motive®

Pubs Group Reference Guide

Q4 2005

PID 000000-0000

Contents

Introduction

This document describes the process, format, and mechanics of creating documents for Motive software products. To ensure that your documents are consistent with those created by others, you should follow the guidelines in this document. However, the needs of the audience and product requirements should ultimately dictate the form of the document deliverable. For information about usage, spelling, and style, see the *Microsoft Manual of Style for Technical Publications*.

Audience

This document is intended for technical writers, editors, desktop publishing specialists, or anyone else that is creating documentation for Motive products.

Conventions

This document uses the following typographical conventions:

- **Bold** typeface represents graphical user interface elements, options, commands, fields, and labels.
- *Italic* typeface represents placeholders, such as function or method parameters, that must be provided by the user. Also identifies documentation titles.
- Monospace typeface represents code and command samples, as well as information that you should enter exactly as shown.
- *Monospace italic* typeface identifies parameters whose actual names or values you must provide at a screen prompt or text field.

1 Project Management

This chapter covers:

- The documentation plan
- Project roles
- Creating documents
- Reviewing documents
- Archiving documents

The documentation plan

The documentation plan defines the documentation deliverables and strategy for a particular project. Requirements inevitably change throughout a project, and the lead writer should update the doc plan so that it is always accurate. At the end of the project, the doc plan will accurately reflect all the documentation changes for the project.

After the release contents are established, the documentation lead creates the doc plan. The doc plan includes:

- **Deliverables**. A list of documents that will be updated for the release.
- **Non-deliverables**. A list of documents that will not be updated for the release.
- **Deliverable specifics**. A summary of each document deliverable as well as a detailed list of changed sections or topics. This section will be updated throughout the project.
- **Other resources**. Links to project resources, such as the project page, schedule, and developer documents.
- **Change history**. A list of changes to the documentation plan, with a date and description for each change.

For a sample documentation plan, see:
http://motdot/Development/Writers/DocPlanTemplate/SampleDocPlan.html.

Project roles

Most Enterprise documentation projects follow a set of similar procedures. This consistency helps the writing team (and other departments) quickly and easily find the documentation they need, both during the project and after GA. By setting expectations early, the writing team can avoid common questions (When will you have a draft of that document? What changed in that Help set? Who's working on that guide?).

There are three roles for a typical writing project:

- Documentation lead
- Writers
- Reviewers

In addition, the writing manager plays an important role by delegating resources based on the needs of the documentation leads.

Documentation lead

The documentation lead is responsible for assigning tasks to individual writers, scheduling work assignments, and supervising the creation and delivery of all documentation.

For each project, the documentation lead:

- Creates a documentation plan.
- Schedules milestones within the project schedule.
- Requests resources from the documentation manager.
- Communicates documentation progress to the entire project team, including the writing team.
- Verifies that writers and reviewers are aware of their responsibilities, meet their milestones, and have the knowledge and sources they need to be successful.
- Maintains test servers for use by the writing team and keeps up with relevant changes by maintaining close contact with the verification lead.
- At release time, verifies the documentation on the CD image, and archives the documentation appropriately.

Writers

The writers are responsible for creating or updating the documentation assigned to them by the project lead. They must meet the deadlines set within the project schedule and notify the documentation lead of any problems.

Reviewers

Reviewers are members of the documentation team that perform technical and grammatical reviews. They must meet the deadlines set within the project schedule and notify the documentation lead of any problems.

For more information about reviews, see “Reviewing documents” on page 4.

Creating documents

Writers create or update documentation as defined in the documentation plan. Depending on their writing level or time of entry into the project, writers may be given a specific outline or may be asked to scope out the document themselves. For existing documents, writers should check the defect tracker for defects open against the document.

Writers are responsible for writing according to the writing group’s standards. For more information, see “Print Documents” on page 7 and “Help” on page 23. They must also manage their documents using CVS. For more information, see “Version Control” on page 27.

Reviewing documents

Thorough reviews are important for excellent documentation. A consistent review process ensures quality by requiring documents to be approved at certain stages, and facilitates document development by enforcing incremental milestones.

There are three several types of documentation reviews: peer review, technical review, and final review. Every document should get every type of review before it is released.

Peer review

For peer reviews, someone from the writing team reviews your document. Peer reviews are flexible, but should ensure that the document is grammatically correct and adheres to Motive style. A peer review can also include testing the document against the product for accuracy.

Technical review

For technical reviews, members of other departments–such as Development, Test, and Marketing–review your document for technical accuracy and completeness. Ideally, the technical review includes someone from each relevant department. Typically, reviewers check the documents against the product to ensure that the information and procedures are correct.

The writer is responsible for notifying reviewers of their responsibilities and deadlines.

Before you send out a technical review draft, ensure that:

- The content is mostly complete, except for the index.
- It has been spell-checked.
- Cross-references are complete and correct.
- The table of contents is complete and accurate.
- The page layout is correct.

If you are sending a Help set out for review, include a print-friendly version of the document.

Reviewers can review the document on paper or online using Acrobat 5.0. When the document is ready for review, post it to the writer’s review page at:
http://motdot/Development/Writers/reviewdocs/review.html.

