ‘PATH’ environment variable.

## Environment variables

The table below describes environment variables which may be used to configure the default operation of *DZbatcher*.

## Environment variables (continued)

Name	Description
DZB_MAKERNAME	<p>Name of Frame product binary to invoke to start Frame session.</p> <p>To use <i>DZbatcher</i> with FrameMaker+SGML, set this environment variable to framemaker+sgml (Windows) or makersgml (Unix).</p> <p>See also the ‘-makename’ command line option on page 12.</p> <p>Default for DZB_MAKERNAME is: framemaker (Windows) maker (Unix)</p>
DZB_RETRIES	<p>Number of connection attempts made to Frame session before timing out. Default for DZB_RETRIES is: 20</p>
FMHOME (Unix only - REQUIRED)	<p>Frame product installation directory. Fatal error if not set.</p>
DISPLAY (Unix only)	<p>Name of X-display to use for running Frame session. Default for DISPLAY is: unix:0</p>

### Environment variables (continued)

Name	Description
DZB_XDISPLAYTIMEOUT (Unix only)	Number of seconds to wait before timing out on opening \$DISPLAY. Default value: 4

### DZBatcher version

Under both Windows and Unix platforms the version number of DZbatcher available can be determined by the following command:

`dzbatcher -l`

(the command line flag ‘-l’ is ‘-’ followed by the letter ell).

# User Guide

*DZbatcher* provides a command-line interface to FrameMaker or FrameMaker+SGML, allowing files to be opened, updated, printed, reformatted and saved without displaying them on-screen.

For example, the following list of *DZbatcher* commands opens, updates and prints a Frame document called ‘tutorial.fm’, then saves the document and exits the *DZbatcher* session:

- Open tutorial.fm
- Update tutorial.fm
- Print tutorial.fm
- Save tutorial.fm
- Exit

*DZbatcher* is intended to provide similar functionality to the UNIX *fmbatch* utility supplied with UNIX versions of FrameMaker and FrameMaker+SGML.

This document covers the following areas:

- | using *DZbatcher* for interactive or batch operation
- | *DZbatcher* command descriptions

## Media contents (continued)

- 1 calling *DZbatcher* from DOS .bat files (Windows) or shell scripts (UNIX).

## Command line options

Under Windows *DZbatcher* must be run from a Command Prompt window, either directly or via a DOS ‘.bat’ script. A Command Prompt window may be started from the Start→Programs menu. Under Unix *DZbatcher* must be run from a terminal or Xterm, either directly or via an executable shell script.

The usage summary for *DZbatcher* is as follows:

```
DZbatcher [-v] [-makename framemaker+sgml] [cmd_file]
```

where

-v

causes each command to be echoed as it is executed

-makename framemaker+sgml

Selects FrameMaker+SGML rather than FrameMaker. (See also the description of the **DZB\_MAKERNAME** environment variable in the ‘Installation’ section of this guide. )

cmd\_file

specifies the name of a file containing a list of *DZbatcher* commands.

If no cmd\_file is specified, *DZbatcher* reads commands from standard input.

## Interactive operation

If no cmd\_file name is specified, *DZbatcher* starts a Frame session and displays the following prompt.

```
dzbatcher->
```

In this case, the user works interactively, entering *DZbatcher* commands directly from the keyboard. *DZbatcher* interprets the commands and uses them to drive the Frame product session. When *DZbatcher* encounters a Exit command it terminates the Frame session and exits.

## Batch operation

If a cmd\_file name is supplied as an option to *DZbatcher*, commands are read from the cmd\_file and executed. *DZbatcher* terminates when it encounters a Exit command, or the end of the command file, whichever comes first.

Batch operation is useful when the same set of commands is to be used frequently.

It is also possible to use a DOS .bat file (Windows) or a Shell

script (Unix) to build an *DZbatcher* command file, where batch processing commands such as `for` and `if` may be used in conjunction with the ‘\*’ or ‘?’ wildcards to conditionally apply a set of *DZbatcher* commands to a selected set of Frame documents. See “Using DOS .bat files (Windows)” on page 29, and “Using Shell Scripts (Unix)” on page -30.

## Summary of commands

The following table gives a list of *DZbatcher* commands.

Each command must begin on a new line. All commands other than the `echo` and system commands occupy a single line.

`Close [-s] filename(s)`

Closes *filename(s)* without saving or, using the optional ‘-s’ flag, saves files and then closes them. See: **Close**

`DeleteFormats [-pftcv] filename(s)`

Deletes selected formats from *filename(s)*  
See: **DeleteFormats**

`echo message`

Prints message to standard output See: **echo**

`Exit`

Terminates the *DZbatcher* session See: **Exit**

`help or ?`

Displays list of *DZbatcher* commands  
See: **help or ?**

## Summary of commands (continued)

ImportFile [-B] [-E] [-I | -P | -Cop] [-f “flowtag”] *fromfile tofile(s)*

Imports the contents of flow tagged “flowtag” (default is flow “A”) from one open document, *fromfile*, at the beginning (-B) or end (-E) of another open document.

See: [ImportFile](#)

ImportFormats [-pflcvrtxkmBO] *fromfile tofile(s)*

Copies all, or selected, format definitions from one open document, *fromfile*, into another open document, or several other open documents, — *tofile(s)*

See: [ImportFormats](#)

ListOpen

Displays all the files which are currently open in this *DZbatcher* session. See: [ListOpen](#)

Open *filename(s)*

Opens documents or book files See: [Open](#)

OpenTextFile -a | -t *filename(s)*

Opens text files See: [OpenTextFile](#)

ping

Checks that *DZbatcher* is able to communicate with the Frame session. See: [ping](#)

Print [-p *printsettingsfilename*] *filename(s)*

Prints open files using their own print settings, or by specifying the name of another open file which contains different print settings. See: [Print](#)

Save *filename(s)*

Saves open files See: [Save](#)

**Summary of commands (continued)**

- SaveAs [-m | -a | -t | -d | -l | -p | -x] *openfile savefile*  
 Saves an open file under a different, name optionally in a different format See: **SaveAs**
- system *system-command*  
 Execute DOS command See: **System**
- Update *filename(s)*  
 Updates cross references and generates all files in an open book or document See: **Update**
- #  
 Treat remainder of line as a comment

**Command details**

The *DZbatcher* commands are described in detail below.

**Close** Closes a document, without saving any changes made during the *DZbatcher* session. If the ‘-s’ is used the document or book is saved before being closed.

-> Close [-s] *filename(s)*

EXAMPLE

- DZbatcher-> Open tutorial.fm # Opens tutorial.fm
- DZbatcher-> Close tutorial.fm # Closes tutorial.fm  
 # without saving it first
- DZbatcher-> Open tutorial.fm # Opens tutorial.fm
- DZbatcher-> Close -s tutorial.fm # Closes tutorial.fm  
 # saving it first
- DZbatcher-> Exit # Exits DZbatcher

## Command details (continued)

**DeleteFormats** Deletes formats from the paragraph, character, variable and cross-reference definition catalogs.

-> DeleteFormats [-pftcv] *filename(s)*

Using DeleteFormats with no flag option results in *all* formats being deleted from the paragraph, character, variable and cross-reference definition catalogs, with the exception of user variable and cross reference formats which are used within the body of the document. System variables are never deleted.

Note that the appearance of paragraphs, characters or tables within the document will not change if their associated paragraph, character or table formats are deleted.

For example, using 'DeleteFormats' to delete paragraph definitions will leave the appearance of paragraphs tagged as 'Heading 1' unchanged, however the 'Heading 1' definition will no longer be available within the paragraph catalog.

You can select a subset of formats to delete using one or more of the flag options shown in the table below. The string comprising one or more flag options must be immediately preceded by the

**Command details (continued)**

minus, '-', character.

<b>Option</b>	<b>Description</b>
p	All paragraph formats
f	All character formats
t	All table formats
c	Cross-reference formats not used in the document.
v	User variable formats not used in the document. (System variables are never deleted).

**EXAMPLE**

DZbatcher-> Open tutor.fm

DZbatcher-> DeleteFormats tutor.fm

**echo** Prints a "message" string to standard output. The message text must be surrounded by double quotes and may extend across several lines.

-> echo "message"

**EXAMPLE**

DZbatcher-> echo "About to open:

> tutorial.fm

> abc.fm"

**Exit** Terminates the *DZbatcher* session

## Command details (continued)

-> Exit

EXAMPLE

DZbatcher-> Exit

**help or ?** Displays the *DZbatcher* help message.

-> help

EXAMPLE

DZbatcher-> ?

Commands:

Close [-s] filename(s)-to-quit

DeleteFormats [-pfvtc] filename(s)

Exit

ImportFormats [-pflcvrtxkmbODs] formats-file filenames(s)

ImportFile [-BE] [-ICPop] -f flowtag fromfile tofile(s)

ListOpen

Open filename(s)

OpenTextFile -a | -t filename(s)

Print [-p print-options-filename] filename(s)

Save filename(s)

SaveAs [-m | -a | -t | -d | -l | -p | -r | -h | -x | -j ] source-file dest-file

Update filename(s)

Update [-afmrisGFMAPSRTC] book(s)

?

help

echo [one or more args to echo]

ping

system system-command

DZbatcher->



## Command details (continued)

Option	Description
C	Applies the formatting from the <i>tofile</i> .
o	Retains format overrides when -C option is used.
p	Retains manual page breaks when -C option is used.

