

# Using the Open Toolkit plug-in for FrameMaker

## TABLE OF CONTENTS

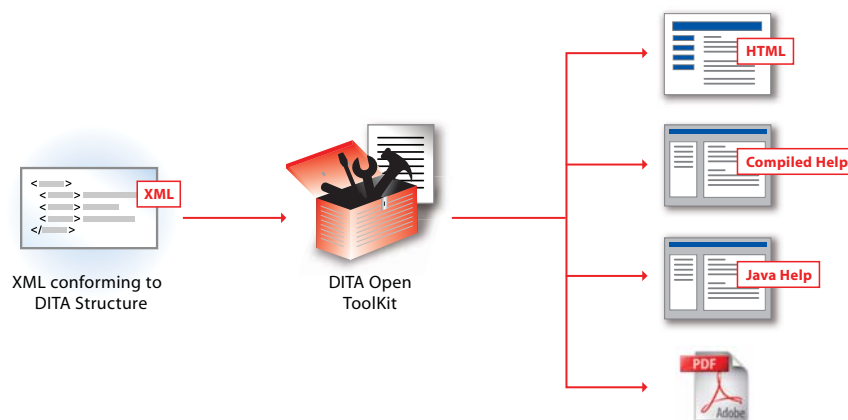
- 1 Overview of DITA Open Toolkit
- 1 Working with the Open Toolkit plug-in for FrameMaker
- 3 Advanced features
- 9 Appendix 1: Installing the DITA Open Toolkit
- 13 Appendix 2: Installing the Open Toolkit plug-in for FrameMaker
- 14 Next steps
- 14 Resources

With the Darwin Information Typing Architecture (DITA) emerging as the de facto standard for creating topic-based content, Adobe® FrameMaker® 9 provides extensive support for authoring DITA content. In addition, you can use the DITA Open Toolkit with FrameMaker to obtain deliverable outputs. This document describes how to use the DITA Open Toolkit with FrameMaker 9.

You are expected to know how to use DITA in FrameMaker. For information on installing DITA Open Toolkit, see “[Appendix 1: Installing the DITA Open Toolkit](#)” on page 9, and “[Appendix 2: Installing the Open Toolkit plug-in for FrameMaker](#)” on page 13.

## Overview of DITA Open Toolkit

DITA Open Toolkit is a set of software tools, originally developed by IBM®, for publishing DITA to various formats, including HTML, Microsoft HTML Help (.CHM) and Java™ Help. If your content conforms to the DITA structure, you can use the DITA Open Toolkit to generate a wide variety of outputs.

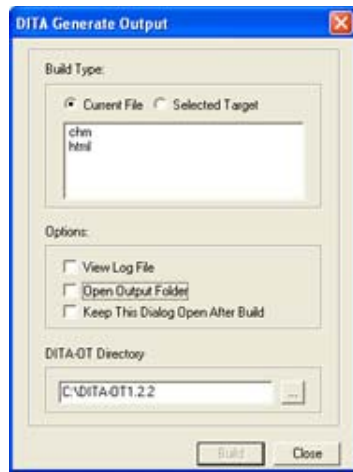


*The DITA Open Toolkit—an Open Source tool chain for multichannel publishing*

DITA Open Toolkit is available for free distribution on the Organization for the Advancement of Structured Information Standards (OASIS) website at [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=dita](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=dita). OASIS maintains the DITA specification.

## Working with the Open Toolkit plug-in for FrameMaker

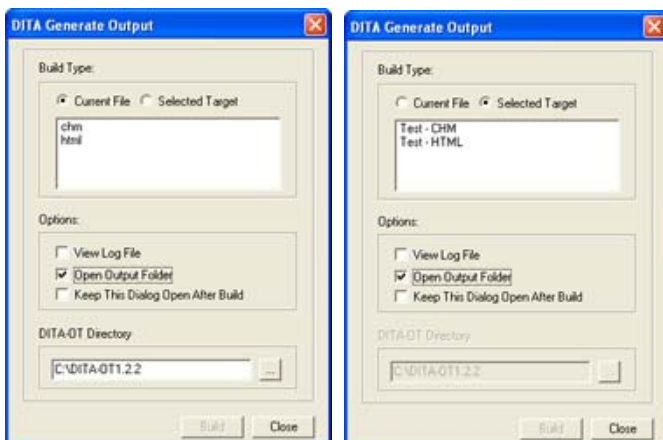
The Open Toolkit plug-in for FrameMaker lets you manage publishing through the FrameMaker user interface. The plug-in provides a DITA menu for publishing multiple outputs through the DITA Open Toolkit, using the default options or your customized builds.



The Open Toolkit plug-in for FrameMaker

### Generating output from the current file

After installation, you can run the DITA Open Toolkit processes from within FrameMaker. There are two ways to do this, corresponding to the two options, Current File and Selected Target in the DITA Generate Output dialog box:



The DITA Generate Output dialog box offers two forms of DITA Open Toolkit processing.

