

FrameMaker 2017 and MS Word 2016 Feature Comparison

How to determine which product best matches different projects or workflows
for technical documentation



Adobe FrameMaker (2017 release)

For a 30-day free trial, visit www.adobe.com/go/tryframemaker

To schedule a private demo, visit www.adobetechcommdemo.com

FrameMaker and Word are not “competitors”

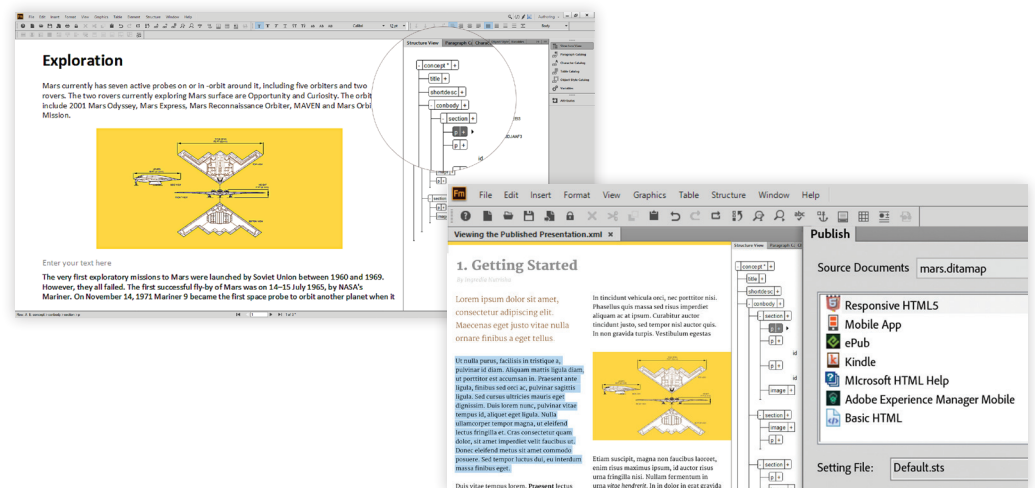
It's a common misconception that FrameMaker 'competes' with Microsoft Word. Not really. Many customers are forced to choose between the two products due to budget constraints or misinformed management.

At first glance, they seem like similar authoring solutions: both the products perform word processing and multi-column page layout. Both are capable of working with multiple files and generating Tables of Content and Indices, albeit on very different levels of strength and productivity. It's when you compare tables, equations, and options for positioning graphics that the contrast between FrameMaker and Word becomes especially evident.

FrameMaker and Word grew up with very different customer needs

Microsoft Word was initially developed as a 'personal productivity' tool: a word processor for a personal computer. For years, Microsoft's main focus for Word was the single-author creation of letters, memos, and typical short office documents. Anything longer than 10 pages was considered a 'long' document.

FrameMaker was introduced in 1986, and Microsoft Word in 1983. For the first decade and a half of its history, very few Word users utilized the software to create complex, technical documentation comprising elements like multiple tables and graphics with varied page placements, because it was all simply too difficult. Also, the early computers that initial versions of Word ran on had limited disc space and memory.



FrameMaker was born and developed on UNIX workstations, while Word was born and developed on personal computers with far less memory and disc storage. As a result, FrameMaker used multi-tasking and the ability to process thousands of pages even in its earliest versions.

FrameMaker had a distinctly different beginning and customer base from Word. The earliest computers that ran FrameMaker were powerful UNIX workstations which possessed memory and disc capacity that would not be available on a PC or MAC for nearly 10 more years. As a result, the earliest users of FrameMaker created long, complex documents which involved collaboration amongst many authors, and content reuse.

For an appreciation of just how different FrameMaker was in the early days, you should review the 2011 Content Wrangler blog [“25 Years of FrameMaker: Why This Product Still ‘Rocks After A Quarter Of A Century’”](#). It highlights a number of FrameMaker's early powers; features that would not be available on any 'competitive' authoring tools for many, many years to come. As an example, within the first year of FrameMaker's creation, the product successfully created, opened, and printed a single document that was over 1,000 pages long! Even today, one can only imagine how long it may take for Word to open a file of that size.

