Conventions Used in the JDF Specification

# Spell checking and Template files

Here are the rules for spell checking and common styles. There are two templates

* **TemplateForEdit.fm:** for spell checking during editing
* **TemplateForFinal.fm:** for final version to publish

The two templates differ only in the character styles of Attribute, Element, Enumeration, Link\_internal, Message, Process and Resource. In TemplateForEdit.fm, all of the aforementioned character styles have a language of “US English” so that the spell checker can check their spelling. As a side effect, such words may be hyphenated. In TemplateForFinal.fm, all of the aforementioned character styles have a language of “none” so that the words are not hyphenated. As a side effect, the spell checker does not check their spelling.

The character style can be changed quickly by selected all documents in the book and then importing formats (character formats only).

When spell checking is run. All “find” options except “straight quotes” are checked and no “ignore” options are checked.

To make spell checking work, the dictionary user.dct has all JDF terms (in ArtOrginals folder). There are two caveats. JDF terms should not have plurals, such as QueueEntrys. Instead the phrase should be “QueueEntry elements”. Framemaker is unable to spell check words that exceeds about 30 characters including hyphens for hyphenation points. Framemaker truncates the word in the dictionary.

# Character Styles

**Attribute:** a JDF attribute

**Bold:** makes text bold and disallows spell checking

**Bold\_italics:** makes text bold-italics and disallows spell checking

**Change\_Flag:** for blue text that specifies whether a feature is new, modified or deprecated feature.

**ConformWord:** for all conformance words, such as MUST, SHOULD and MAY. The style forces the word to display in upper case.

**DeletedText:** not used

**Element:** a JDF element

**Enumeration:** a JDF enumeration (value)

**EquationVariables:** use in one equation in Resources. Spell checking checks this text.

**InsertedText:** not used

**Italics:** makes text italics and disallows spell checking.

**Link\_External:** blue text for URLs

**Link\_internal:** specified automatically in cross references in order to make text be 50% gray.

**Message:** a JDF message

**NoSpell:** sets language to “none” to prevent spelling checking from occurring, such as for “ ?” in the Name column and various other spelling anomalies.

**Process:** a JDF process

**Resource:** a JDF resource

**Sample\_Emphasis:** used to change JDF sample text to a color of “medium green”

**smallSize:** used to change text to 9pts in a few places to squeeze long words in to small spaces.

# Paragraph Styles

**AnchorOnlySmall:** used for paragraphs that contain only anchors for tables or figures. The paragraph is 2pts so that it doesn’t take up space. These paragraph are either in the main text and anchor a table or they are in a table and anchor a figure.

**AppTitle:** for the title of each appendix

**Body:** for all paragraphs in the main text flow except for Body\_open\_paragraph’s

**Body\_open\_paragraph:** paragraph after a section head or table.

**Bullets:** bulleted paragraph that is flush left with body paragraphs

**Bullets2:** bulleted paragraph that is indented one level from body paragraphs

**Bullets3:** bulleted paragraph that is indented two levels from body paragraphs.

**Callout\_paragraph:** paragraphs in boxed text.

**CalloutTitleReference:** title paragraph in boxed text.

**CalloutTitleSummary:** title paragraph in boxed text.

**CalloutTitleThought:** title paragraph in boxed text.

**Caption\_Figure:** style of each caption for a figure; always in a table of style Graphics

**Caption\_Table:** style of each caption for a table, always in a table of style Standard Table

**CellBody:** style of each cell of a Standard Table except for the header rows.

**CellHeading:** style of each cell of a Standard Table header row.

**CellHeadingRow:** style of a cell of a table that is important (bold); typically the leftmost column.

**ChapTitle:** for the title of each chapter

**EnumDef:** for each paragraph that defines an enumeration; has a hanging indent.

**EnumDefAfterPara:** for a paragraph that follows an EnumDef, and.is not an EnumDef; adds a bit more space above the paragraph.

**FM\_Header:** for the title of each front matter chapter, such as the Preface

**FM\_Ref:** used in references to other documents, such as in the References Appendix

**Footnote:** for footnotes.

**Graphik:** not used.

**Heading2:** section heading

**Heading3:** section heading

**Heading4:** section heading

**Heading5:** section heading

**Heading6:** section heading

**Heading7:** section heading

**Heading\_ResTable:** table header only in appendix of deprecated elements.

**Heading\_Sub-head:** header without numbering

**HeadingOfAttribute:** header for tables of Attributes values; automatically starts with word “Attribute”

**HeadingOfElement:** header for Element tables; automatically starts with word “Element”

**Numbered:** numbered list.

**Numbered1:** first item of a numbered list.

**Resource\_Body:** for resource properties that precede the table; final part and not bold.

**Resource\_body:** for resource properties that precede the table; final part and in bold.

**Run-in-Head\_ResourceRef:** for resource properties that precede the table, initial part.

**Run\_in\_Header:** used as run-in header in a few places.

**Sample\_code:** for JDF examples; spell checking is turned off**.**

**TableFootnote:** table footnote

# Table Styles

The spec uses the following table styles:

**Graphics:** A style for figures which defaults to one column and two rows, the first for the drawing and the second for a line of paragraph style Caption\_Figure. The rules are turned off. The use of the table to enclose the drawing and captions ensures that the two remain on the same page.