Final review

During the final review, someone from the writing team reviews your document according to the review checklists. This is the last review before the document is released. This review should be performed on the final output version of the document.

Archiving documents

After the project is complete, the documentation lead ensures that all the documentation is committed to CVS and creates a library page for the project at:
http://motdot/Development/Writers/library/library.html.

The lead writer also updates the Enterprise Release Summary page at:

http://motdot/Development/Writers/library/ent_summary.html.

2 Print Documents

This chapter covers:

- Overview
- Template files
- Paragraph and character formats
- Tables
- Images
- Cross-references
- Creating PostScript and PDF files

Overview

Although the publications group does not professionally print documents very much anymore, we still create documents in a printer-friendly format. This section describes the process of creating PDFs from Adobe FrameMaker.

Template files

The writing group maintains a set of FrameMaker templates that enable a consistent appearance across documents.

You can find current templates in the forgery CVS repository at:
\pubs-internal-docs\FMTemplates.

For more information about CVS, see “Version Control” on page 27.

The following sections describe the files that make up the FrameMaker templates.

Note Before you make any changes to the template files, you should review them with the writing group.

title.fm

This file is always the first in a manual. It consists of two pages–the title page and the copyright page.

The title page contains the document title, subtitle, and version number.

The copyright page contains the copyright, relevant trademark information, credits, and part number. The wording has been approved by the legal staff and should not be changed.

You should update the part number according to the doc plan. The part number naming convention is:

[Market Segment Abbreviation][Document Title Abbreviation] - [4-digit release number]

- For regular releases (X.X), change the version number on all documents.
- For patch releases (X.X.X), change only the version number on documents that have changed, because they are the only documents that will be delivered on the product CD.

Example: ENTADM-4500

Market Segment Abbreviations

Abbreviation	*Segment*
ENT	Enterprise Services
ESD	Employee Support Division
DIT	Departmental IT
EBU	E-business
EIT	Enterprise IT
ETS	Enterprise Technology Services
FIN	Financial Services

Market Segment Abbreviations

Abbreviation	*Segment*
PCO	PC/OEMs
SWE	Software for the Enterprise Services group
SWH	Software for the Home Services group
TSP	Technical Service Providers

Most documents use the Enterprise Services market segment abbreviations. Use the specialized market segments only when a book only supports a single market segment.

Document Title Abbreviations

Abbreviation	*Document type*
ADM	Administration Guide
REF	Reference Guide
CFG	Configuration Guide
INS	Installation Guide
REL	Release Notes

If you are unsure about a part number for your document, contact the documentation lead.

pubs-template-bookTOC.fm

This file contains the formats for the Contents page. The Contents file is generated automatically by FrameMaker and will eventually take on the name of the book file.

preface.fm

This file contains the formats for the preface. The preface provides readers with an overview of the document. Information in this section should give readers an understanding of why they should read the document.

The preface includes:

- Introduction to the contents of the document.
- Audience for whom the book is intended and the knowledge level expected.
- Formatting conventions.
- Motive Customer Support contact information.

chapter.fm

This file contains the formats for a typical chapter. The contents and organization of a chapter will vary depending on the document and the subject matter.

The first page of each chapter should contain a list of cross-references to all level 1 headings in the chapter.

appendix.fm

This file contains the formats for a typical appendix. Appendixes contain supplemental information that is related to but not integral to the main contents of the document. Each appendix should contain only one type of information.

The first page of each appendix should contain a list of cross-references to all level 1 headings in the appendix.

glossary.fm

This file contains the formats as well as the terms and definitions for the Enterprise glossary. The glossary should be between the last chapter or appendix and the index. You can remove terms that do not apply to your product.

pubs-template-bookIX.fm

This file contains the formats for the index. The index file is automatically generated by FrameMaker and will eventually take on the name of your book file.

For information about indexing concepts and conventions, see the *Microsoft Manual of Style for Technical Publications*.

Paragraph and character formats

You can use FrameMaker paragraph and character formats to give your documents a consistent appearance. Use the guidelines in this chapter to format your manual in accordance with Motive conventions.

The formats in the Paragraph Catalog affect whole paragraphs, while the formats in the Character Designer affect any amount of selected text—typically a single word or a portion of a paragraph.

The names of all formats in the Paragraph, Character, and Table catalogs consist of one word (no spaces) with camel-case notation. The name generally denotes the function of the format. All formats related to the same function begin with the same initial name. For example, all paragraph and character formats related to bulleted lists begin with the word Bullet.

Paragraph formats

Each paragraph in this section describes a paragraph tag in the chapter template. The paragraph tags are divided into the following sections:

- Lists
- Headings
- Text
- Code samples

Lists

The FrameMaker template includes formats for numbered and bulleted lists. Autonumbered paragraphs are numbered consecutively in a text flow.

Bulleted Lists

Use the following formats for bulleted lists:

- This paragraph is *Bullet*. Use it for bulleted lists that start under a body paragraph.
- This paragraph is also *Bullet*.

  This paragraph is *BodyIndent*. Use it for second paragraphs under bulleted or numbered paragraphs.

  - This paragraph is *BulletIndent*. Use it for secondary bulleted lists.
  - This paragraph is also *BulletIndent*.

Numbered Lists

Use the following formats for numbered lists:

1. This paragraph is *Numbered1*. Use this paragraph to begin a numbered list.
2. This paragraph is *Numbered*. Use this paragraph to continue a numbered list.

   This paragraph is *BodyIndent*. Use it for second paragraphs under bulleted or numbered paragraphs.
3. This paragraph is also *Numbered.*
   a. This paragraph is *NumberedIndent1*. Use it for alphabetically-numbered lists under a numbered paragraph.
   b. This paragraph is *NumberedIndent*. Use it to continue alphabetically-numbered lists.
4. This paragraph is also *Numbered.*

Headings

The paragraph catalog includes formats that you can use for headings.

Chapter titles

The *ChapterTitle* format is used for chapter titles only. The text frame for this format allows a maximum of three lines.

The title page of each chapter should include a bulleted list of the main sections that the chapter will cover. These bulleted items should be cross-references to the chapter's Head1 tags.

The first main paragraph heading should start at the top of the left page as described in the following section.

First- to fourth-level headings

The paragraph catalog includes formats Head1 through Head4. Each format is used for various levels of chapter and section headings within the book. You should use *Head!First* for the first heading after the chapter title page.

Procedure headings

The template contains a *HeadProcedure* format that you should use before procedures. Use an infinitive phrase for procedure headings.

Text

The following section contains examples of paragraph formats you can use for body text.

This paragraph is Body.

> This paragraph is BodyIndent.

This paragraph is also Body.

This paragraph is *DTerm1 and is used for terms*

> This paragraph is DDef1 and is used for definitions.

This is an example of a note:

Note This paragraph is NoteText.

To create a note, insert a table using the *Note1*, *Note2*, or *Note3* format.

Code samples

The following section contains examples of paragraph formats you can use for code samples. The template includes Syntax formats with 8 levels of indention. The formats that include a plus sign, such as *Syntax2+*, have less space above the paragraph.

Use the *SyntaxBlock* table format if you want a border around the code sample.

```
// This paragraph is Syntax0
// This paragraph is also Syntax0
// The following pseudo-code demonstrates second- and third-level indentation of syntax
    this paragraph is Syntax1
    then mySyntax = indented
    else
        this paragraph is Syntax2
        then mySyntax = indented
```

Character Formats

Use character formats to apply formats to any amount of selected text–usually a word or group of words.

Note Do not use the toolbar buttons to apply bold, italics, or other character formatting. Use the formats in the Character Catalog instead.

The template contains the following character formats:

Format	*Example*
Bold	Click **OK**.
Monospace	mySyntax = indented
Italic	*SyntaxVariable*

Use the *Microsoft Manual of Style for Technical Publications* to determine which character format you should use to highlight a piece of text.

Tables

The following section contains examples of paragraph formats and table formats that you can use for various kinds of tables.

Tables can break across pages, but ensure that the table heading appear on subsequent pages. If the header rows do not appear on subsequent pages, select **Add Row To Heading** from the **Add Rows or Columns** dialog box.

For note tables, see “Text” on page 12. For code samples, see “Code samples” on page 13.

The template also includes several paragraph formats you can use inside tables, such as *TableNumber1* and *TableBullet*.

Basic

Use the following paragraph formats and table format for a basic table.

Table title

CellHeading	*CellHeading*	*CellHeading*
CellBody	CellBody	CellBody
CellBody	CellBody	CellBody
CellBody	CellBody	CellBody

Grid

Use the following paragraph formats and table format for a grid table.

Table title

CellHeading	*CellHeading*	*CellHeading*
CellBody	CellBody	CellBody
CellBody	CellBody	CellBody
CellBody	CellBody	CellBody

Images

You can use images in your document to clarify a concept or explain screen elements. An image is any illustration, whether titled or not.

Overview

Use the GIF format for screen captures, and the EPS format for vector artwork, such as line drawings.

Import screen captures at 122 dpi. Perform any necessary alterations, cropping, and cleanup before importing the file. If the image is too large at 122 dpi, consider cropping it. If you are unable to crop the image, import it into FrameMaker at a higher dpi. This will reduce the physical dimensions of the image.

Always use the Imported Graphic Scaling dialog box to resize an image. Do not use the sizing handles to change the size of an image.

For text in images, use 10 pt. Arial. For emphasis, you can use 10 pt. Arial bold. Consider using a graphics application, such as Paint Shop Pro, to add callouts to the image.

Anchored frames

Use the following paragraph formats to align the anchored frames.

This paragraph is Caption

Adding images

There are three steps to adding an image: creating an anchored frame, importing the image, and aligning it.

Step 1: Create an anchored frame

You should use anchored frames to hold images in your document.

▶▶ To create an anchored frame

1. Start a new paragraph and apply the appropriate paragraph format, such as *Figure1*.
2. From the **Special** menu, click **Anchored Frame**.
3. In the **Anchored Frame** dialog box, enter the following:
   - For **Anchoring Position**, select **Below Currrent Line**.
   - For **Alignment**, select **Left** or **Right**.
   - For **Width**, enter an appropriate value, such as 4.75.

     You can adjust the width later if you need to.
4. Click **New Frame**.

Step 2: Import the image

You should import images by reference rather than by copying files directly into documents. Importing by reference keeps graphics linked to the source file, and if the image is edited, FrameMaker will update the document with the latest version.

▶▶ To import the image

1. Select the anchored frame.
2. From the **File** menu, click **Import**, and then click **File**.
3. Browse to the location of the graphic source file and select it.
4. Select **Import By Reference**.
5. Click **Import**.
6. Select **Custom dpi:** and enter 122.
7. Click **Set**.

Step 3: Align the image

You should align the image with the top and left sides of the anchored frame.

▶▶ To align the image

1. Select the image (not the anchored frame).
2. From the **Graphics** menu, click **Align**.
3. Select **Tops** and **Left Sides**.
4. Click **Align**.

   This aligns the graphic to the top and left of the anchored frame.
5. Add or remove space from the bottom of the anchored frame as needed.

Note You can make finer adjustments to the size of the anchored frame if you clear the Snap option. From the **Graphics** menu, clear the **Snap** option.

Cross-references

Using cross-references prevents inconsistency in the text, because FrameMaker updates the cross-references (numbers and text) when the book file is updated.

Cross-reference formats

The template includes several cross-reference formats.

Format	*Description*	*Example*
Chapter Contents	Use this format on the chapter title page to list the first-level headings in the chapter.	Cross-references
Chapter Title & Page	Use this format to reference a chapter.	the <BookTitle>Cross-references chapter on page 17
Heading & Page	Use this format to reference a heading.	“Cross-references” on page 17

You can add cross-reference formats if necessary.

Creating cross-references

Insert a cross-reference in your text any time you make a reference to another part of the document.

▶▶ To create a cross-reference

1. From the **Special** menu, click **Cross Reference**.
2. Do one of the following:
   - To insert a reference to a section in the same file, select **Current** from the **Document** list box.
   - To insert a reference to another file, first open the file, then select it from the **Document** list box.
3. From the **Source Type** menu, select **Paragraphs**.
4. Select the paragraph format for the paragraph that you want to reference. The **Paragraphs** column shows the text of all paragraphs with that format.
5. From the **Paragraphs** column, select the target paragraph.
6. From the **Format** list box, select the appropriate format.

   For information about the formats, see “Cross-reference formats” on page 17.
7. Click **Insert**.
8. Verify that the cross-reference is correct.

Creating PostScript and PDF files

When you are ready to generate the PDF output for a document, first create a PostScript file, then use Acrobat Distiller to convert the PostScript file to PDF.

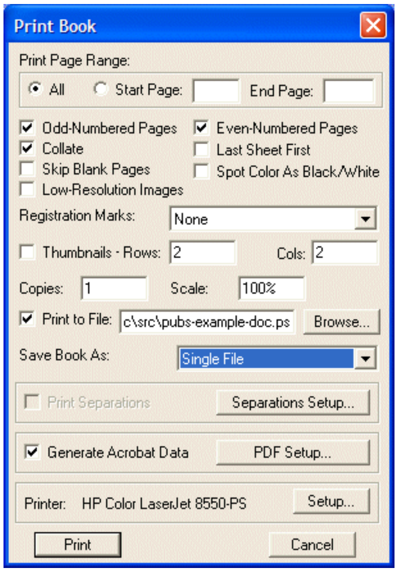
If you are creating the final version of a document for a release, ensure that page numbering, cross-references, text insets, and generated files are up to date.

Creating a PostScript file

Create a PostScript file by printing to a file from FrameMaker.

▶▶ To create a PostScript file

1. Open the FrameMaker book file or document.
2. From the **File** menu, click **Print Book** or **Print Document**.
3. For **Printer**, ensure that an appropriate PostScript printer is selected. If not, click the **Setup** button and select one.
4. Enter the following:
   - For **Print Range**, click **All**.
   - Select **Print Only to File** and specify a file name if necessary.
   - Select **Generate Acrobat Data**.



Print Book

Print Page Range:
● All ○ Start Page: End Page:

☑ Odd-Numbered Pages ☑ Even-Numbered Pages
☑ Collate ☐ Last Sheet First
☐ Skip Blank Pages ☐ Spot Color As Black/White
☐ Low-Resolution Images

Registration Marks: None

☐ Thumbnails - Rows: 2 Cols: 2

Copies: 1 Scale: 100%

☑ Print to File: c\src\pubs-example-doc.ps

Save Book As: Single File

☐ Print Separations

☑ Generate Acrobat Data

Printer: HP Color LaserJet 8550-PS

5. Set up Acrobat options. For more information, see “Setting up Acrobat data” on page 20.
6. Click **Print**.

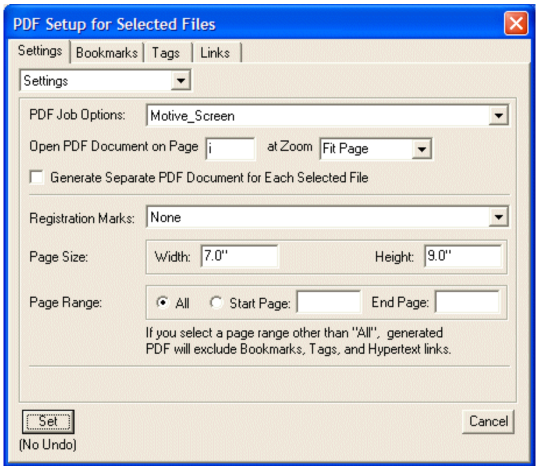
   FrameMaker creates a PostScript file in the specified directory.

Setting up Acrobat data

FrameMaker can include Acrobat data, such as bookmarks, in the PostScript file.

To set up Acrobat data

1. In the **Print Book** or **Print Document** dialog box, click the **Acrobat Data** button.
2. Click the **Settings** tab and enter the following options:



PDF Setup for Selected Files

Settings | Bookmarks | Tags | Links

Settings

PDF Job Options: Motive_Screen

Open PDF Document on Page: i at Zoom: Fit Page

□ Generate Separate PDF Document for Each Selected File

Registration Marks: None

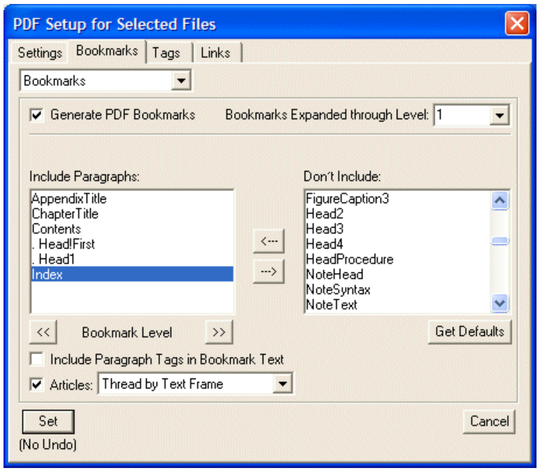
Page Size: Width: 7.0" Height: 9.0"

Page Range: ● All ○ Start Page: End Page:

If you select a page range other than "All", generated PDF will exclude Bookmarks, Tags, and Hypertext links.

Set (No Undo) Cancel

3. Click the **Bookmarks** tab and enter the following options:



PDF Setup for Selected Files

Settings | Bookmarks | Tags | Links

Bookmarks

☑ Generate PDF Bookmarks Bookmarks Expanded through Level: 1

Include Paragraphs:

AppendixTitle
ChapterTitle
Contents
. Head!First
. Head1
Index

Don't Include:

FigureCaption3
Head2
Head3
Head4
HeadProcedure
NoteHead
NoteSyntax
NoteText

<--- ---> << Bookmark Level >> Get Defaults

□ Include Paragraph Tags in Bookmark Text

☑ Articles: Thread by Text Frame

Set (No Undo) Cancel

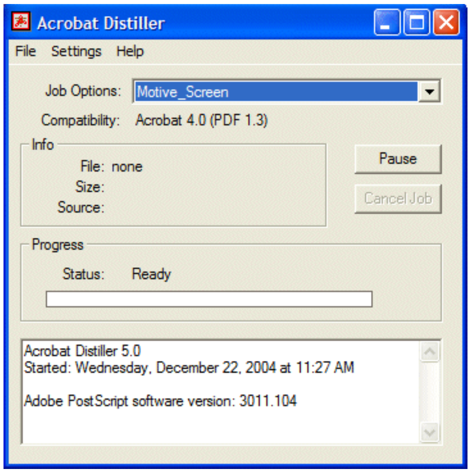
4. Click **Set**.

Creating a PDF file

After you have created the PostScript file, use Acrobat Distiller to create a PDF.

To generate a PDF file

1. Start Acrobat Distiller.
2. For **Job Options**, select **Motive_Screen**.



Job Options: Motive_Screen

Compatibility: Acrobat 4.0 (PDF 1.3)

File: none
Size:
Source:

Status: Ready

```
Acrobat Distiller 5.0
Started: Wednesday, December 22, 2004 at 11:27 AM

Adobe PostScript software version: 3011.104
```

   If **Motive_Screen** does not appear in the Job Options list, see “Setting up Acrobat Distiller” on page 21.

3. From the **File** menu, click **Open**.
4. Browse to the appropriate PostScript file and click **Open**.
5. In the **Specify PDF File Name** dialog box, specify a directory and name for the PDF file and click **Save**.

Setting up Acrobat Distiller

Set up Acrobat Distiller so that the Motive job options appear in the Job Options list.

To add Motive job options to Distiller

1. From **\\earth\dev\Pubs\DistillerJobOptions**, copy the **.joboptions** files.
2. On the local computer, paste these files into the **Program Files\Adobe\Acrobat 5.0\Distillr\Settings** directory.

When you open Acrobat Distiller, these job options should appear in the **Job Options** list.

Setting up logical page numbering

In our documents, the front matter is numbered with Roman numerals and the body matter is numbered with Arabic numerals.

This can cause confusion with some users because there can be a discrepancy between the page number on the document page and the page number in Acrobat. You can solve this problem by setting up logical page numbering.

▸▸ To set up logical page numbering

1. Open the PDF document in Adobe Acrobat.
2. From the **Document menu**, click **Number Pages**.
3. Select **From** and enter **1** to the number of front matter pages you have. For example, if chapter 1 starts on page 11, enter **1** to **10**.
4. Select **Begin new section** and select **i, ii, iii,** for **Style**.
5. Click **OK**.

3

Help

This chapter covers:

- Help systems
- Help topics
- Context-sensitive help

Help systems

Motive provides help systems for all Motive applications. Most applications use context-sensitive help topics to deliver information to users.

Help tools

Motive uses RoboHelp X3 to organize and generate help systems. This includes creating the table of contents and index and generating the help system. RoboHelp is not used to author individual help topics because it inserts proprietary tags within the topics. Instead, Motive uses Dreamweaver (or another “clean” text editor to author and edit topics.

Motive uses HTML code for topic content and Cascading Style Sheets (CSS) standards for formatting font, color, and spacing attributes. CSS files are linked to HTML topic pages with JavaScript browser sniffer tags that link specific style sheets according to a current user’s operating platform and browser application.

Help creation

Motive delivers help in Web Help format for web applications and in CHM format for desktop applications. (These formats may vary for older releases with the 5.x or 6.x Motive framework.)

When generating help, the name of the file that launches help should be start.html or start.chm. When generating a Web Help formatted help set, in RoboHelp’s WebHelp Options dialog box:

- Set the Select Skin field to (Traditional style- - no skin).
- Set the Preferred Format field to DHTML > Java Applet > Pure HTML.

Note Never set your help target to the directory where you checked out the help files from CVS. Generating the help system deletes all files in the target directory and will remove the **cvs** directory, which is needed to check files back into CVS.

Help topics

Motive writes help topics using HTML with a very simple set of coding tags. All help topics should be written using Dreamweaver or another clean text editor.

HTML editors

In RoboHelp, when you assign a default editor and double-click topic files, RoboHelp automatically opens HTML files in the chosen application. Motive recommends the use of Dreamweaver.

▶▶ To set Dreamweaver as the RoboHelp default HTML editor

1. From the **Tools** menu, click **Options** and click the **HMTL Editors** tab.
2. If the list of available editors includes **Dreamweaver** click the title and click **Set as Default**.

   If the list does not contain **Dreamweaver**, click **Add** and follow the onscreen prompts to enter the name, and location of **dreamweaver.exe**.
3. Select **Use Default Editor** and click **OK**.

Topic contents

Help topics are separated into concepts and procedures. Conceptual topics provide detailed information about a particular subject and links to related procedures. Procedural topics provide only a task introduction (such as “To create a model”) and a list of step-by-step tasks to complete the procedures. At the end of a procedure, a Related Topics section includes links to conceptual or procedural topics associated with the current topic.

Fonts in topics

By default, topics should use the paragraph tag (<p>) and heading tags (such as <h1>) to apply styles to topic text. This enables the style sheets control all topic formatting. The courier typeface should be used to represent example map code and output, such as XML output.

Graphics in topics

Graphics are used in help topics only to identify portions of an application or identify function buttons that use graphic (not verbal) identifiers.

Context-sensitive help

Context-sensitive help topics provide users with information relevant to their location within an application. To access these topics, users click a Help button or link and the application opens the associated topic.

The process for implementing context-sensitive topics varies based on release framework.

▶▶ To implement context-sensitive help for modeling releases

1. Modify the properties file to include help topic names that correspond to each context-sensitive location in the application. The entries in the file are formatted as *applicationLink=pathToHelpTopic*

   For example, DeleteModel=pubs-modeler-help/export/start.chm::_DeleteModel.html links the _DeleteModel.html to the DeleteModel location in the application.

2. Check the properties file and all help files into CVS and test context-sensitivity on the next available build.

▶▶ To implement context-sensitive help for 5.x/6.x releases

1. Review the properties file (located in the ***component/*****export** directory) to determine the ID for each screen and link this screen ID to your HTML topic.

   Ensure that you name the context-sensitive HTML files in a manner that allows you to easily identify which files contain only links to other topics. For example, in the Enterprise Console Help, it made sense to name the HTML topics using the screen ID from the properties file. In the Insight Help, the HTML topics were named using the prefix “Topics.”

   The exception to this rule is if you’re linking to a single procedure topic that’s already established. In that case, link to the file directly rather than creating a “link” file that only links to one topic.

2. Copy the **bsscnbar.js** file (in the **\\earth\dev\pubs\Help-ShowNav** directory) to the **src** directory.
3. At the top of each topic in the Help set (except the popup topics, which start with an underscore), paste the show navigation javascript (**\\earth\dev\pubs\Help-ShowNav\ShowNavJavascript.txt**).
4. Add the Main Topic List topic (named **topiclist.html**) to the RoboHelp project.
5. In **File**, **Project Settings**, select the **topiclist.html** file as the **Default Topic**.
6. Add all new files to the RoboHelp project.

   Note Do not index any topics that are context-sensitive specific.

7. Generate WebHelp without a TOC navigation pane.

4 Version Control

This chapter covers:

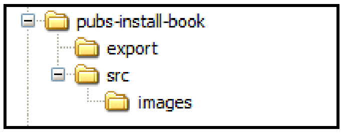
- The project directory
- About CVS
- CVS repositories and settings
- Installing WinCVS
- Configuring WinCVS
- Using WinCVS
- Naming local root directories

The project directory

The project directory contains all the files needed for the project, organized in a predictable way. The project directory typically includes a directory for the source files and a directory for the output files.