The earliest users of FrameMaker created high page-count documentation which underwent frequent revisions. Single-source publishing was an early innovation for FrameMaker; by 1991, it was possible to use conditional text control to create multiple versions of documentation from a single set of source files. Note that this was seven years before the birth of XML!

Unlike early versions of Word, FrameMaker was developed to handle hundreds of imported graphics, especially screen captures. Because graphics were used so extensively in software documentation, customers influenced the developers of FrameMaker to create many ways to position and control anchored frames. In stark contrast to Word, the default setting for importing a graphic in FrameMaker is to do so by reference, rather than create an embedded graphic.

The ability to 'transport' content more swiftly

Let's compare FrameMaker and Word to two types of vehicles. Imagine Word as a small, personal pickup truck, designed for light duty and built to carry enough cargo for one or two people. FrameMaker, on the other hand, would then resemble a commercial cargo vehicle, like a moving van. So the vehicle that represents FrameMaker can easily transport many times the amount of the cargo carried by Word in a single trip.

It is theoretically possible, but either painful or impractical, to accomplish many goals with the smaller vehicle, the pickup. For example, if you had a very large house with a lot of furniture, it would take several dozen trips to move all your belongings from one location to another with the pickup. However, with a larger, commercial vehicle (FrameMaker), it would take only a trip or two to move the same amount of cargo.

If you moved this amount of cargo only once in every few years, you could 'live' with the smaller vehicle (Word). However, if your needs changed, and you decided to create a small business that moved cargo seven days a week as a service, you would logically conclude that you need to move 'up' to the larger capacity vehicle (FrameMaker), which is built to move higher loads more efficiently.

Just as families or businesses sometimes outgrow a small vehicle, we sometimes outgrow the limits of Word as our document and content capacity needs increase. Ironically, returning to our analogy, no business would ever ignore vehicle wear, high fuel/mileage ratios, or lost time when pickup truck drivers are unavailable for new projects making 'unnecessary' trips.

Personal cargo capacity and
enterprise-level cargo capacity



In spite of this, a number of workgroups fail to account for the many lost hours, unavailable manpower, and other sacrifices that arise from continuing to use Word for a workload that is just too big for it to handle. This would be similar to running a furniture-moving business with a small pickup, and not including the number of trips or excessive gas mileage required into the profit-and-loss analysis. As we noted, when it comes to choosing an authoring solution, some people overlook potentially enormous 'cash leaks' in terms of human resources and project deadline overruns.

How to determine if you have 'outgrown' Microsoft Word

Although Word is a fine product and can be productive with a properly constrained workload, there is a point where continued usage can become painful if your workload has outgrown Word's capacity.

The following are some symptoms that indicate you may have outgrown Word:

- Your staff spends an excessive amount of time waiting for large files to open or print
- Modest edits to documents with mixed page or column layout require more than a few minutes to fix damaged page breaks
- Documentation requires complicated auto-numbered paragraphs which are 'fragile' and 'break' easily when using Word section breaks and other features
- Maintaining multiple product versions of documents causes publishers to resort to duplicate files, copy/paste, and other crude techniques
- Corrections to index markers take an inordinate amount of time
- What seems to be a 'simple' edit has added 17 minutes of publishing time correcting 'slippery' graphics that are moving to the wrong page or column
- An equation requires a correction, and your staff are creating a new equation from scratch, rather than attempting to edit the existing instance
- Multiple file (chapter) projects are getting out of hand using 'master documents' or other techniques to bind separate files together
- You may be required to regularly work with resource-intensive, high-resolution graphics
- Your customers may be demanding dynamic examples presented in videos and screen motion captures

Structured editing and content reuse

Single-source publishing—FrameMaker had early support for conditional text, which could show and hide selected parts of content, creating different versions of a document or manual from one set of source files. For instance, early versions of FrameMaker used conditional text control to create a UNIX and Windows version of FrameMaker documentation. Early versions of FrameMaker supported "save as HTML" as well as "save as hyperlinked PDF".

