IDG Process Document

Last Updated: February 4, 2002

The IDG Process Document can help you to learn how our department works and how we create documentation.

The PDF version of the Process is also available.

If you need more help, you can consult the *IDG Style Guide* on Vibe.

If you find anything missing, incorrect, or outdated, please create a bug report (from http://bugs.nextpage.int) and assign it to IDG.

Conventions

Lowercase entries represent words as words (except in cases where the word must be capitalized). These entries are also always singular. For example, the entry for "above" is lowercase because it deals with using the word "above" in documentation.

On the other hand, uppercase entries deal with guidelines. These entries are also always plural. For example, the entry for Punctuation Guidelines deals with some basic rules for using punctuation in documentation.

So, the entry for makefile is lowercase and singular because it deals with using the word "makefile" in documentation. There could also be an entry for Makefiles that dealt with how to make and use makefiles to create content collections of your documentation.

As far as possible, given the above information, the appropriate style has been applied to this Guide. If you find a discrepancy, report it to the person responsible for maintaining the guide.

Document Information and Copyright Notice

Document Name	Version	Date
IDG Process	1.1	February 4, 2002

Published and printed in the USA.

NextPage, Inc.

3125 West Executive Park Way

Lehi, UT 84043

U.S.A.

Alt Text Uses

"Alt text" is the common term for the descriptive text that appears as an alternative to a graphic image on Web pages. The text is indicated in the HTML file by the attribute "ALT" in an image tag. The code used for the graphic and the alt text looks like this:

```
<IMG SRC="image.gif" ALT="Add the description of the image here">
```

Always use alt text for graphics (except for bullets or spacing graphics). Use a descriptive word or phrase so that it is easily understood for those who do not have graphic loading turned on, or for the visually impaired who use screen reading software.

See Also

"Graphics" in the *IDG Style Guide*

Builder

"Builder" is the common name for "NextPage Application Framework Build Utilities" or "NAF Build Utilities" for short. If you need to point to the documentation for NAF Build Utilities Help, just use text directing the user to the "NextPage Application Framework Build Utilities Help file" since the content collection may not be installed on the server.

See Also

Building Content Collections

"Cross References" in the *IDG Style Guide*

Building Content Collections

You must install the NAF Build Utilities to create content collections. After the installation is complete, copy the npdocs-title.xil file from H:\altdoctypes directory to your Builder directory (probably C:\Program Files\Builder 3). Register the Builder DLL with Windows by dragging the DLL (srdseu.dll in the Builder directory) to the C:\WINNT\system32\regsvr32.exe file.

Follow these steps to create a content collection:

1. Create a makefile using the Make File Starter utility, or edit an existing makefile.
2. Save the makefile in the "Online" portion of your directory structure.
3. Use the default **DTD**.
4. **Name** the content collection based on the topic name.
5. The **ID** should be 10.1048/<productname>/<version>/<topicname>. The <topic-name> can't have any spaces.
6. The **Title** should be "<Topic Name> Help".
7. The **Directory to traverse** is the <Topic Name> directory that is a child of the Online directory for your product.
8. Select **Traverse subdirectories**.
9. Select the **Include files with these extensions** option. Enter `.css,.xsl,.gif,.xml`.
10. For **Hide files with these extensions**, enter `.gif,.css,.xsl`.
11. The **Indexsheet** is `npdocs-title.xil`, in your Builder directory (or in H:\altdoc-types if you neglected to copy it over).
12. Click **Create Make File**.
13. Open the makefile in a text editor like Notepad.
14. Add the following after the prj-root entry:

```
<!ENTITY style-root 'H:\DocFiles'>

<!ENTITY srdseu-class-id "{212A3586-44C7-4eef-8EC3-F53DB5F7E671}">

<!ENTITY search-replace SYSTEM "H:\altdoctypes\searchreplace.txt">
```

15. Use the makefile template in H:\altdoctypes to copy and paste all the information from the DSE section into your makefile.
16. Do a search and replace for all the XML entries: search for *content-type="text/xml"* and replace with *content-type="text/xml" dse="search-replace-dse"*.
17. Reorganize the entries so they reflect the correct organization.
18. Insert title="<Task Title>" to get the titles to appear correctly.

    Note: I've found this easiest to do with the Search and Replace utility. The search string is `document name="*.xml"` and the replace string is `document name="%1.xml" title="%1"`.