- **Current File** uses a pre-configured Ant script: This Ant script is the ditafm-ant.xml file that you copied to your DITA-open toolkit folder on installation. This script generates output from a topic that is currently open or from the map file to a sub folder of the folder containing the topic. On a configured system, you can generate output without having to modify the Ant script. However, you can edit the ditafm-ant.xml file to customize the processing of output. You must specify the path where you installed the DITA Open Toolkit.
- **Selected Target** lets you specify an Ant script you've created. You must set up your own Ant script and reference the script in the ditafm.ini file. The list of selected targets does not appear until you define them in the Ant script and reference them in the ditafm.ini file.
- Most users may require only the Current File method starting the DITA open toolkit. Most FrameMaker users may use the Current File method for running the DITA Open Toolkit processes.

You can use Current File to process the map file or DITA topics that are open.

In the Selected Target method, the Ant scripts referenced in your DITA initialization file are run. This method indicates the following:

- Files to process
- Applications and processes to run
- Destination folders and filenames

Thus, the Selected Target method is more appropriate for calling publishing processes that generate output from source files in the locations specified in your custom Ant scripts.

## Current File Method

Make the following changes in `fminit\ditafm\ditafm.ini`

To process a topic file that is open, change the `ditafm.ini` file as follows:

```
[BuildFile]
AntCommand=ant
AntScript=ditafm-ant.xml
Count=2
DitaDir=C:\DITA-OT1.4.2.1
1=chm
2=html
```

To generate output from the example Help file DITA map, using the Current File option:

- 1 Open a ditamap file.
- 2 If prompted, select the structured application DITA-Map-HTML, and click Continue.



Use the Structured Application dialog box.

- 3 Select the path where the DITA Open Toolkit is installed.
- 4 Select the Current File option.
- 5 Select `.html`.
- 6 Select the output folder, and click Build.

On a fully configured system, the example DITA content is saved in HTML.

### HTML output

To view the HTML output, double-click the `index.html` file, which is located in the output folder.

### Compiled Help (.chm) output

You can generate compiled Help (`.CHM`) easily. This process is similar to generating HTML output. Open the `.ditamap` file, and choose DITA-->Generate CHM Output. A folder named CHM is created in the output folder. The CHM folder contains the CHM file.

When you use the Current File option to generate output using the DITA Open Toolkit, you do not need to modify any of the Ant, XSLT, or initialization files.

### PDF output

The recommended way to generate a PDF for DITA files is through the Save As PDF option in FrameMaker.

However, to generate PDFs using the DITA Open Toolkit:

- 1 Install a FO processor that helps in converting FO files to PDF. (Apache FOP is free.)
- 2 With Apache FOP installed, add `fop.jar`, `batik.jar`, `avalon-framework-cvs-20020806.jar` file path to the CLASSPATH environment variable.
- 3 Update `ditafm.ini` to add an entry for PDF.

The route to generate PDFs through DITA Open Toolkit is added to the DITA Generate Output dialog box.

## Advanced features

The publishing capabilities of Adobe FrameMaker are enhanced by the Open Toolkit plug-in for FrameMaker. To take advantage of these features, you must have knowledge of the following:

- The settings in the DITA initialization file (ditafm.ini), which determine the Apache Ant file to be used for different target outputs.
- Apache Ant scripts, in general, and their role in defining publishing processes that generate output from the DITA Open Toolkit.
- The XSLT style sheets that are used to generate HTML and other forms of output with the DITA Open Toolkit, as well as the other files that control the format of output, such as cascading style sheets.

### Using the Selected Target option

A user can customize using the selected target option. The ditafm.ini file provides users with entry points from where they can run various Ant scripts. Also, for customizing Ant scripts, the user also needs to have a basic understanding of these scripts.

When you use the Selected Target option, rather than starting an Ant script that is preset to process the currently open file, you start an Ant script without parameters. The Ant script must contain explicit instructions about the following:

- Files to process
- Folders to be used for different forms of output
- Publishing methods and processes

While the Selected Target option requires more setup than the Current File option, it offers more fine-grained control.

### About Ant scripts

Apache Ant, a Java™ application installed with the DITA Open Toolkit, is used for automating software build processes. For example, the following types of publishing processes can be automated:

- 1 Transform specified XML files, such as DITA map files or associated topic files, with XSLT style sheets, to generate XHTML output.
- 2 Run a subsequent process to place these files into specific folders for deployment to a website.

With Ant, such processes are completely controlled by an Ant script or XML build file, which defines the following:

- All steps of the publishing process, including the processes or applications called, the target and destination folders and files
- Processing sequence

### Example custom Ant script

If the DITA Open Toolkit is properly installed, you should see a file named PROJECT.xml in your DITA\_OT folder, such as C:\dita-ot1.2.\PROJECT.xml. You can define any custom publishing processes by copying, renaming, and modifying the PROJECT.xml file. In addition, you must modify the DITA initialization file.