Automatic hyperlinks—Very early versions of FrameMaker automatically placed a hyperlink in generated Tables of Contents, List of Figures, List of Tables, Indices, and between cross-references (xrefs) and target text. These hyperlinks could be used for cross-document navigation during authoring, and were automatically placed within generated HTML files.

XML and Structure—About 15 years ago, a version of FrameMaker was developed to handle SGML. This evolved into the current product, which has an intuitive approach to handling DITA, XML, and other structured applications. This has become a solid growth area for FrameMaker, and over half of the user base authors create/design content in DITA or XML on a regular basis. Since topic-based authoring is a natural fit for FrameMaker, it is a popular authoring tool companion or CMS (Content Management System) in environments with high volumes of content.

Word has some of the capabilities described in this section, but to a lesser degree. This is primarily because the original publishing model was designed for single users aiming for an increase in personal, not enterprise, productivity.

Tables of contents

Selective Feature Capability Comparison

- [Page Layout](#)
- [Tables](#)
- [Book Building Features](#)
- [Catalogs, Paragraph, and Character Formatting](#)
- [Graphics, Multimedia, Anchored Frames, and Equations](#)
- [Long Document Support](#)
- [XML and Structured Authoring](#)
- [Automated Authoring and Formatting](#)
- [Multi-Channel Publishing](#)
- [Role-Based Publishing](#)

Page Layout

[Back to TOC](#)

| Feature | How FrameMaker handles this | How Word handles this |
|---|--|---|
| Master Pages for shaping/paginating text | FrameMaker supports a variety of master page styles which automatically shape pages with single or multiple text flows. This is done using either paragraph tags or using elements and can be fully automated. | Highly developed page layout tools, especially concerning text wrap around. Text may even wrap around tables. Product performance diminishes if documents achieve a high (e.g. 500) page count. |
| Rotated or landscape pages mixed with portrait pages | FrameMaker enables user to instantly change page orientation to landscape. Text may either paginate through a landscape page, or around it. This can be done in conjunction with automated page layout making it effortless. | Word users must create section breaks around the pages they wish to change to landscape. Pagination is halted; there is no pagination possible through the landscape pages. Section breaks have a history of being delicate and can cause problems with adjacent numbered headings. |
| Multiple column layout | FrameMaker has the ability to allow: <ul style="list-style-type: none"> Mixing of multiple column pages with single column pages, and pagination through or around these pages. Multiple columns that may be of equal or different widths. Text frames for columns on the master page to be placed in any layout combination (e.g. a newsletter) and maintain automatic pagination. | Word requires section breaks to segregate pages that have different numbers of columns. Paginating text columns must be of equal widths. Pages with a different number of columns are "isolated" and cannot be part of the preceding pagination flow. This requires a great deal of cut/paste to eliminate white space. |
| Page headers and footers | FrameMaker has almost limitless control over rich page headers/footers. Header or footer content may occur anywhere on the page, and may even be rotated. Up to 18 system variables can control automated content that appears in headers and footers based on hidden markers (ideal for catalogs needing a variety of 'part numbers'). System variables may pick up numbered paragraph prefix, number only, and/or text of heading paragraphs. Catalogs may display first or last instance of a part number. Headers and footers may rotate and appear in side margins or on any portion of the page. In XML-based documents the content of these variables for headers and footers can be driven based on elements as well as attributes and their values. | Word document page headers or footers are limited to a "strip" at the top or bottom of the page. There are a limited number of Quick Parts that may create some automatic display of text in this zone. There is no obvious facility for displaying an entire numbered heading (first or last instance on the page). If left/right pages are to have exceptions to their header/footer display, section breaks or other awkward tools must be employed. |
| Sideheads | Paragraphs have three types of page positioning: <ol style="list-style-type: none"> in column, sidehead (resume heads), and run-in headlines on same baseline as following paragraphs. FrameMaker is still the only product on the market that can move paragraphs into the page margin based on style tags. | Word does not support true sidehead paragraphs. This effect may only be achieved via boxed text, single-row tables, or possibly with a complex combination of reverse line spacing. |
| Run-in heads | FrameMaker can position two paragraphs on the same baseline. The purpose is to allow bold, inline text, which can introduce a topic, to be extracted into a Table of Contents or display in an automatic cross-reference. | No equivalent feature. |

Tables

[Back to TOC](#)

| Feature | How FrameMaker handles this | How Word handles this |
|--|---|---|
| Table styles | FrameMaker table catalogs store styles which may instantly change appearance of a table and can also globally update the appearance of all table instances of that style. Alternating rows or columns with varying colors/shades or rule weights will automatically re-adjust themselves when rows/columns are deleted. Repeating table titles may have a continuation header variable and a sheet count variable in table header or footer rows. | Table styles are handled the same way as paragraph and character styles. You can globally update tables when you update the style. Table structure is not as dynamic as FrameMaker's, which will automatically reapply patterns of color, tint, and rule weight as a table is edited. |
| Table title | FrameMaker tables have titles which are part of the table structure itself. Table titles can be multi-line and multi-paragraph. A special system variable will automatically drop in 'Continued' at end of a table title if a page break occurs. The special system variable can also display number of pages or sheets of paper that a table takes up, e.g. sheet 1 of 13, for long, page-breaking tables. It is possible to make a cross-reference to a table title and pick up both table number and the text. | Word tables can repeat header rows. Table titles (captions) are not part of the table and cannot automatically repeat with as much flexibility as FrameMaker. You have to anchor them to the table. No table continuation for title. No sheet count available. No ability for cross-references to display both table number and table text. |
| Table footnotes | FrameMaker has a second type of footnote, for tables, which can display as alpha, numbers, Roman numerals or symbols (e.g. asterisk and dagger) | No equivalent functionality. Such footnotes have to be inserted as endnotes instead, and will not update if their order is changed. |
| Table styles can remember paragraph styles for specific columns | In FrameMaker, when Table Styles are defined or globally updated, the table catalog remembers which paragraph style is in each column. You may define a table that will create a certain number of columns, and give you a numbered list in column 1, bullets in column 2, and different paragraphs in other columns. | Table styles can include formatting for specific rows (first, last, odd or even), columns (first or last), and cells (top right, top left, bottom right, bottom left). Other locations cannot be customized with a specific format. |
| Table styles can remember paragraph styles for header rows | When table styles are defined or globally updated, the table catalog remembers which paragraph style is in the table header and footer. You can apply a unique format to the header or footer of a table and customize it to, for example, be bold, and white text with a blue background, and this automatically applies the next time the table type is used. | Table styles include paragraph formatting for header and total (or footer) rows. |

Book Building Features

[Back to TOC](#)

| Feature | How FrameMaker handles this | How Word handles this |
|---------------------------------|--|---|
| Tables of Contents (TOC) | Automatically generated from any paragraph or XML element text. Can include entire or partial numbered paragraph prefix. Order of text, leader dots, and page numbers are completely programmable with simple building blocks on a Reference Page. No real limit to number of levels to TOC, or the degree of customization. | Modest control over creation of an automatic Table of Contents. User has very limited control over leader dots, or fancy formatting (like placing the page number first). |

Book Building Features (Contd.)

[Back to TOC](#)

| Feature | How FrameMaker handles this | How Word handles this |
|----------------------------------|---|---|
| Index | Index entries may be sorted and displayed in virtually unlimited nested indents. Workspace pod displays all index markers sorted in a variety of ways. Character tags may be used to format portions of index entry as bold, italic, etc. Index entries can be configured with mixes of "additional entries", "see also", and parent/child from one dialog. | You can plant markers to create an automatic index. Nested index entries are more limited and Word lacks FrameMaker's format controls (character tags for bold or italic words) within markers. |
| List of Tables or Figures | The same controls can be used to create separate LOT (List of Tables) or LOF (List of Figures). As with TOC, placement of elements (e.g. page numbers) is completely open. | Generated lists (e.g. list of references) require more customization and may rely on macros. |

Catalogs, Paragraph, and Character Formatting

[Back to TOC](#)

| Feature | How FrameMaker handles this | How Word handles this |
|--|--|--|
| Catalogs available | FrameMaker has multiple catalogs that segregate discrete styles to add strong structure through formatting for documents that do not use XML: Paragraph Catalog Character Style Catalog Table Style Catalog Object Style (graphics) Catalog | Paragraph and character styles display together in Word. It takes several steps to create a new style. The operator is rewarded with instant results for having the majority of paragraphs tagged in the Normal style and uses toolbar macros to create bullets and lists. |
| Smart catalogs | A keyboard shortcut will display available paragraph or character styles "in place." Users can type the first few characters of the style name to limit choices. | Word allows a few styles to be assigned to keyboard shortcuts. Not all styles are available through this facility. |
| Find and repair format overrides | FrameMaker has the ability to find and restore any paragraph or section of text on the character level that does not match the catalog definition. It is even possible to search for tables or object styles that do not match the catalog definition. This makes it possible to swiftly eliminate operator error from documents that have been over-edited. | Word has the ability to eliminate all overrides to selected text. However, character styles are not preserved. No facility for table or graphic styles. |
| Auto-numbered paragraphs | FrameMaker has virtually unlimited number strings. It is possible to create number threads for lists, for instance, that will automatically restart at '1' if a new heading precedes it. | Word can only create numbers to nine levels. It is challenging to create complex, mixed number strings like: '13.A.iii Title'. |
| Creating and managing paragraph or character styles for templates | The Designer used by FrameMaker contains multiple functions for format in one place, with minimal requirements to open/configure multiple levels within it. Updates to formats can be global, or can be local, and are done incredibly quickly. | Styles can be set up as part of a template but require careful configuration. Updates to paragraph and character formats can be global. Otherwise you override the style. Styles easily and frequently get out of control with overrides unless the writer very strictly follows best practices. |

Graphics, Multimedia, Anchored Frames, and Equations

[Back to TOC](#)

| Feature | How FrameMaker handles this | How Word handles this |
|--|---|---|
| Anchored frames | FrameMaker will import graphics as externally referenced objects as a default condition. It is possible to embed graphics in FrameMaker, but you have to ask for it. This second choice will bloat files and diminish system performance. Referenced graphics also automatically update when the source is changed ensuring the image seen is the most current image, every time. | Although it is possible to import graphics by reference in Word, most operators do not know how to do this. One of the chief problems with Word is that the majority of documents have embedded graphics that cannot easily be altered. |
| Object styles | Vector objects or anchored frames may have named styles applied. Global updates can be applied. For instance, all named ovals can rotate, or all named anchored frames can change from below current line to 'fit in paragraph' for text wrap around. | No equivalent functionality. |
| Hotspots from vector graphics to related text | Choose certain vector objects and assign a hotspot that will enable PDF output to include a hyperlink to related text. Think of this as cross-references for graphics. | No equivalent functionality. |
| Generate table of parts, views, or animations from 3D vector diagrams | A single step can help create an attractive table that displays part names, views, or animations contained within a target 3D diagram. PDF output has automatic hyperlinks between all table entries, and in PDF animation and part's shading will occur upon selection. | No equivalent functionality. |
| Integration with Photoshop and Adobe Illustrator | Photoshop or Illustrator files may be imported into FrameMaker in native format. Double-clicking on the referenced graphic will launch the appropriate Adobe software to edit the image. | No equivalent functionality. |
| Ability to embed video and rich media | Highly sophisticated ability to not only import videos and screen simulations, but to also indicate views or stop-and-play marks within the video. These become active upon output to PDF. | Ability to import videos if they are available online (YouTube, Bing, or by embed code). You cannot add stop-and-play mark, screen simulations, or views. You can alter the thumbnail with a static graphic of your own. |
| Equations | FrameMaker has a very robust equation editor that can actually "solve" equations and is extremely stable for making edits and revisions. Equations can "round trip" through XML. | The equation editor in Word is still very fragile. Often, it is easier to start the equation over from scratch than to edit one. If you have too many equations in one document, they will cause crashes. |

Long Document Support

[Back to TOC](#)

| Feature | How FrameMaker handles this | How Word handles this |
|---|--|---|
| Book building for multiple document projects | Unstructured FrameMaker has a powerful book-building feature for combining multiple files in a hierarchical manner. Selected files may be excluded from print or output during the approval process. Structured (XML) editing in FrameMaker supports DITA map, an XML structure that is similar to a book. | The master document feature in Word has some limited capabilities parallel to FrameMaker. Master doc is somewhat fragile and is known to cause Word to crash under certain circumstances. |

Long Document Support (Contd.)

[Back to TOC](#)

| Feature | How FrameMaker handles this | How Word handles this |
|--|--|--|
| Chapter exclusion | Selected chapters can be temporarily "excluded" from print or publishing while they are in a draft status. | N/A |
| Cross-references | Any paragraph or XML element may be a target of a cross-reference. Special pods display all xrefs in current or all open documents. Unresolved xrefs are highlighted. | Simple cross-references may be created to internal or external documents. Note: large projects with 100s of xrefs in Word may have serious performance issues. |
| Cross-reference format | Xref format made of simple building blocks which can compose an entire sentence automatically, e.g. 'For more information, see Table 3.1 How to Get Started located on page 245 in Chapter 3, First Things First!' | Word does not have an xref format. All of the text beyond 'Table 1' or 'page 3' must be typed in by hand, and will not update globally. |
| Fixing unresolved xrefs | Cross Reference pod will display unresolved cross-references with a red X and provide hints to missing target text. | No equivalent tool available. |
| Speed and performance | Opening and saving long documents does not reduce performance. A 1,100-page document will take a few seconds longer to open up than a 50-page document. | Word is markedly slower when opening up longer documents, and docs with lots of tables or a very high number of large graphics. |
| Performance during output to PDF or other formats | FrameMaker document page count has little effect on the time it takes to output to PDF, HTML, or other formats. | Word's performance will diminish considerably when outputting large files with lots of graphics to rich PDF. |
| Footnotes | Complex footnote notation and styles are available in both paragraphs and tables. | The footnote feature in Word is decent and fairly strong. |
| Project management | FrameMaker allows projects to be created. These are collections of folders which are linked to the current computers folder structure. | No equivalent functionality. |

XML and Structured Authoring

[Back to TOC](#)

Note that Microsoft Word is not generally designed to work with XML or to be a structured authoring tool in the same way that FrameMaker is. Most of the following functions are simply not available without extensive customizations or buying 3rd party tools.

| Feature | How FrameMaker handles this | How Word handles this |
|---|---|--|
| Role-based XML authoring | FrameMaker is now a full-featured XML and DITA editor. Authors have three views: (a) XML view, (b) Authoring (content) view, and (c) WYSIWYG (preview) view. Full editing is possible in all three views. It is also easier now to hide menus and customize workspaces. | No equivalent functionality. |
| Ability to edit DITA or XML files directly | Can use structured applications that will support native *.xml files, not just *.fm binary files with embedded structure. | Current versions of Word have the underpinnings of XML structure with the *.docx format. However, Word is not a full-featured DITA or XML editor. Expensive third-party add-on products would be required to achieve just a part of the functionality of FrameMaker. |

XML and Structured Authoring (Contd.)