The root directory for a project is called a module. Typically, the module name should include the prefix “pubs-” and include a description of the document, such as **pubs-install-book**.

A typical project looks like this:



```
pubs-install-book
    export
    src
        images
```

You can create any other directories as you see fit. These instructions are intended as general guidelines.

About CVS

Motive Development uses CVS (Concurrent Versions System) to manage source files, including documentation source files. Using CVS, you can track changes to files, share files, and access past versions of a file.

CVS uses a client-server architecture–the current version of a file is stored in a central repository and clients connect to the repository to check out a copy of the file. You work on the local copy and then check your changes back in to the repository. For more information, see “Using WinCVS” on page 30.

CVS repositories and settings

Motive Development maintains three CVS repositories:

- forgery
- olympus
- verne

forgery

The forgery repository is used for traditional enterprise applications.

CVSROOT= **:pserver:<username>@cvs.motive.com:2401:/cvsroot**

Web access: **http://forgery/cgi-bin/cvsweb/Forgery/**

olympus

The olympus repository is used for traditional consumer applications.

CVSROOT= **:pserver:<username>@olympus.inhouse.broadjump.com:/export/CVS/src**

Web access: **http://olympus/cvsweb/**

verne

The verne repository is used for modeling applications.

CVSROOT= **:pserver:<username>@cvs2.motive.com:2402/fountain**

Web access: **http://verne/cgi-bin/cvsweb/Fountainhead/**

Installing WinCVS

Before you install WinCVS, ensure that you have a CVS account. For more information, talk to your manager.

▶▶ To install WinCVS

1. Go to http://www.wincvs.org/.
2. Download version 1.3 or later.
3. Install it with the default values for the full installation. Install cvsnt if you are prompted to do so.

Configuring WinCVS

After you install WinCVS, configure it for the appropriate repositories. This procedure describes version 1.3.

▶▶ To configure WinCVS

1. Start WinCVS. You may be prompted to install Python. Install it if you want to use CVS macros.
2. From the **Admin** menu, click **Login**.
3. For **CVSROOT**, enter the appropriate CVSROOT and click **OK**. For more information, see “CVS repositories and settings” on page 28.
4. In the **Password Authentication** dialog box, type your password and click **OK**.

   If you logged on successfully, CVS will display a message similar to this:

   ```
   Logging in to :pserver:<username>@cvs:2401:/cvsroot
   ***** CVS exited normally with code 0 *****
   ```
5. From the **Admin** menu, click **Preferences**.
6. Click the **CVS** tab.
7. For **HOME**, specify your CVS HOME directory, such as C:\CVS_HOME.

   The HOME directory stores basic account information. Regardless of how many repositories you work with, you have only one HOME directory.
8. Click **OK**.

Using WinCVS

After you have WinCVS installed and configured, you can use it to manage files. CVS stores files in a centralized repository. Typically, you use the CVS client to get the latest version of a file (update or check out), edit it on your local drive, then commit your changes back to the repository.

Checking out modules

You check out a module when you want to get a local copy of existing project files.

To check out a module

1. From the **Remote** menu, click **Checkout module**.
2. For **Module name and path on the server**, enter the name of the module.

   If you want to browse modules, go to http://cvs.motive.com/cgi-bin/cvsweb.
3. For **Local folder to checkout to**, enter the directory you want to use to store the module locally. For more information, see “Naming local root directories” on page 35.
4. Ensure that the appropriate CVSROOT is specified. For more information, see “CVS repositories and settings” on page 28.
5. If you want to check out to a branch or a tag, click the **Update options** tab and enter the appropriate branch or tag name. If you are unsure about which values to use, talk to your manager.
6. Click **OK**.

Updating files or directories

When you update a file or directory, WinCVS gets the latest version from the repository. You should always update before you start working on a file. You can update only files that you have already checked out.

To update a file or directory

1. In WinCVS, select the item you want to update.
2. Click the Update selected button.

Editing files

When you check out project files, they are typically marked as read-only. You should use WinCVS to mark them as editable before you start working on them.

To edit files

1. In WinCVS, browse to the module that contains the files you want to edit.
2. Select a file or module.
3. Click the Edit button ✎.
4. Press F2 to open the local directory in Windows Explorer.

Committing files

You commit a file when you are ready to post your changes to the repository.

To commit a file

1. In WinCVS, browse to the module that contains the files you want to commit.

   Files that have been locally modified appear with a red icon.
2. Select the file you want to commit.
3. Click the Commit selected button ↰.
4. In the **Commit settings** dialog box, enter a log message. Use the following format:

   BugIds:*23295***:** *Description*

   where *23295* is the ID of the associated defect and *Description* is a descriptive phrase about the change.

   You can put in several Bug IDs separated by commas, such as BugIds:23295,23296,23297,23302.

   If there is no defect associated with the change, enter **BugIds:0000**.
5. Click **OK**.

   WinCVS commits the file to the repository and increments the version number.

Adding files and directories

If you create a file or directory, you should add it to the repository.

To add files and directories

1. In WinCVS, browse to the module in which you want to add a file or directory. If you have not checked out the module, see “Checking out modules” on page 30. If you want to add a module, see “Creating modules” on page 32.
2. Press F2 to open the local directory in Windows Explorer.
3. Create or copy the file or directory you want to add.
4. In WinCVS, the items appear as unknown.
5. Select the item you want to add and do one of the following:
    - If the item is a text-based file, click the Add selected button .
    - If the item is a binary file, click the Add selected binary button .
    - If the item is a directory, click the Add selected button .

    If you are unsure about the file type, talk to your manager.
6. Commit the new file or directory. See “Committing files” on page 31.

Creating modules

A module is a directory at the root level of the repository. If you are creating a new project, you will create a component.

You can create a module from WinCVS or from the command line.

To create a module in WinCVS

1. Create a local directory, such as C:\FORGERY_TRUNK\pubs-install-book. Ensure that the directory has at least one file in it.
2. In WinCVS, browse to the new directory.
3. Select **Remote > Import Module**.

If the **Import Filter** dialog box appears, click **OK**.

4. In the **Import Settings** dialog box, enter the following information:
   - Repository path: enter a name for the new module, such as pubs-install-book.
   - Vendor tag: **motive**
   - Release tag: **seed**
   - Log message: enter a log message, such as “initial checkin”.
5. Ensure that the appropriate CVSROOT is selected.
6. Click **OK**.

Before you can work with the new module, you must check it out. See “Checking out modules” on page 30.

▶▶ To create a module from the command line

1. Create a local directory, such as C:\FORGERY_TRUNK\pubs-install-book. Ensure that the directory has at least one file in it.
2. Open a command prompt in the new directory.
3. Type **cvs import** <*module*> **motive seed**.

   The arguments **motive** and **seed** are Motive conventions. They have no real effect when importing a component that is intended for the trunk.
4. In Notepad, add a log message, then save and quit.

Note You may have to set the CVSROOT system variable first. The command is set CVSROOT=<value>.

Example: set CVSROOT=:pserver:<username>@cvs:2401:/cvsroot.

Deleting files

To delete a file, you can remove it or erase it. Removing a file removes it from the repository. Erasing a file removes it from the local disk only. To remove directories, you must get approval from the CVS administrator.

If you want to recover a file you have accidentally removed, see “Recovering files” on page 34.

▶▶ To remove a file

1. In WinCVS, select the file you want to remove.
2. Click the Remove selected button ✕.
3. Click the Commit selected button .

The file is moved to the attic in CVS and can no longer be checked out as part of the module.

▶▶ To erase a file

1. In WinCVS, select the file you want to remove.
2. Click the Erase selected button ♻.

The file is deleted from the local disk only. You can still check out or update the file. This function is useful if you want to discard local changes and reload the current version from the repository.

Recovering files

You can use the web interface to recover a file that has been removed.

▶▶ To recover a file from the attic

1. Go to http://cvs.motive.com/cgi-bin/cvsweb.
2. Browse to the file you want to recover.
3. Click the file to view its revision history.
4. Click the **[download]** link for the version you want to recover.
5. In the **File Download** dialog box, click **Save this File to Disk** and click **OK**.
6. In the **Save As** dialog box, browse to the local directory in which you want to save the file and click **Save**.
7. In WinCVS, add the file as described in “To add files and directories” on page 32.

Naming local root directories

If you are working in more than one repository, or one more than one branch, you should set up distinct local root directories to keep the files separate.

For example, if you are working with the verne repository on the trunk as well as a branch, you would create one directory for the trunk and one for the branch.

Use the following formula to name the root directories:

RepositoryName_BranchName

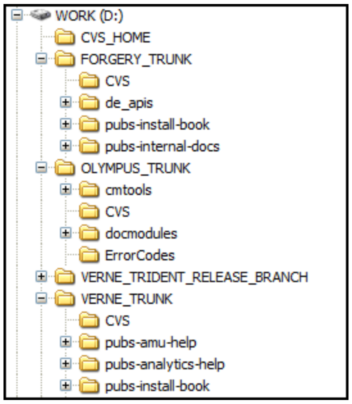
For example, for files on the trunk of verne, you would create the following directory:

\VERNE_TRUNK

For files on the Trident release branch of verne, you would use:

\VERNE_TRIDENT_RELEASE_BRANCH

Your local drive might look like this:



```
⊟ WORK (D:)
    CVS_HOME
  ⊟ FORGERY_TRUNK
      CVS
    ⊞ de_apis
    ⊞ pubs-install-book
    ⊞ pubs-internal-docs
  ⊟ OLYMPUS_TRUNK
    ⊞ cmtools
      CVS
    ⊞ docmodules
      ErrorCodes
  ⊞ VERNE_TRIDENT_RELEASE_BRANCH
  ⊟ VERNE_TRUNK
      CVS
    ⊞ pubs-amu-help
    ⊞ pubs-analytics-help
    ⊞ pubs-install-book
```

CVS adds a **CVS** directory to each directory that is added to the repository. The CVS directory stores metadata about the files and module. Do not modify the CVS directory or its contents.

Index