19. Save and close the makefile.

20. Copy a "Build and Update" batch file from another project, or from H:\altdoctypes. Change the directory structure to correspond to your specific case. Save and close the file.
21. Double-click the batch file to run it and create the content collection.
22. If it is a new content collection, add it to the NXT site and check to make sure everything works. Check everything into SourceSafe.

See Also

Builder

Copyrights

Always create a file entity for the standard copyright information after the Introduction page for each main topic. The file entity should point to H:\DocFiles\Copyright.xml.

For copyright year updates, use the following guidelines:

- Do not update the year when only grammatical errors have been fixed.
- If more than 85% of the content is new at reprinting, the copyright year should only list the reprint year.
- If less that 85% of the content is new at reprinting (in other words, it is mostly the same with modifications to existing topics), the copyright year should list the previous year or years and the reprint year.

Drive Mappings

The IDG team has standardized on network drive mappings that help in building content collections and in sharing information. Also, these mappings are referred to frequently by their letters in conversation and instructions.

The drive mappings are as follows:

Drive Letter	**Network Mapping**
H	\\Documentation2\c
K	\\Documentation2\D drive
P (or any letter you wish, just match it to your SourceSafe check in and out batch files)	\\Pro-fs2\Dev\Gemini\DocVSS

Other valuable network locations:

Network Mapping	**Use**
\\Installs\installs	Setup programs for the software we use frequently.
\\Pro-fs2\Test	Latest and previous builds of developed products.
\\Fs3\Users	Lets you create a directory (the standard is to use your first initial and your last name as a folder name) for your use. Many use this directory for storing important files or for keeping system backups.
\\Cdburn\BURN	Share for the CD-R in Engineering

Entities

There are many types of entities in Epic: graphic, text, character, and file.

Text entities are useful is you type a certain word or phrase often and want to speed up your typing a bit. It is also helpful to use a text entity when the product names or functions keep changing. A text entity works by having you assign a set of characters a value. When you change the value in one place, it is changed everywhere else automatically.

Follow these steps to create a text entity:

1. Open the book file.
2. Select **Entities** and then **Text** from the menu.
3. Type in a **Name** for the entity (for example, "cnm" to take the place of "Content Network Manager").
4. Enter the text that could change or that is repetitive in the **Content** field.
5. Click **Declare**.
6. To insert the entity into the document from this window, click **Insert**. Otherwise, it is easy to use the dropdown list for text entities and select the appropriate one to add at the appropriate location.

If you use text entities, you need to update the searchreplace.txt file in H:\altdoctypes so that they get replaced on the conversion to a content collection.

If the content will also be included in the HTML Help, the search and replace script also needs to have the entity reference included or the HTML Help build will fail. Currently, there are entity references for the following words and phrases:

Word or Phrase	**Entity**
Content Network Manager	cnm
NXT 3	platform
Matrix (or BakerMAKS--change this in your makefile).	application
NXT 3	nxt
NXT 3 Server	nxtserver
Master Glossary	glossary
Imagemap	imagemap
Site Design	sitedesign

Unix	unix
a non-breaking space (NOT the phrase, but the actual non-breaking space defined as)	space
Lotus Notes	notes
Lotus Notes Service	notesservice
Lotus Notes Content Adapter	lnca
DOCS Open Service	doservice
DOCS Open Content Adapter	doca
DOCS Open	do
NextPage Application Framework Build Utilities	builder
NAF Build Utilities	buildershort
Matrix	Matrix
replacement variable	replacevar

You can insert these into your book file and then insert them in any document that you open from that book file.

Epic Editor Tips

Epic is what we use to edit and create XML files. You also need it to create a PDF.

After you install Epic, you need to send a note to Stacy with the **Server code value** (you find this in the License Administrator utility for Epic) to get a license number to use Epic. After Stacy sends the license, copy and paste the value into the License Administrator and click **Install Key**. You're ready to use Epic.

Here are a few tips on working with Epic:

- Epic always opens to the same folder (something like C:\Program Files\Epic\epicuser). Create a shortcut in this folder for the book file for each of your projects. This will make it very easy to open the project you need without having to browse through several directory levels. You can do this by right-dragging the book file from its location on H or K to the appropriate folder on your computer, then selecting **Create Shortcut(s) Here**.
- If you are worried about a name or field changing, or if you find that you consistently type the same thing over and over, try creating a text entity to do the job for you.
- Ctrl+S saves the current document, but Ctrl+W does not close it. Ctrl+W deletes the word or the entire tag directly in front of the cursor. There is not a shortcut key (apart from Alt+F4) that closes the current document.

File Naming Conventions

Because of the way that Epic, NXT, and HTML Help handle files names, we have had to adopt a set of standards for how we name our files.

Document file names (an XML document, specifically) can have spaces in them.

Graphic file names cannot contain spaces and they must start with a letter.

Capitalization of file names for documentation does not matter, however, it does matter for Unix, so be sure to match whatever capitalization system is used in the content collection makefile so that links will continue working.

The first document for a Help topic is "Introduction.htm".

Each folder in a Help topic (except the main folder, which has "Introduction.xml") has an "Overview.xml" file.

Glossary Terms

We maintain a separate content collection that has terms and definitions for most of the words unique to NextPage products. You can find the updated list of all topics in the Glossary Links document.

Provide a link to the glossary definition of a term the first time it is used in each topic; do not include more than one link to the same definition from the same topic.

The term that is linked from does not need to be identical to the term in the glossary. Use whatever form or tense is appropriate for the context of your help information.

Graphic Size Guidelines

In order to maintain consistency, and to ensure that the graphics display correctly in PDF and other formats, here are some guidelines about width and height for graphics.

The widest a graphic can be is 432 pixels (6 inches) and the longest is 648 pixels (9 inches).

Most of the graphics we do (not including icons and other inline images) are wider than they are longer, thus the guidelines below reflect this. However, the measurements can be switched so long as they do not exceed the maximum width and height.

Handmade graphics (through Visio and others)

Size	**Width x Height**	
	Pixels	**Inches**
Full Page	432 x 648	6 x 9
Half Page	432 x 324	6 x 4.5
Quarter Page	216 x 324	3 x 4.5
1/8 Page	108 x 162	1.5 x 2.25

Screen Captures

Note: We are working on standardizing the GUI window sizes so this becomes easier

Window	**Width x Height (pixels)**
Main Window	432 x 302
Wizard Window	432 x 351
Properties Window	386 x 435
Secondary Window	366 x 374
Misc. Small Window	360 x 166

Misc. Medium Window	428 x 262
Misc. Large Window	432 x 440

HTML Help Files

Content Network Manager uses HTML Help to display help topics.

The program links to the help file and sometimes to specific topics in that file. This list of links is in the Help Define document.

In order to get this HTML help file, you need to follow several steps.

1. Copy all the XML files to a neutral location (like your computer).
2. Use the Search and Replace tool to search for and replace all the text entities that are used (this works best through the use of a script file. See before.srs for an example).
3. Use the XSLT.js file (available from Hammersmith) to convert the XML to HTML.

   Note: This works best and fastest when done in a batch file. See doconvert.bat and doconvert2htm.bat for examples.

   The syntax is as follows:

   ```
   FOR %%i IN (*.xml) DO cscript "c:\<Path to File>\xslt.js"
   "%%~fi" "H:\DocFiles\xml-to-html-conversion.xsl"
   "c:%%~pni.htm"
   ```
4. Run a search and replace on the HTM files for links: the extra space after a link word, fixing links to folders and topics outside the content collection, and changing the linked file from ".xml" to ".htm". See sr.srs for an example of this script.

   Note: If you have trouble with the above link, or if the file shows in the browser instead of asking if you want to download it, try the htmlhelp.zip file, which contains all the scripts and batch files mentioned.
5. Although not completely necessary, it makes it easier to copy the resulting HTM files, along with the CSS and any GIF files to another directory.
6. Use the **HTML Help Workshop** program to create and compile the help file.

   This program can also be run from a command line (and hence a batch file):

   ```
   HHC "<path to help project file>\<help project file
   name>.hhp"
   ```

   However, this will only work on an existing help project file. To create a new file, see the next set of steps about creating a project for the first time.
7. When the process completes, copy the CHM file to the HTML Help directory on H (in the directory structure for the product you are working on).

Here is some information about creating a help project for the first time:

1. Start the **HTML Help Workshop**.
2. Select the option to create a new project.
3. Specify where you will put the file, and what to call it (Content Network Manager is cnmgr.hhp, Lotus Notes is LNProperties.hhp, and DOCSOpen is DOProperties.hhp).
4. Select that you have already created HTML files for this project.

5. You can add the files now or later. When doing this, you can select multiple files from a directory (but not files from different directories) by using the Ctrl or Shift keys.
6. After the wizard creates the project, double-click **[Options]** to edit the properties for the project.
7. Enter a title for the help file (like "Content Network Manager Help").
8. The default file is the one that comes up if the user clicks the help button in the main window or when there is no other context. This is usually the Introduction.htm file.
9. On the **Files** tab, select the **Automatically create contents file (.hhc) when compiling** check box.
10. On the **Compiler** tab, select the **Compile full-text search information** check box.
11. Click **OK**.
12. Click the **Save and Compile** button.

    Alternately, you could select **Save all Files** and then **Compile** from the **File** menu.
13. You can also use the program to view the compiled file to make sure it worked.
14. Before compiling again, double-click **[Options]** to edit the properties for the project.
15. On the **Files** tab, clear the **Automatically create contents file (.hhc) when compiling** check box.
16. Now you can edit the contents listing and move entries around to match the structure you want.

Inspection Checklists

We have two inspection checklists:

The Documentation Content Specification is a document that outlines what will be in the documentation. Basically, the content specification outlines what will be included in the documentation and provides a brief outline.

The Documentation checklist is used for reviewing documentation at the Beta 1 or later state. This is when we hand it out for technical or internal reviews.

Links and Epic

When creating a link in Epic, use the absolute path from your topic base directory. For example, "Tasks/Certain Task/Performing a Task.xml".

If you need to link to another content collection, use the CollectionLink tag. The Path attribute should look as if you are linking to the file normally, as if it were part of your topic.

For the ID attribute of any link, use the file name (minus any spaces, punctuation, or extension) of the topic you want to link to.

Makefiles

In your makefiles, link to the XSL at H:\DocFiles\npdoc.xsl instead of copying one to your directory. This will make sure you always have the latest version for a build.

The previous version of the XSL is also in H:\DocFiles, but called npdoc_old.xsl (in case you need to use it for building an older content collection).

Meetings

We have a weekly group meeting every Friday (unless something comes up and Stacy changes it). Be prepared to discuss what you are working on and what you plan on working on in the coming week.

Also, you must schedule a weekly one-on-one hour-long meeting with Stacy, to discuss whatever you need. Once per month, this one-on-one should be a lunch meeting.

NXT Installations

For installing NXT, use a serial number like the following: 30VDK3123456aa. You must have a "3" after the "VDK", followed by any six numbers and ending with any two letters.

After the installation has finished, you may want to do the following:

- Change the NXT files in the NXT 3\doc directory from read-only to writable.
- Edit the siteview.ini file so that `Cache=0`.

Organization Guidelines for Documents

Organize your topic as follows: Tasks, Concepts, Reference, Examples, Screenshots.

In lists and contents listings, organize by importance, most used, order of access, or need. Use alphabetical as a last resort. The best places for alphabetical lists are in the contents listings for Reference and for Screenshots.

The directory structure on H or K (documentation2) works as follows:

- altdoctypes: miscellaneous folders, NPDoc folder for copying to your C drive under your own altdoctypes folder, XIL file for copying to your Builder directory, other miscellaneous files.
- DocFiles: CSS and XSL file to link to from makefiles, searchreplace.txt to point to from makefile that contains all the search and replace strings for content collections, other miscellaneous files.
- <Product Code Name>: PDFs directory for creating PDFs, PostDocs directory for content collections ready to go live on the Docs server, Documentation directory.
- Documentation: Several folders and files for creating a nightly build process (not currently working), Src folder.
- Src: revision history template, readme, file size document, directory for each product topic (Access Control, Content Network Manager, Manage Content, etc.).
- <Product Topic> (child of Src): Book file for topic, Online folder, PDF folder containing the latest PDF (if you use the PDFs folder to create a PDF to, drag the completed PDF to the batch file to automatically copy it to the correct location).
- Online: Batch file or files for creating content collection, makefile, content collection file, any log files, <Product Topic> folder.
- <Product Topic> (child of Online): Introduction.xml, custom CSS and XSL files if not using ones in H:\DocFiles, revision history.htm, Concepts folder containing concept XML files (including Overview.xml), Examples folder containing example XML files (including Overview.xml), Imagemaps (or Screenshots) folder containing Screenshots XML files (including overview.xml) and graphic and icon files, Reference folder containing reference XML files (including overview.xml), Tasks folder containing Tasks subdirectories and critical tasks.xml and Overview.xml files. You may further include subdirectories (especially under Tasks) as needed, but do not go more than one level under each main level because HTML Help can't handle it.