[Back to TOC](#)

| Feature | How FrameMaker handles this | How Word handles this |
|---|--|---|
| Full DITA 1.2, DITA 1.3, and XSLT 2.0 support | Virtually all DITA constructs are supported. EDD allows complex nesting of elements and automatic insertion of sibling elements. | N/A |
| Configure DITA editing for beginners | Banner text will now prompt users on what type of content to insert. This substantially reduces training time. Support for banner text is upgraded for DITA 1.2 and matched in most cases in DITA 1.3 templates. | N/A |
| XML view: Code completing and error tracking | When authoring in XML view, author is prompted for legal elements. When an element is entered, the 'closing' value of the element drops in automatically. | N/A |
| FrameMaker 2017 XML Author | FrameMaker now has a less expensive, companion product for your staff who only need to author and edit XML or DITA projects. This economical product is ideal for staff members who do not need the many rich features of "Full" FrameMaker 2017. | N/A |
| Out-of-the-box support for DITA Open Toolkit | No customization necessary to work with this popular DITA standard. If customization is desired, support is fully integrated. | N/A |
| Smart paste | Users may copy unstructured data, like portions of Word documents, unstructured FrameMaker, even HTML, and "smart paste" it into the structure view of an XML document. This automatically converts the unstructured data to valid DITA/XML. | N/A |
| Conditional output | HTML5 content that is published from DITA can use attributes such as product, platform, or audience to allow readers to filter content and see exactly what they want, all from one default template. This means that, for example, if two versions of a product have similar features, a single document comparing them can be created, and users can toggle between the versions online using a smartphone, tablet, or computer. | The footnote feature in Word is decent and fairly strong. |

Automated Authoring and Formatting

[Back to TOC](#)

| Feature | How FrameMaker handles this | How Word handles this |
|---|---|---|
| Scripting | Supports ExtendScript and custom plug-ins via FDK to automate tasks associated with high-volume production. | Word has Visual Basic and a macro language that can automate some tasks. There are fewer tasks or features available to work with here, than with FrameMaker. |
| XSLT 2.0 support for transforms | XML can be transformed through custom use of XSLT. | N/A |
| Intelliprompt for elements and attributes | Authoring in XML mode will prompt user to choose correct element or attribute. | N/A |

Multi-Channel Publishing

[Back to TOC](#)

| Feature | How FrameMaker handles this | How Word handles this |
|---------------------------------|--|-----------------------|
| Native multiscreen HTML5 output | Publish to this format natively from within FrameMaker, without any costly 3rd party plug-ins. | N/A |
| Native EPUB 3 output | Publish to this format natively from within FrameMaker, without any costly 3rd party plug-ins. | N/A |
| Native WebHelp output | Publish to this format natively from within FrameMaker, without any costly 3rd party plug-ins. | N/A |
| Native CHM output | Publish to this format natively from within FrameMaker, without any costly 3rd party plug-ins. | N/A |
| Native MOBI output | Publish to this format natively from within FrameMaker, without any costly 3rd party plug-ins. | N/A |
| Native KF8 output | Publish to this format natively from within FrameMaker, without any costly 3rd party plug-ins. | N/A |

Role-Based Publishing

[Back to TOC](#)

| Feature | How FrameMaker handles this | How Word handles this |
|-------------------------------------|---|-----------------------|
| Custom workspaces | Workspaces can be swiftly customized and 'named' to match tasks for specific users and specific tasks. Training time can be substantially reduced on complex projects. | N/A |
| Hide all or part of pull-down menus | Special config files may be used to make unwanted formatting or other pull-down menus unavailable. This is ideal with workflows in which you do not wish to have the publisher apply unapproved styles. | N/A |
| Expert mode for XML and DITA | Technical users who prefer to directly manipulate and author XML markup can work in XML view, with no preview of page output. | N/A |
| Authoring view | Content creators who need to be prompted for correct XML elements and have a generic preview during editing can now use Author View instead of WYSIWYG View. | N/A |

For any queries, please write to us at techcomm@adobe.com