**Standard Table:** A style for defining JDF elements, which defaults to one header row and a few body rows. The number of columns is usually either 2 or 3. The rules are dark blue and the header row has a dark blue shading. The table also has a title above the table with a style of Caption\_Table. Each such caption must include a “Section M of N” variable which conditionally displays when the table occupies more than one page.

**Tabular Data:** A style for showing tabular data in a few parts of the spec. The rules are turned off.

**Untitled Table:** A style that is now present only in the appendix for deprecated elements.

# Table Column Widths

The table column widths vary considerably, but always sum to 6.5”. The strategy for column widths is that the name column MUST be wide enough for all names to fit without a line break. Otherwise, (if they break) they are usually not searchable by Acrobat. In 6 tables a few name columns for excessively long names extend into the next (data type) column. The data types usually don’t break, but longer ones are allowed to break in order to keep the table from getting excessively long. The Description column needs enough width to keep the table from becoming too long. When attribute values would cause a large amount of white space, they are broken out into their own table. The tables with names that extend into the data type column are: LayoutPreparationParams, Media, PageList, PageData, StrippingParams and Document Properties.

# Graphics

See the file ArtOriginals/ListOfFigures.doc for a list of all figures in the spec and the corresponding original art file.

Here are the rules for creating graphics in the JDF spec

1. Create the graphic in some format, such as Visio. MS Word or Adobe Illustrator This width should be a bit less that 6.5” or 165mm, i.e. about 6.4” or 162mm. This sizing avoid scaling in the spec and the resulting reduction in size of text.
2. Save the graphic in the appropriate subfolder of "Art originals". The appropriate subfolders of Visio. MS Word or Adobe Illustrator are "Visio", "Word\_Art\_Conversion\_Source" and "AI", respectively. The "unknown" folder contains 2005 drawings in Visio that come from some unknown format.
3. Create a png or jpg file from the original – png is best for anything that is not a photo. The file has the following characteristics
   1. It has the same name as the original except for the “png” or “jpg” suffix
   2. It is saved in the folder “Framemake 1.x/Graphics”, a parallel structure to the one in “Art originals”.
   3. When saving the png files, the following settings must be used:
4. **Resolution:** set to "custom" with value of 150x150 dpi
5. **Size:** set to “source”.
   1. When saving the jpeg files, the following settings must be used:
6. **Quality:** set to 90%
7. **Resolution:** set to "custom" with value of 150x150 dpi
8. **Size:** set to "source" if size would be about 6.5” or less or scaling in Framemaker is acceptable. Otherwise, click to “screen” and back so “source” to see actual dimension in “custom”. Then set the custom width to the desired size, e.g. a bit less that “6.5” and set the height to preserve the aspect ratio.
9. In Framemaker, use the “files/import/file…” and select “import by reference”. Select a dpi of 150 dpi. This ensures that Framemaker doesn’t scale the 150-dpi-created jpeg.
10. Note: some existing drawings have a larger jpeg drawing and allow Framemaker to do the scaling. The png drawings are all at the same scale as used by Framemaker.

# Producing PDF file

## Producting PDF file for Both

1. Open ArtOriginals/MSWord/Cover\_Art\_1.6.doc. Change date. Create PDF file.
2. Open ArtOriginals/MSWord/Cover\_Art\_2.0.doc. Change date. Create PDF file.
3. Copy PDF files Cover\_Art\_1.6.pdf and Cover\_Art\_2.0.pdf to FrameMaker/Graphics.
4. Start FrameMaker 10.
5. Open JDF1.book for the appropriate spec.
6. Open first file Cover.fm
7. Perform a “Find” with options “Unresolved Cross-Reference” and “Look in: Book”.

* For each one found, fix it
* When “Find” finds nothing in the entire book, go to next step.

1. Select the book.
2. Perform “Edit/Update Book”.

* In the options window, select all check boxes except “Apply Master Pages”. The “Generate” box should list the 4 files JDF1TOC.fm, JDF1LOF.fm, JDF1LOT.fm and JDF1LOE.fm.
* Press “Update” button. If there are errors, fix them.
* Error Message: Inconsistent Numbering Properties”. Check box for “Skip Remaining Inconsistent Numbering Properties Messages”

1. Select the book.
2. Perform “Save Book As”, select type as “PDF” and name as, e.g., JDF-both.pdf.

* Depress “Save” button
* Use default options (including 8.5 x 11 size) and Depress “Set” button

1. Move resulting PDF file JDF-both.pdf to pdf/. Rename it to include date,   
   e.g. JDF-both-20150925.pdf

## Producting PDF file for XJDF 2.x

1. Select FrameMaker file and perform “Send to/Compressed (zipped) folder.
2. Move zip file to some work folder, e.g TrialVersions
3. Extract zip file in the work folder.
4. Open FrameMaker/JDF1.book with FrameMaker 10.
5. Open FrameMaker/TemplateForFinalAllBlackJDF2x.fm
6. Select all documents in the book
7. Perform “File/Import/Formats…”.