For example, to process DITA files located in the C:\Projects\Test folder, do the following:

- 1 Create a folder named C:\Projects\Test and copy all .dita and .xml files from the source folder to it. Source refers to the location where you extracted the OpenToolkit.zip file. See Appendix 2 for details.
- 2 Copy PROJECT.xml to your Open Toolkit (typically, C:\dita-ot1.4.2.1) folder and rename PROJECT.xml to Test.xml.
- 3 Open Test.xml in Windows® Notepad® (Start > Programs > Accessories > Notepad). In steps 4 through 9, you will set the main properties indicated in the comments, which appear at the beginning of the file.

```

Test.xml - Notepad
File Edit Format View Help
<?xml version="1.0" encoding="UTF-8" ?>
<!-- (C) Copyright IBM Corp. 2001, 2005 All Rights Reserved. -->
<!-- *****
Sample Ant script for use with the Adobe FrameMaker 7.2 Application Pack for DITA.
Be sure to set the following properties for your system/project:
- dita.dir
- project.ditamap
- project.dir
- project.output.dir
- dita.extname
***** -->
<project name="dita.build" default="prompt" basedir=".">

```

Test.xml open in Windows Notepad

4 Set the property value for dita.dir to your Open Toolkit path (for example, C:\DITA-OT1.4.2.1).

```

Test.xml - Notepad
File Edit Format View Help
<property name="dita.dir"
value="C:\DITA-OT1.2.2"/>
<property name="dita.script.dir"
value="${dita.dir}${file.separator}xs1"/>

```

Property name and value settings for dita.dir

5 Set the property value for project.ditamap to the name of your DITA map file, and the value for project.dir to the path of your project folder (in this case, C:\Projects\Test). If necessary, you can change the output folder. In the initial configuration of PROJECT.xml, the default output folder is the sub folder, out, in the folder defined as project.dir.

```

Test.xml - Notepad
File Edit Format View Help
<property name="project.ditamap"
value="ditafm.ditamap"/>
<property name="project.dir"
value="C:\Projects\Test"/>
<property name="project.output.dir"
value="${project.dir}${file.separator}out"/>

```

Property name and value settings for project.dita

6 Set the property value for dita.extname to the extension of your referenced topic files (in this case, .xml).

```

Test.xml - Notepad
File Edit Format View Help
property="HTMLHelpCompiler"
value="C:\Program Files\HTML Help
workshop\hhc.exe"/>
<property name="dita.extname" value=".xml"/>
<property name="clean.temp" value="true"/>
</target>

```

Property name and value settings for dita.extname

7 Save and close Test.xml.

At this stage, Test.xml must include all parameters required for running builds that target the specific test folders and files. However, such Ant scripts must be referenced in the initialization file before their targets show as options in the DITA Generate Output dialog box.

### Editing the DITA initialization file

To complete the process of adding the selected targets, edit the DITA initialization file.

**Note:** Take a backup of the DITA initialization file before changing anything.

1 Using Windows Notepad, open ditafm.ini from the fminit\ditafm folder.

2 Add the following lines at the end of the file:

```

[AntBuild]
AntCommand=ant
Count=2
1=Test - CHM
2=Test - HTML

[ANT:Test - CHM]
BuildFile=C:\DITA-OT1.4.2.1\Test.xml
Target=chm
OutputDir=C:\Projects\Test\out\chm
LogFile=C:\Projects\Test\ant-buildlog-chm.txt

[ANT:Test - HTML]
BuildFile=C:\DITA-OT1.4.2.1\Test.xml
Target=html
OutputDir=C:\Projects\Test\out\html
LogFile=C:\Projects\Test\ant-buildlog-html.txt

```

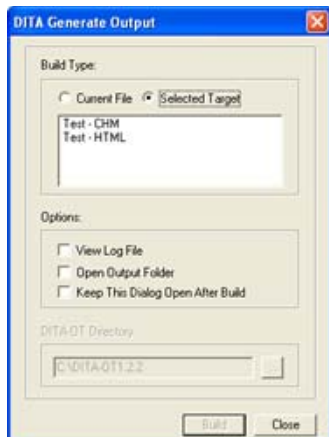
3 Save and close ditafm.ini.

Now you can run the selected targets. Note that Test.xml is a single script that contains multiple targets, such as CHM and HTML.

To run the selected targets you have defined (formats like CHM, PDF, and so on):

4 In FrameMaker, select DITA > Generate Output.

5 Click Selected Target. The list of targets specified in the .ini file appears.



*DITA Generate Output dialog box with Selected Target option selected*

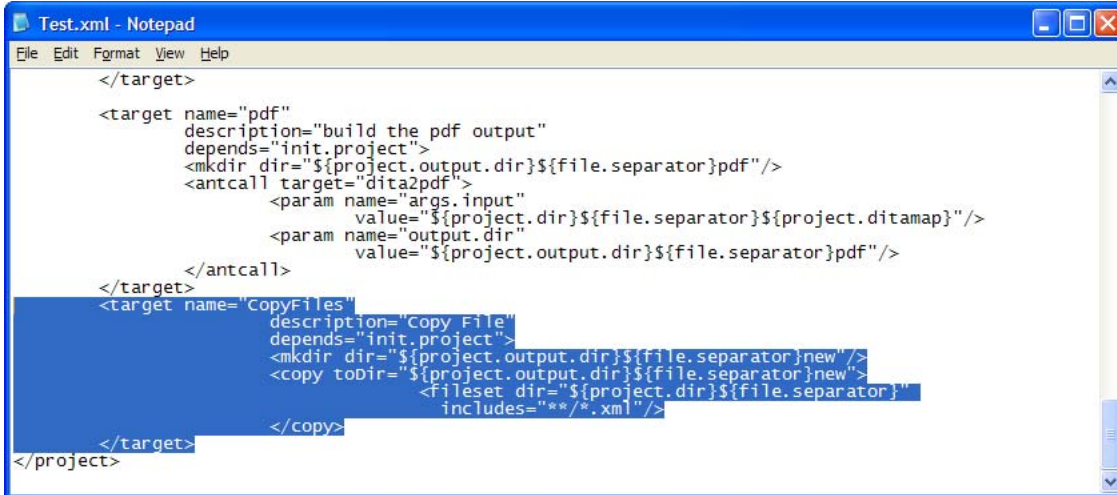
6 Select one of the targets and click Build.