Overviews

Every topic needs an overview (typically called Introduction.xml).

Every major division (tasks, concepts, etc.) needs an overview page (called Overview.xml). And every division within those divisions needs an overview page (called Overview.xml).

Every dialog box and window (other than the main program window), including wizard panels, needs an overview page that points to every reference topic that relates to it. This overview page is used in the HTML help and can be used in the PDF version to help organize the references section better.

Revisions

We need to track major changes to the documentation between releases, including additions, deletions, and changes. Grammar and spelling changes are not considered revisions unless they are substantial (and why are they substantial in the first place, we want to know?!).

When you make a change to a document, increment the document version in the Introduction.xml file for the topic. Use a dash and a number after the official version. For example, Version: 3.2-1 for the first set of changes and Version: 3.2-2 for the next set of changes.

Do not update the ID of the content collection however, otherwise links will break.

Use the *revision32* tag to mark where you made the change.

Update the "revision history.xml" file for the product. The entry should include the incremented document version, the date the change was made, and a bullet item for each change that briefly describes what is changed and links to the change.

See Also

"Version Numbers" in the *IDG Style Guide*

SourceSafe Uses

We use SourceSafe mainly for backup purposes.

The path for the documentation is as follows: \\Pro-fs2\Dev\<Product Code Name>\DocVSS

The path for the development side is \\Pro-fs2\Dev\<Product Code Name>\VSS

If you are asked for a user name, enter your email ID (for example, Garrett.Winn). Do not enter a password.

Status Reports

Send your status report to Stacy by the end of day on Friday each week (or 8 a.m. Monday at the latest), showing the tasks you worked on for the week and plans for the next week.

Use your One-on-One agenda as a guide in what to include in the Status Report.

Be sure to include quantifiable values for how much you accomplished, and how much more needs to be done.

Tables and Epic Editor Tips

Here is some information about creating tables in Epic (if you want them to work right in the PDF and Content Collections). These attributes are accessible by right-clicking anywhere in the table. Always specify each item using **pixels**, not inches or anything else.

Attribute	**How**
Border (line width around table)	Modify --> Table. Border Width. If you don’t specify, the default is a 1-pixel border. AND Modify --> Borders. If you want it to be the same in the PDF as online. Select None or All. If All, select a line “style” you only have 3 choices: regular, skinny, or wide (try to match the Border Width property). If you select any other Border Presets, they will work in the PDF, but not online.
Cellpadding (amount of space between the border of a cell and the contents of the cell)	Modify --> Table. Cell Margins. I only can use the value for the Left Margin. I recommend making them all the same so the PDF looks similar. If you don’t specify anything, the default is 1 pixel.
Cellspacing (width of lines between cells)	Always 2 pixels.
Width (table width)	Modify --> Table. Display Width. If “Use Default” is checked, you’re ok. Otherwise, make sure the value is in pixels. Default is the width of the screen (it actually doesn’t specify anything, that’s just what the browser uses).
Cell Alignment (alignment of cell contents)	Select one or more cells. Modify --> Cells. Horizontal Justification, Cell. Justified doesn’t work. Default is left, even if you don’t specify anything.
Cell Vertical Alignment (vertical alignment of cell contents)	Select one or more cells. Modify --> Cells. Vertical Justification. Default is top, even if you don’t specify anything.

What Does Not Work

Attribute	Why
Cell Margins	Modify --> Table. HTML only supports one value for everything. However, if you do use these, they will show up in the PDF. Left Margin is used for HTML.
Cell Dimensions	Modify --> Cells. Dimensions tab. Epic uses weird formatting, so the column widths don’t work right in HTML. They work fine in PDF.
Cell Shading (Color)	Modify --> Cells. Shading tab. Working on fixing it. Works in PDF.
Composed Output	Modify --> Cells. Composed Output tab. Don’t know what this is for.
Cell Font	Modify --> Cell Font. Working on fixing, but for now will only affect PDF.
Cell Merging or Splitting	Span/Unspan Cells. Epic does weird stuff to the source, so I don’t know how to get it to work. I suggest doing tables inside tables to get close to the same effect. However, column spanning does work okay if you really need it (row spanning is still iffy).