The third level of options [-f“flowtag”] allows a text flow with a specified flowtag to be imported from *fromfile* into *tofile*.

Option	Description
f“flowtag”	Denotes a specific flowtag to be imported from <i>fromfile</i> to <i>tofile</i> [default: main document flow, normally “A”]

All the above option flags are optional, so:

-> ImportFile *fromfile tofile*

is the same as:

-> ImportFile -E -I -f“main document flow” *fromfile tofile*

**ImportFormats** Copies formats from one open document to other open documents.

-> ImportFormats [-pflcvrtxkmBO] *fromfile tofile(s)*

Using ImportFormats with no key letters results in *all* formats being copied into the destination documents. You

## Command details (continued)

can select a set of formats to copy using one or more of the key letters shown in the table below. The string comprising one or more key letters must be immediately preceded by the minus, '-', character.

Option	Description
B	Preserve manual page breaks
c	Cross-references
f	Character formats
k	Color definitions
l	Page and column layouts
m	Math definitions
O	Preserve format overrides
p	Paragraph formats
r	Reference pages
t	Table formats
v	Variable definitions
x	Conditional text settings

**ListOpen** Lists all the documents which are currently open during this *DZbatcher* session.

-> ListOpen

EXAMPLE

DZbatcher-> ListOpen

**Open** Opens a Frame document or book file

**Command details (continued)**

-> Open filename

EXAMPLE  
DZbatcher-> file.fm?

**OpenTextFile** Opens a text file

-> OpenTextFile -a | -t *filename(s)*

*filename* is the name of the text file to open. Use ‘a’ or ‘t’ to specify how line endings should be treated, as follows:

<b>Option</b>	<b>Description</b>
-a	Treat each line as a separate paragraph
-t	Merge successive text lines into a single paragraph, treating empty lines as paragraph breaks.

EXAMPLE  
DZbatcher-> OpenTextFile -a file.txt# Opens file.txt  
# converting each line to  
# a Frame paragraph

**ping** Checks that the *DZbatcher* Frame session is still alive.

If the Frame session is not alive and error is printed and the *DZbatcher* session terminates.

-> ping  
EXAMPLE  
DZbatcher-> ping

**Print** Prints open documents.

## Command details (continued)

‘Print’ prints an open document either using the document’s own print settings or those contained in the *printsettingsfilename* specified using the optional ‘-p’ flag.

The print settings used are as follows, and are displayed on the print dialog box for the given file:

### Print settings

print page range [only first page or *all* pages]

odd/even numbered pages

collate

last sheet first

skip blank pages

spot color as black/white

low resolution images

registration marks

thumbnails (rows and columns)

number of copies

scale factor

print to file

print separations

generate acrobat data

Certain combinations of settings are not allowed: for example, setting ‘thumbnails’ disables odd/even numbered pages, collate, last page first, number of copies and scaling. Similarly checking ‘Generate Acrobat Data’ disables thumbnails, skip blank pages and last sheet first, and forces printing to a file.

Under Windows, *print page range start and end page*

## Command details (continued)

*values always reset to the first page in the document when a document is opened.* Using *DZbatcher* it is therefore possible to print either the *first* page, or *all* pages in a document.

Similarly, the printer name always resets to the default system printer when a document is open, so it is possible to print only to the default printer using *DZbatcher*.

If ‘print to file’ is specified, *DZbatcher* will always print to the default file postscript file name appropriate to the document being printed, even if a print settings file is specified using the ‘-p’ flag. For example, a document ‘abc.fm’ would always be printed to a PostScript file called ‘abc.ps’ in the same folder. Please note that the file *printsettingsfilename* must be a FrameMaker binary file, but the file to be printed (*filename(s)*) can be a FrameMaker binary or a Maker Interchange File (MIF).

-> Print [-p *printsettingsfilename*] *filename(s)*

EXAMPLE

DZbatcher-> Open file.fm

DZbatcher-> Open printtofile.fm

DZbatcher-> Print -p file.fm printtofile.fm # Prints file.fm using  
settings

# from printtofile.fm

**Save** Saves an open file, without closing it

## Command details (continued)

-> Save *filename(s)*

EXAMPLE

DZbatcher-> Open tutor.fm

DZbatcher-> Update tutor.fm# update cross references in tutor.fm

DZbatcher-> Save tutor.fm# save tutorial.fm

**SaveAs** Saves an open file under a different name and/or file format.

-> SaveAs [-m | -a | -t | -d | -l | -p | -x] *fromfile tofile*

*fromfile* specifies the name of an open file or book, *tofile* is saved in the current format of *fromfile*, unless one of the optional format flags is used.

The formats specified by these flags is shown in the table below:

Option	Description
-m	MIF format
-a	Text format, one line per line
-t	Text format, one line per paragraph
-d	Normal (document) format
-l	Locked (view-only) format
-p	PDF format
-x	XML format
-r	RTF format
-h	HTML format
-j	RTF (Japanese)

EXAMPLE

Open test.fm

**Command details (continued)**

SaveAs -m test.fm test.mif# Save test.fm as MIF file test.mif  
 SaveAs -l test.fm test.mif # Save test.fm as locked document

**System** Executes a DOS command or a UNIX Shell command.

-> system *command and parameters*

EXAMPLE

# --- Copy Frame document to floppy diskette  
 DZbatcher-> system "copy test.fm a:"

**Update** Updates variables, cross references and text insets in a document or book.

-> Update *filename(s)*

-> Update [-afmrisGFMAPSRTC] book(s)

If a book file is specified, *DZbatcher* performs the same operation as choosing the Generate/Update option from the Book File menu. Generated file types may be excluded from update using the options given in the table below:

<b>Option</b>	<b>Generated file type to exclude from update</b>
-a	Author
-f	Index of Formats
-m	Index of Markers
-r	Index of References
-i	Standard index
-s	Subject index
-G	List of figures
-F	List of formats

## Command details (continued)

Option	Generated file type to exclude from update
-M	List of markers
-A	List of markers in alphabetic order
-P	List of paragraphs
-S	List of paragraphs in alphabetic order
-R	List of references
-T	List of tables
-C	Table of contents

### EXAMPLE

```
DZbatcher-> Open test.fm
DZbatcher-> Update test.fm
DZbatcher-> Save test.fm
```

## Example of DZbatcher command file

The following is an example of an *DZbatcher* command file.

```
# -----
# - Open book file, template file and book component files
# - Import all formats from template file to book component
# - files, update the book to renumber pages and update
# - cross references, then print the book components.
# - Finally close and save the book and all files
Open radio.book
Open smalltemplate.fm chapter1 chapter2 chapter3
ImportFormats smalltemplate.fm chapter1 chapter2 chapter3
Update radio.book
```

### Example of FMbatcher command file (continued)

```
Print -p smalltemplate.fm chapter1 chapter2 chapter3
Close -s radio.book
Close -s smalltemplate.fm chapter1 chapter2 chapter3
```

Assuming the above is contained in a file called 'example.fmb', then it can be run using the command:

```
DZbatcher -v example.fmb
```

(The -v flag will echo each command as it is executed.)

### Using DOS .bat files (Windows)

The following DOS batch file saveasmif.bat creates a temporary *DZbatcher* command file called 'dzb\_script' which is used to save a set of Frame documents as MIF. The set of files to be processed is given as a list of parameters to the saveasmif.bat batch file, and may include wildcards such as "\*.fm" or "???.book"

```
REM -----
REM --- saveasmif.bat: creates a temporary DZbatcher
REM --- command file and executes it
REM -----
@echo off
rem --- Delete DZbatcher command file if it exists
if exist dzb_script del dzb_script
REM --- Create command script to process each
argument
:STARTLOOP
if "%1" == "" goto ENDLOOP
```

## Using DOS .bat files (continued)

```

for %%f in (%1) do echo Open %%f >> dzb_script
for %%f in (%1) do echo SaveAs -m %%f %%f.mif >> dzb_script
for %%f in (%1) do echo Exit %%f >> dzb_script
shift
goto STARTLOOP
:ENDLOOP
DZbatcher -v dzb_script
del dzb_script

```

### EXAMPLE

```
saveasmif *.fm abc.book
```

## Using Shell Scripts (Unix)

The following Unix shell script `saveasmif` creates a temporary DZbatcher command script which is used to save a set of Frame documents as MIF. The set of files to be processed is given as a list of parameters to the `saveasmif` shell script, and may include wildcards such as `"*.fm"` or `"???.book"`

```

#!/bin/sh
# -----
# --- saveasmif.sh - creates a temporary DZbatcher command file
# --- to save arguments as MIF, and executes it
# -----

# --- Delete DZbatcher command file if it exists
rm -f dzb_script$$

```

**Using DOS .bat files (continued)**

```
# --- Create temporary command script to save each file as MIF
for i in $*
do
    echo “
    Open $i
    SaveAs -m $i $i.mif
    Close $i
“ >> dzb_script$$
done
```

```
# --- Execute temporary command script
dzbatcher -v dzb_script$$
```

```
# --- Remove temporary command script
rm dzb_script$$
```

```
EXAMPLE
saveasmif *.fm abc.book
```

# Appendix

## DZbatcher bugs

The following are bugs in *DZbatcher*:

- | *DZbatcher* has not been properly tested. For this reason **always make sure that you have backed up your documents before using *DZbatcher*.**
- | Opening too many files at once can overload or crash your system.
- | Under Windows, The GUI (outside the DOS-prompt) is subject to the ‘splashes and flashes’ syndrome.
- | Wildcards are not supported.