You can review the output, which is similar to the output generated using the Current File option.

## Customizing Ant

You can customize Ant scripts to automate a wide range of publishing processes.

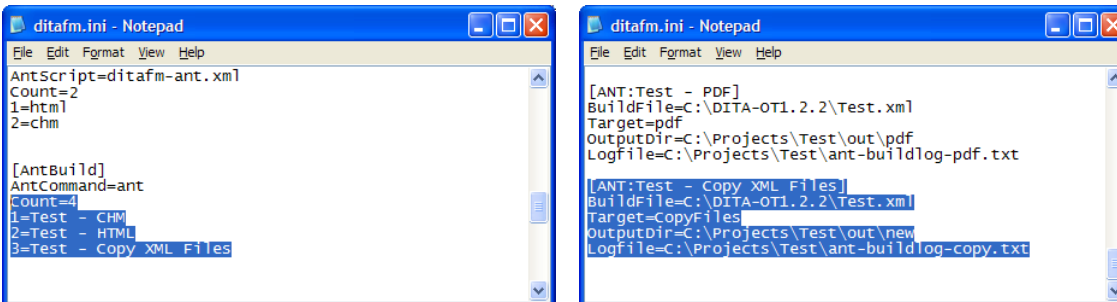
For example, you might use an Ant script to copy files from one location to another. You can either create a script or add a target to an existing script, as shown in the following figure:



```
</target>
<target name="pdf"
description="build the pdf output"
depends="init.project">
<mkdir dir="${project.output.dir}${file.separator}pdf"/>
<antcall target="dita2pdf">
<param name="args.input"
value="${project.dir}${file.separator}${project.ditamap}"/>
<param name="output.dir"
value="${project.output.dir}${file.separator}pdf"/>
</antcall>
</target>
<target name="CopyFiles"
description="Copy File"
depends="init.project">
<mkdir dir="${project.output.dir}${file.separator}new"/>
<copy toDir="${project.output.dir}${file.separator}new">
<fileset dir="${project.dir}${file.separator}"
includes="**/*.xml"/>
</copy>
</target>
</project>
```

Customizing the Text.xml Ant script

To add this target as an option in FrameMaker, you can modify the DITA initialization file to reference and define the new target, as shown in the following examples:



```
ditafm.ini - Notepad
File Edit Format View Help
AntScript=ditafm-ant.xml
Count=2
1=html
2=chm

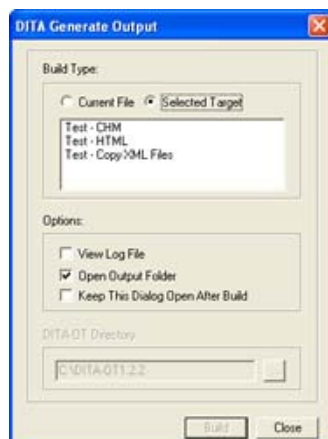
[AntBuild]
AntCommand=ant
Count=4
1=Test - CHM
2=Test - HTML
3=Test - Copy XML Files

ditafm.ini - Notepad
File Edit Format View Help
[ANT:Test - PDF]
BuildFile=C:\DITA-OT1.2.2\Test.xml
Target=pdf
OutputDir=C:\Projects\Test\out\pdf
LogFile=C:\Projects\Test\ant-buildlog-pdf.txt

[ANT:Test - Copy XML Files]
BuildFile=C:\DITA-OT1.2.2\Test.xml
Target=copyFiles
OutputDir=C:\Projects\Test\out\new
LogFile=C:\Projects\Test\ant-buildlog-copy.txt
```

Modifying the ditafm.ini file

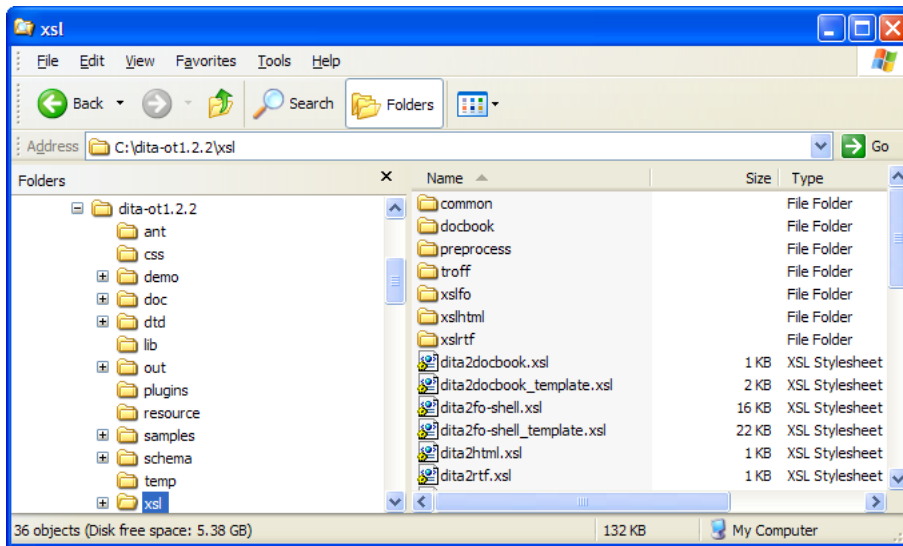
When the ditafm.ini file is updated, the new form of output is listed as a target in the DITA Generate Output dialog box.



New target available in DITA Generate Output dialog box

## Customizing DITA Open Toolkit files

In addition to customizing Ant scripts, you can customize other components of the DITA Open Toolkit publishing process. The DITA-OT contains the files you can modify. Ant processing can include files in any location, and processes and applications not included in the DITA Open Toolkit distribution.



Main files of the DITA Open Toolkit

You can modify some of the following files to customize the output generated using the Open Toolkit plug-in for FrameMaker.

- XSLT style sheets that control HTML and other forms of published output. You can change the XSLT style sheet to customize HTML design, or you can use XSLT to generate XML in a different structure from your core DITA structure to interface with specific systems.
- CSS style sheets that are used in HTML output
- Ant scripts
- Any other tools used in new targets

This customization must be separated from the core toolkit processing. For example, you can customize the XSLT that generates XHTML to accommodate a specialized structure by doing the following:

- 1 In the xsl sub folder in your DITA Open Toolkit folder, rename dita2xhtml.xsl to dita2xhtml\_original.xsl.
- 2 Create a file named dita2xhtml.xsl that imports both the renamed .xsl file, and a second .xsl file where you add your custom processing:

```
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:import href="dita2xhtml_original.xsl"/>
  <xsl:import href="/path/to/custom2xhtml.xsl"/>
</xsl:stylesheet>
```

- 3 Create the XSLT for custom processing, and save it in a folder outside the DITA Open Toolkit folder. By doing this, you can manage Open Toolkit updates, without losing any customization. You must install a new version of the Open Toolkit and then do steps 1 and 2 above.

## Appendix 1: Installing the DITA Open Toolkit

The DITA Open Toolkit includes the basic document type definitions (DTDs) associated with DITA, as well as some XSLT style sheets for processing DITA into various output formats. The toolkit depends on several freely available software applications, such as the following:

- The Java Development Kit, required for core processing of published output
- Apache Ant, required to define different forms of output, including location of source and destination files
- JavaHelp, used when you are publishing your content as Help for Java applications
- SAXON, an XSLT processor used for HTML publishing
- Microsoft® HTML Help, used to generate compiled HTML Help (.chm) files
- XML-Resolver - xml-resolver-1.2.jar file required for DITA-OT1.4.2.1. (Located in the path containing openToolkit.dll)

A complete installation includes installing these applications and Open Toolkit, then setting the Environment Variables on the computer on which you've installed these so that the various applications can work together. To run the DITA Open Toolkit from FrameMaker, you will also need to install the Open Toolkit plug-in for FrameMaker. For information on downloading and installing the plug-in, see "[Appendix 2: Installing the Open Toolkit plug-in for FrameMaker](#)" on page 13.

### Installing the Java Development Kit

At the core of the DITA Open Toolkit is the Java Development Kit (JDK), which provides the base Java functionality on which most other toolkit programs depend. To install the JDK:

- 1 Go to <http://java.sun.com/javase/downloads/index.jsp>.
- 2 Click the Download J2SE SDK link to download the JDK. Check for JDK 6 Update 3.
- 3 Accept the license agreement, and click Windows Offline Installation.
- 4 Click Save to save the file to your hard disk; note where you saved the file.
- 5 Open Windows Explorer again by right-clicking the Windows Start menu and choosing Explore from the pop-up menu. Navigate to the folder where you saved the file, and double-click the executable file.
- 6 Click Run and accept the default installation options.
- 7 Open Windows Explorer again. Look in the root folder to see where the JDK was installed, for example, C:\Program Files\Java\jdk1.6.0\_10 (the precise name depends on the version of the JDK you installed).

### Installing Apache Ant

Apache Ant is a build tool used by the DITA Open Toolkit for defining different processing steps for generating published output. To install Apache Ant:

- 1 Go to <http://ant.apache.org/bindownload.cgi>.
- 2 Click the .zip archive link to download the current version of Apache Ant (for example, apache-ant-1.7.1-bin.zip [PGP] [SHA1] [MD5]).
- 3 Click Save to save the archive file to your hard disk; note where you saved the file.
- 4 Open Windows Explorer by right-clicking the Windows Start menu and choosing Explore from the pop-up menu. Navigate to the folder where you saved the archive file.
- 5 Extract the contents of the archive file to the C:\ folder.

### Installing JavaHelp

JavaHelp creates Help for use in Java-based applications. To install JavaHelp:

- 1 Go to [http://java.sun.com/products/javahelp/download\\_binary.html](http://java.sun.com/products/javahelp/download_binary.html).
- 2 Download the latest version of JavaHelp (for example, JavaHelp 2.0\_02 [Zip]), accepting the license agreements.
- 3 Click Save to save the archive file to your hard disk; note where you saved the file.
- 4 Open Windows Explorer by right-clicking the Windows Start menu and choosing Explore from the pop-up menu. Navigate to the folder where you saved the archive file.
- 5 Extract the contents of the archive file to the C:\ folder.

### Installing the SAXON XSLT Processor

The SAXON XSLT Processor is an Open Source XSLT processor built by Michael Kay, used to transform source XML conforming to the DITA architecture to HTML. To install the SAXON XSLT Processor:

- 1 Go to <http://saxon.sourceforge.net/>.
- 2 Click the download link under the heading Saxon 6.5.5.
- 3 Click one of the download options on the SourceForge.net page.
- 4 Click Save to save the archive file to your hard disk; note where you saved the file.
- 5 Open Windows Explorer by right-clicking the Windows Start menu and choosing Explore from the pop-up menu. Create a folder within the root folder called Saxon (for example, C:\Saxon).
- 6 In Windows Explorer, navigate to the folder where you saved the archive file.
- 7 Extract the contents of the archive file to the C:\Saxon folder that you created.

### Installing Microsoft HTML Help

Microsoft HTML Help is a tool for authoring online Help, which is used by the Open Toolkit to create compiled Help (.chm) files. To install Microsoft HTML Help:

- 1 Go to <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/htmlhelp/html/hwMicrosoftHTMLHelpDownloads.asp> (make sure this is a single line).
- 2 Click Download Htmlhelp.exe and click Save to save the executable file to your hard disk; note where you saved the file.
- 3 Open Windows Explorer by right-clicking on the Windows Start menu, and choosing Explore from the pop-up menu. Double-click the Htmlhelp.exe file and follow the steps in the HTML Help installation wizard to complete the installation.

### Installing the DITA Open Toolkit

The DITA Open Toolkit itself provides files that control the dependent applications, such as the JDK, Ant, and SAXON, to generate published output. To install the DITA Open Toolkit:

- 1 Go to <http://dita-ot.sourceforge.net/>.
- 2 Click Download in the left pane, and click Download DITA Open Toolkit.
- 3 Select and download the most recent DITA Open Toolkit .zip archive (for example, DITA-OT1.4.2.1\_full\_easy\_install\_bin.zip), selecting a mirror site for your download.
- 4 Click Save to save the archive file to your hard disk; note where you saved the file.
- 5 Open Windows Explorer by right-clicking the Windows Start menu and choosing Explore from the pop-up menu. Navigate to the folder where you saved the archive file.
- 6 Extract the contents of the archive file to the C:\ folder.

### Technical skill required

Note that installing the DITA Open Toolkit requires considerable expertise, and may not be appropriate for those without at least some programming background.

### Time allocation

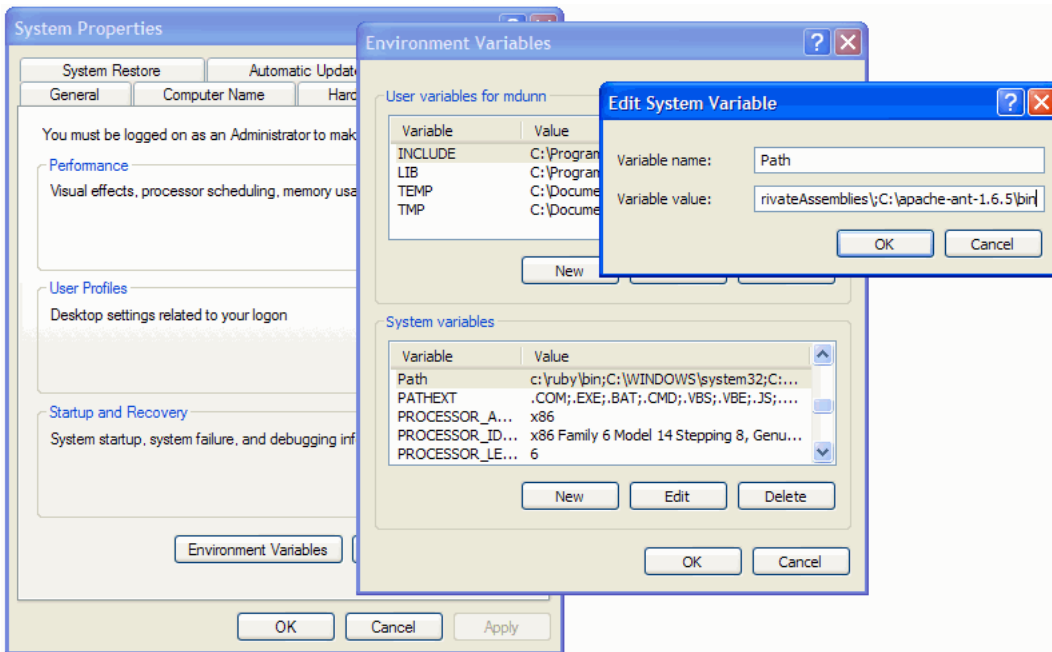
This process can be time-consuming, as you will be installing multiple Open Source applications that support the toolkit, as well as the toolkit itself. Allocate several hours for installation and testing.

## Setting environment variables

Before you can use the DITA Open Toolkit, you need to set various environment variables, so the applications can work together. To set environmental variables in Windows XP:

- 1 Click the Windows Start Menu, right-click on My Computer, and then choose Properties from the pop-up menu.
- 2 Select the Advanced tab and click Environment Variables to open the Environment Variables dialog box. For the DITA Open Toolkit, modify only the System Variables.
- 3 Edit existing System Variables by selecting them from the list and clicking Edit.

**Note:** Do not overwrite path values; instead, add additional paths to variables by delimiting paths with semicolons.



When editing System Variables, be sure to add a semicolon at the end of existing values, followed by the additional value you want to add.

- 4 Add new System Variables by clicking New instead of Edit, and setting the variable value.

Follow the instructions in the table below for each environment variable listed. Before adding new variables, check the System Variables list to see if they are already present. If so, first add a semicolon after the existing value, then add the new path.

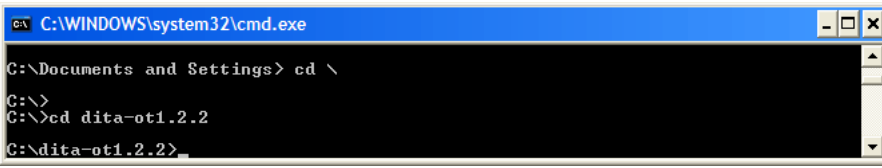
ENVIRONMENT VARIABLE	INSTRUCTION
Set the ANT_Home variable to the location where you installed Apache Ant.	Click New under System Variables and set the variable name to ANT_HOME, and the variable value to the path created by the Apache Ant installation (for example, C:\apache-ant-1.6.5).
Set the ANT_OPTS variable to allow sufficient memory to run the test Ant script.	Click New under System Variables and set the variable name to ANT_OPTS, and the variable value to -Xmx256M.
Set the Classpath variable to include the Open Toolkit dost.jar file.	If you already have a Classpath variable in your list of System Variables, select the Classpath variable in the list of System Variables and click Edit. Add a semicolon (no spaces before or after) to the end of the current Classpath value, followed by the path of your Open Toolkit dost.jar file (for example, C:\dita-ot1.4.2.1\dost.jar). If you don't have a Classpath variable, create one by clicking New and setting the variable name to Classpath, and the variable value to the path of your Open Toolkit dost.jar file (for example, C:\dita-ot1.4.2.1\dost.jar).
Set the Classpath variable to include the SAXON .jar file.	Select the Classpath variable in the list of system variables and click Edit—add a semicolon (no spaces before or after) to the end of the current Classpath value, followed by the path of your SAXON .jar file (for example, C:\Saxon\saxon.jar).
Set the JAVA_Home variable to the folder where you installed the JDK.	Click New under System Variables and set the variable name to JAVA_HOME, and the variable value to the path created by the JDK installation (for example, C:\j2sdk1.4.2_12).
Set the JHHome variable to the folder where you installed JavaHelp.	Click New under System Variables and set the variable name to JHHOME, and the variable value to the path created by the JavaHelp installation (for example, C:\jh2.0).
Set the Path variable to the folder where you installed the Apache Ant bin folder.	Select the Path variable in the list of System Variables and click Edit. Add a semicolon (no spaces) followed by the path of your Apache Ant bin folder (for example, C:\apache-ant-1.6.5\bin).
Set the Classpath variable to include the xml-resolver-1.2.jar file.	If you already have a Classpath variable in your list of System Variables, select the Classpath variable in the list of System Variables and click Edit. Add a semicolon (no spaces before or after) to the end of the current Classpath value, followed by the path of xml-resolver-1.2.jar file (for example, C:\xml-resolver-1.2.jar). If you don't have a Classpath variable, create one by clicking New and setting the variable name to Classpath, and the variable value to the path of xml-resolver-1.2.jar file (for example, C:\xml-resolver-1.2.jar).

*Overview of environment variables and instructions*

### Verifying the installation

To verify that you have installed the DITA Open Toolkit correctly, including its dependencies and configuration:

- 1 Go to the command prompt by choosing Run from the Windows Start menu, and then typing cmd and clicking OK.
- 2 Type `cd \` at the command prompt to move to the root folder.



```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings> cd \
C:>
C:>cd dita-ot1.2.2
C:\dita-ot1.2.2>
```

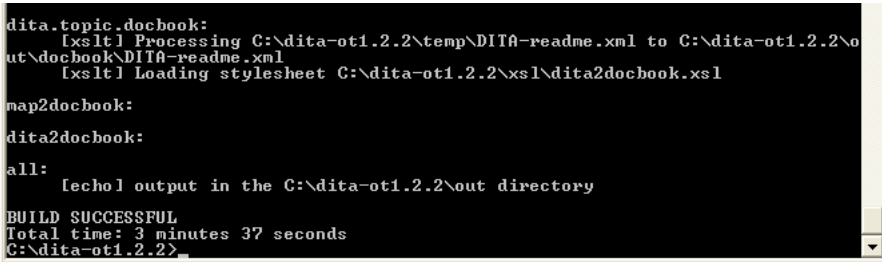
Moving to DITA Open Toolkit folder in command window

3 Navigate to the DITA Open Toolkit folder (for example, by typing `CD dicta-ot1.4.2.1`).

**Note:** This folder name must exactly match the folder where you installed the DITA Open Toolkit.

4 At the command prompt, type `ant all`

If your toolkit is properly installed, you should see text scrolling through the command window for several minutes, as the DITA Open Toolkit generates test output, completing with the text “BUILD SUCCESSFUL.”



```
dita.topic.docbook:
  [xslt] Processing C:\dita-ot1.2.2\temp\DITA-readme.xml to C:\dita-ot1.2.2\out\docbook\DITA-readme.xml
  [xslt] Loading stylesheet C:\dita-ot1.2.2\xsl\dita2docbook.xsl
map2docbook:
dita2docbook:
all:
  [echo] output in the C:\dita-ot1.2.2\out directory
BUILD SUCCESSFUL
Total time: 3 minutes 37 seconds
C:\dita-ot1.2.2>
```

Successful build reported in command window

If you receive an error, double-check the installation instructions above, or see the DITA Open Toolkit documentation.

## Appendix 2: Installing the Open Toolkit plug-in for FrameMaker

This section guides you through installing the Open Toolkit plug-in for FrameMaker and its associated components. The various sets of files you need are contained in the `openToolkit.zip` archive. Follow the directions for expanding and extracting each set of files to their proper locations.

### Installation overview

There are only two main steps to setting up the Open Toolkit plug-in for FrameMaker:

- 1 Extract all the files from the `.zip` archive to specified areas in FrameMaker installation folder (for example, `C:\Program Files\Adobe\FrameMaker9`).
- 2 Make the corresponding entry in `maker.ini`.

### Extracting and placing the archive files

- 1 Download and save the `OpenToolkit.zip` archive to a work area on your computer (for example, a `downloads` or `temporary files` folder).
- 2 Extract the contents of the `OpenToolkit.zip` file.
- 3 Copy `openToolkit.dll` into `fminit` folder.
- 4 Copy the two `xml` files: `PROJECT.xml` and `ditafm-ant.xml` into `fminit\ditafm` sub folder.
- 5 Open `maker.ini` file located in the FrameMaker installation folder. Navigate to the `APIClients` section. Add the following entry at the end of this section:

```
ditaOpenToolkit=Standard, DITA Open Toolkit, fminit\openToolkit.dll, structured
```

6 Check the final file locations. When complete, the extracted files should be in the following locations in your FrameMaker9 installation folder:

FILE TYPES	FINAL FILE LOCATIONS IN FRAMEMAKER INSTALLATION FOLDER
Core plug-ins	fminit\openToolkit.dll
Files for use with the DITA Open Toolkit	fminit\ditafm\ (Ant-related files with .xml extensions: these should be copied to your DITA-OT folder if you are using the DITA Open Toolkit)

*Table of file types and final file locations in FrameMaker installation folder*

7 Close the .zip archive if your extracting application does not close it automatically.

8 Now launch FrameMaker 9. Navigate to the DITA menu. A new submenu: Generate Output has been added.

### Moving the DITA Open Toolkit Files

There are two files that are used for working with the DITA Open Toolkit, which have to be copied into the fminit\ditafm folder in your initial installation.

1 In Windows Explorer, navigate to the ditafm folder in your FrameMaker installation folder (for example, C:\Program Files\Adobe\FrameMaker9\fminit\ditafm), and copy ditafm-ant.xml and PROJECT.xml.

2 Navigate to your DITA-OT folder (for example, C:\dita-ot1.4.2.1), and paste the two files.

### Next steps

Now that you've seen the power of publishing with the Open Toolkit plug-in for FrameMaker, you may want to learn more about the DITA Open Toolkit itself, the technologies associated with specific forms of output, such as the XSLT and CSS style sheets that generate and format HTML.

Different users will have different degrees of interest in customization. To some, it will be enough to get the DITA Open Toolkit and the Open Toolkit plug-in for FrameMaker installed, and to gain the benefit of automatic output of formats, such as compiled Help without any customization.

It should be noted that the DITA Open Toolkit is in a work-in-progress state. Some of the information in this paper may become obsolete as the toolkit evolves, so it is a good idea to check with the official DITA Open Toolkit project page for definitive, up-to-the-minute information.

### Resources

<a href="http://sourceforge.net/projects/dita-ot/">http://sourceforge.net/projects/dita-ot/</a>	The DITA Open Toolkit project site
<a href="http://www.oasis-open.org/committees/dita">www.oasis-open.org/committees/dita</a>	DITA technical committee at OASIS
<a href="http://docs.oasis-open.org/dita/">http://docs.oasis-open.org/dita/</a>	Core DITA documentation DITA
<a href="http://ant.apache.org/">http://ant.apache.org/</a>	Apache Ant project site
<a href="http://ant.apache.org/resources.html">http://ant.apache.org/resources.html</a>	Apache Ant resources
<a href="http://groups.yahoo.com/group/dita-users/">http://groups.yahoo.com/group/dita-users/</a>	DITA users Yahoo group
<a href="http://groups.yahoo.com/group/frameaker-dita">http://groups.yahoo.com/group/frameaker-dita</a>	FrameMaker DITA Yahoo group

Adobe Systems Incorporated • 345 Park Avenue, San Jose, CA 95110-2704 USA • [www.adobe.com](http://www.adobe.com)

Adobe, the Adobe logo, and FrameMaker are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. IBM is a trademark of International Business Machines Corporation in the United States, other countries, or both. Java is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are the property of their respective owners.

© 2009 Adobe Systems Incorporated. All rights reserved.