* For Import from Document, select FrameMaker/TemplateForFinalAllBlackJDF2x.fm
* Press “Deselect All” button
* Click on check box “Conditional Text Settings”
* Press “Import” button

1. Continue with step 8 in “Both” above.
2. There may Unresolved Cross-References that need fixing in 2.x

## Producting PDF file for JDF 1.x

1. Same as above except for use of FrameMaker/TemplateForFinalAllBlackJDF1x.fm instead of FrameMaker/TemplateForFinalAllBlackJDF2x.fm.

# Producing HTML files

# This section has not be verified for 2015

## With Save As

1. Open JDF1.book with Framemaker 7.2, then open each document and do a “Save As”. Select appropriate html folder and then select “htm” file and change suffix to “html”.

**Note**: Framemaker 9 has a bug whereby a Wingding conversion of “l” and “n” code points to special characters in Framemaker becomes pervasive for all fonts and thus turns all “l” and “n” code points to the special symbol even though the letter is not intended for normal fonts. For example the following are “and” and “all”: “a&#9632d” “a&#9679&#9679”.

**Note**: Framemaker supports “Save As” for a book. However, the JDF book is so big that Framemaker crashes before completing the conversion.

1. Using previous version as source and new html folder as target,

* copy index.html (main file)
* copy JDF.css
* it adds lines for putting colored borders around callouts
* It changes left margins for number (2), bullets (3) and ResourceBody

1. Using Framemaker folders:

* Copy Cover\_Art.png from ArtOriginals\MSWord. The png file is produced by opening Cover\_Art.pdf with Adobe Acrobat and saving it as a png file.

1. Run script fixHTMLDocs.bash (copied from previous version of html files)

* The script changes the style line to reference JDF.css
* check for correct conversion with compareHTMLDocs.bash.

1. Edit Cover.html as follows:

* The “Save as” operation doesn’t include the pdf cover art and the Cover-1.png is too narrow.

Change:

<img src="Cover-1.png">

To

<img src="Cover\_Art.png" style="border: none;" alt="Cover\_Art.png" width="602"   
 height="785" border="0" />

<p><img src="Cover-1.png" style="border: none;"   
 alt="Cover-1.png" width="602" border="0" /></p>

1. Main file is index.html

## With RoboHelp

1. Select JDF1.book and start RoboHelp
2. Create new project. Current is JDF1.4a
3. Import Framemaker JDF1.book into new folder (htmlA2 currently)
4. Import files: JDF1LOF.fm, JDF1LOT.fm, JDF1LOE.fm and CoverBack.fm.

* Note: probably don’t have to do a Generate before this step, but didn’t try.

1. In upper left pane, drag files under “HTML Files” into JDF1 folder
2. Edit file JDF1.4a/JDF1.4a.hhc by adding XML item Elements for Cover at beginning and for JDF1LOF, JDF1LOT, JDF1LOE and CoverBack at the end, using another item as a template or a previous version. These last two steps add the files to the HTML TofC.

* Note: there is a similarly named file JDF1.4a\!SSL!\WebHelp\_Pro/JDF1.4a.hhc that the menu item Generate builds, apparently from the JDF1.4a/JDF1.4a.hhc file. There doesn’t appear to be a way to build this file via RoboHelp.

1. Select properties of Layouts (lower left pane)

* Output should be within directory selected for import.
* In Condition Build Expression field, exclude CaptionWContinued and CaptionWSheet so that table caption don’t have the continuation information for the case where the table is split across multiple pages.
* Select Table of Contents as name of project, e.g. JDF1.4a
* Make default topic be “Cover.htm”

1. In RoboHelp pane, edit Cover.htm by copying previous versions cover, but add Cover\_Art.png which is missing because RoboHelp ignore referenced PDF files.
2. Select “WinHelp”

* On right menu: Set as Primary Layout
* On right menu: Generate.

1. Copy JDF.css from previous version into WebHelp\_Pro/JDF1
2. Run ‘fixHTMLDocs.bash’ which is a script that

* creates an “old” directory for source files of command run in this script file
* changse style references with ‘fixHTMLStyles.bash \*/\*.htm’
* checks if just style changed to “../JDF.css” with ‘compareHTMLDocs.bash \*/\*.htm’
* fixes overline errors in change flags with ‘fixHTMLDecoration.bash \*/\*.htm’
* fixes offset of a callout that has a large negative margin with fixIntroduction.bash.
* fixes WingDings in the Hole appendix with fixHoleSymbols.bash.

1. Remove “old” folder which has files before running above fixHTMLDocs.bash script.
2. Move 9 files: Resources/Resources00203.jpg to Resources/ Resources00211.jpg from previous version to new version. The file generated by RoboHelp are wrong. These files came from the first version of html and then doubled in size to 1.6 x 1.1.
3. Main file is “<project name>.htm”, e.g., JDF1.4a.htm: