

xPression Document Design and Troubleshooting Best Practice

Table of Contents

[1 Choosing the Right Design Client](#)

[1.1 xDesign](#)

[1.2 xPresso for Adobe InDesign](#)

[1.3 xPresso for Dreamweaver](#)

[1.4 xPresso for Microsoft Word](#)

[2 Data Design Best Practices](#)

[3 General Document Design Best Practice](#)

[3.1 Naming Conventions](#)

[3.2 MS Word Template](#)

[3.2.1 Copy Style](#)

[3.2.2 MS Word User Defined Styles](#)

[3.3 Business Rules](#)

[3.3.1 Read Rules](#)

[3.3.2 Section Rules](#)

[3.3.3 Output Variables & Variable Rules](#)

[3.4 Attributes/Criteria](#)

[3.5 Content](#)

[3.5.1 Content Item](#)

[3.5.2 Shared Content/Rules](#)

[3.5.3 External Content](#)

[3.5.4 Subdocument](#)

[3.6 Repeating Data/Detail on the Page \(Loops\)](#)

[3.7 Revision Unit](#)

[3.8 Images](#)

[3.9 Others](#)

[4 Troubleshooting](#)

[4.1 Collect Information](#)

[4.2 Troubleshooting Checklist for xDesign](#)

[4.3 Troubleshooting Checklist for xWord](#)

*

Choosing the Right Design Client

Designing within the design client is just as important as selecting the best one for the need. Here is how best to decide allowing you to select the best asset:

*

xDesign

- * Utilizes MS Word as an authoring tool, lowering the learning curve
- * Can create different historical versions of content
- * Accepts RDB and XML data sources
- * Attribute set features provide flexibility for choosing appropriate content
- * Has built-in pending/submit/approve levels to facilitate workflow – This prevents unauthorized users from modifying content
- * Subdocument features can utilize any document created with an xPression application
- * Users need constant connectivity to the xPression server

*

xPresso for Adobe InDesign

- * Good for graphical and frame based output, such as marketing brochures
- * Does not need constant connectivity to the xPRS server
- * Requires experience working with Adobe InDesign
- * Does not support CompuSet
- * Some features in xDesign (attributes, status) are unavailable
- * Cannot have more than one document designer working on a single InDesign document at one time
- * Cannot copy and paste variable pieces from one InDesign document to another – When this is done the variables become static text in the target template
- * Document structure should be consistent with the customer data xpath structure . i.e.,

```
Content Loop A ( xpath = /A/B/C, self-adaptive=true)
----- Content Loop B ( xpath = /A/B/C, self-adaptive=true)
----- Variable C ( xpath=/A/B/C/D)
```

Instead of:

```
Content Loop A ( xpath = /A/B/C, self-adaptive=true)
----- Content Loop B ( xpath = /A, self-adaptive=true)
----- Variable C ( xpath=/A/B/C/D)
```

- * A content instance should be used if a Content Loop with xpath like s /A .

*

xPresso for Dreamweaver

- * HTML oriented output, such as web pages or e-mail
- * Requires experience working with Adobe Dreamweaver and HTML
- * Some functions need to be done in code view – Not all functions can be done in HTML view

- * Does not support CompuSet
- * Some features in xDesign (attributes, status) are unavailable
- *
 - xPresso for Microsoft Word**
 - * Customized documents including features of MS Word
 - * Good for documents that are not complex or logic driven in nature
 - * Most users already have experience with MS Word
 - * You can leverage most of the familiar MS Word editing features (for versions prior to 2007) to create documents
 - * Full support of Microsoft 2007 features
 - * Does not support CompuSet
 - * Some features in xDesign (attributes, status, external PDF) are unavailable
 - *

Data Design Best Practices

Although you won't see it when you create your documents, "working" in the background is the data source assigned to the category to which you add individual documents. xDesign uses the data in these databases to process rules and perform variable replacements during document assemblies. xDesign supports XML and relational database input.

- * Reference examples from XML Recommendations for xPression & xPresso.
- * xPression processes customer data very differently depending upon the data source.
- * RDB data is queried on demand, requiring real-time access to the RDB when the document is being assembled. Every read rule in the BDT is a separate query and round trip between xPression Server and the RDB server.
- * XML data is read from the file system into an in-memory RDB first. Document assembly then executes read rules as in-memory RDB queries.
- * Do not include fields in the schema that are never used (especially in secondary tables). The DocBrowser tool is available to detect fields that are not used.
- * Tables read many times in an assembly should be reviewed carefully.
- * De-normalize or flatten data to eliminate Read rules, wherever possible. Consolidate the records if there is a one-to-one relationship.
- * Set flags in the data to accommodate simplified criteria checks in the document.
- * As any database grows, it will decrease in performance if it is not maintained properly. Adding documents, content items, and rules to an xPression Database greatly increases its size.
- * xPression needs a specific flavor/structure to work.
- * One word.... Flat!
- * Pass only that you need
- * Flags for criteria checks versus longer valued checks

- * Lacking keys in all levels
- * Implied structures don't work
- * Performance Oriented Data Design
- *
 - Verify if the XML is optimized for xPression processing
- * Can the data be de-normalized or flattened to eliminate Read rules wherever possible?
- * Can the fields defined in the secondary tables be grouped in a manner that will reduce the complexity of the document and/or the amount of Read Rules?
- *
 - Verify if there is logic that can be executed in the data transformation/ extraction to avoid unnecessarily processing every time a document is executed.
- *
 - Verify there are no unused fields in schema especially on secondary tables
- *
 - Implement Cache Tuning once the server is close to testing and or production levels.
- *

General Document Design Best Practice

- *
 - Naming Conventions**
 - Provides a consistent approach when it relates to any design client, xAdmin or xDashboard . This also includes output files, jobs, etc.
 - LOB_
 - Dept_
 - DocShortName_
 - Recommend naming all "shared" items with a custom suffix (ex: "_S").
 - Use a suffix instead of a prefix so that the customer can order by prefix.
 - Allows for easy identification and still allows the user to maintain as they see fit.
 - Ex: "ARC- ItemStmntMaster_S "
 - Recommend renaming all criteria statements to identify the logic used.
 - Ex: "If EXTENSIONDATE <> NULL and PROGRAM_CODE <> 'MP' AND RECEIVABLE_TYPE_CODE <> 'CB'"
 - Recommend naming for xAdmin components.
 - Category – <Description>_Cat (ex: Auto_Loan_Cat)
 - Attribute Set – <Description>_Attr_Set (ex: Auto_Loan_Attr_Set)
 - Data Source Group – <Description>_DSG (ex: Auto_Loan_DSG)
 - Data Source – <Description>_DS (ex: Auto_Loan_DS)
 - Never change the attribute set after developing and testing a document unless you plan on regression testing this fully. This along with the document name, data source group, data source will be automatically updated, but it is not good practice to get into this habit.
- *

MS Word Template

The Microsoft Word template allows you to add styles and use them in any document that uses the same template. It also contains macros that allow xDesign to integrate with Microsoft Word.

*

Copy Style

Copying styles can be tricky given the process. We have created and deployed on PowerLink a white paper on this process to help you.

- * You will need to take this action on every upgrade, as macro adjustments may have been made from one build to another.
- * Backup!
- * Make sure you do this action in a separate MS Word session (without xDesign open and running on your workstation).
- * Copy only those styles you use & Normal
- *

MS Word User Defined Styles

We recommend using User Defined Styles instead of direct formatting. This allows for:

- * Better management of use and formatting
- * Definition of its use
- * What docs use this
- * Reduces inconsistencies in coding & QA issues later.
- * Allows for faster changes later with one update for all documents that use that style.
- * Recommend naming all styles with a custom prefix (ex: "U"). This will help identify custom styles when upgrading the MS Word template.
- * DO NOT base styles on Word standard styles (ex: "Normal").
- * This will cause a dependency between xPression and Word style definitions.
- * Upgrades can cause redefinition of Word styles that can cause output differences.
- * Styles should be based on "(no style)".
- * Use names that are intuitive. Keep them meaningful and functional . For example: "Para " for Paragraph styles
- * Should name based on function (ex: U PARA1) instead of attributes (ex: U TNRB14).
- * " Bul " for Bullet styles
- * " Num " for Numbered list styles
- * " Hd " for Heading styles
- * "SHd1" for Level 1 Sub-heading styles
- * "SHd2" for Level 2 Sub-heading styles
- * When Upgrading xPression Version/Builds
- * Always use the newest template
- * Copy over your custom styles and default settings
- * When Modifying the Template
- * Delete (Or backup) the HTML and XML files in the template directory
- * Always Right click and open do not double click filename
- * Pay attention to the Base Style (or 'Style Based on:' option) used when creating a custom style
- * If the base style changes all styles using this style as a base style will be changed as well. ex . Normal style is used a lot as the base style. If Normal changes, all styles that use Normal as base style will also get that change.
- * Use No Style instead of Normal to avoid updates affecting your custom styles.
- * Style references (references to 'Style Based on:' option)
- * keep them simple and relevant
- * Avoid multiple level nesting of references
- * Don't use a character style for a paragraph style reference or base and vise versa
- * Macros

- * Macros control template behavior and the specific functions required to communicate with the xPression database.
- * Never modify the Macros
- * Never copy or remove Macros from the template
- Areas to watch:
 - * Account for the extra time up-front with analysis prior to coding user defined styles in your project.
 - * User defined styles applying lines. This can be done; however, it may be better to use tables and table borders. Solicit your internal organizational tech leads on their approach.

Business Rules

Rules are the framework of every xDesign document, shaping the order in which the document contents are assembled and published. An xDesign document is executed in the exact order the rules are created in the tree like structure on the Design Pane. The evaluation of the rules starts at the top and continues to evaluate down the tree until the last rule at the bottom of the tree is executed.

*

Read Rules

- * This is dependent on the data design and ensuring the best practices we provide has been attained.
- * Ensure you have EOF checks for every Read Rule.
- * Minimize the number and complexity of read rules.
- * Each read rule is a SQL query and physical I/O round trip between xPression Server and the RDB server, if using RDB. If using XML, all reads take place in memory and thus take no physical I/O.
- * The more complex the table joins, the longer it will take to execute.
- * Create indexes for any complex read rule join or frequently-executed read statement.
- * The number of read rules should be a 1:1 for repeating information. If you have to code 3+ read rules to obtain data to use on a page, then you should re-evaluate your data design.
- * Once you code and use a Read rule, be sure to include EOF checks.
- * If you find in your code a Read rule and then a Section Rule with Criteria – Combine these.
- * A section rule used in a read loop should not appear more than once in an assembly list.
- * The Label and Go-To rules should remain on the same level *and* in the same section rule in the xDesign tree.

*

Section Rules

- Sections are designed for combining similar criteria based content together so that efficient “bypass” when this criteria is not met. This was later expanded to allow for coding of Markers and Recipients. Section rules are also used to establish the limits of a Revision Unit.
- These are exceptions to the rules below.
- * No Section Rules should be used when criteria is missing or not defined.
- * Three objects should be at minimum contained within a single Section Rule.
- Areas to watch:
 - * You should not nest Sections within Sections without criteria defined.
- * Rules for organizational purposes, much like how people use Windows Explorer is not efficient and you will encounter a large performance impact.

- * Check for unneeded or overused sections rules (should be at least 3 rules in a section rule, unless needed for recipient processing) .

- *

Output Variables & Variable Rules

- * Use Variable rules only when data is not supplied in the data.
- * Each variable rule takes memory
- * Maintain a listing of these for each document or per environment. The more you have to document, the more you think if you really need it.
- * Initialize all document variables before populating / using them.
- * Naming must match between the design client and the xAdmin definition
- * Migrations impacts on these re-mapping in xAdmin. xDesign definitions and use are ok.

- *

Attributes/ Criteria

- * Use an attribute set that matches your minimum level of criteria needs for ALL content. If you can't, but still have a minimum set, determine your common denominator and this will be your attribute set.
- * We use Jurisdiction, Language, and Effective Date . Defining new attribute set when only half the content or one coded section needs the extra data check. The section where redundant criteria would have been used would prove faster using an attribute set definition; however, the rest of the document's contents would be checking the attribute, so overall this is inefficient.
- * Defining or using an attribute set where you have unused or unmapped attributes.
- * See if there is logic that can be added to conditionally skip rules to avoid unnecessarily processing them.
- * See if attributes can be incorporated in place of repetitively used criteria checks. Do not add attributes for items that change or are added often.
- * Make sure there are no redundancies in criteria for values already being checked at the attribute level.
- * Make sure there are no unnecessary attributes defined in the attribute set.
- * Make all criteria as concise as possible. Use reverse logic to simplify the criteria if needed.
- * Watch out for "NOT EQUAL OR" constructs. Anytime you see something like "VALUE NOT EQUAL 7 OR VALUE NOT EQUAL 6", you need to fix the logic. The example statement is ALWAYS true. What is meant is: "VALUE NOT EQUAL 7 AND VALUE NOT EQUAL 6"

- *

Content

The content items that you create with Microsoft Word are the building blocks of an assembled document.

- *

Content Item

- * Make sure the fewest number of content items possible have been used to achieve the required result.
- * If there is more than one content class, make sure the classes are arranged in the order of most likely hit to least likely hit.
- * Make sure there is minimal redundant content. Consider using shared content and/or subdocuments for redundantly used content .

- *

Shared Content/Rules

- * The use of shared content to allow any document in an environment to share pieces allows for faster coding and ease of maintenance – one time change for all users. Given these can be used for any document, be sure to name these with a prefix of the “LOB/ Dept __ SourceDocument __” prior to the name, so document designers will best understand its contents and if they should use it. Shared text also has version control, whereby Universal Content has a check-in/check-out feature.
- * Sharing rules (beyond content) allow for less redundancy in coding whereby unique objects are read and processed by xPression.
- * Areas to watch:
- * These content items are shared across the environment and categories in that environment. These are not shared across environments (DEV/TEST/PROD)
- * Reverse or Circular migrations with documents containing shared text can create duplicates.
- * Contact your PS team to resolve this. We have a utility that can help now.
- * Only users with Write-level access or higher can create a shared rule.
- * Only those with Shared Admin have rights to edit these content items.
- * Shared content is currently restricted to the individual design tools. Ex: An xDesign document cannot use a content instance created in xPresso for InDesign.
- * All rules can be shared except:
- * Read
- * Table
- * Chart
- * Variable
- *

External Content

- * The external file size will impact performance. Make sure external files are not scanned in documents (usually larger in size).
- * Make sure the external files are located on a SAN or high performing network location.
- * If you suspect external files are a problem, break the link and run a test to validate.
- * Consider storing images/content in the xPression database instead of using external files.
- *

Subdocument

- * Subdocuments can be used to include content created in different design tools. Ex: An xDesign master document can include an xPresso for Word subdocument.
- * Subdocuments currently begin and end at the page boundary; shared content does not have this restriction.
- * Recommend adding page / section breaks in the master documents, not subdocuments. Subdocument will start with a blank page when published as a stand-alone document.
- * Subdocument data mapping may impact performance. Cannot use “Same Schema” mapping when xDesign and xResponse / xRevise don’t have the same default data source.
- *

Repeating Data/Detail on the Page (Loops)

- * Utilize the standard read loop structure and/or Table Rule to process repeating data.
- * Data needs to be organized correctly and efficiently as this topic is one of the largest hit to performance.
- * Always check for EOF both before entering a loop and to exit the loop.
- * Use Assembly listing (in Diagnostics in xDesign) to diagnose nesting / document loops.
- * Flatness of xDesign logic = Efficiency

- * The smaller the Assembly listing, the more efficient the document will be.
- * Stack overflows can come from:
- * infinite loop, or
- * Using a *goto* that has its label outside of the Current Section.
- * Check for improper document loops. Review the Assembly List. If you see something like the following, it is an indication there is improper nesting/looping:

```

                                </
Section>                                </
Section>                                </
Section>                                </
Section>                                </
Section>                                </
Section>                                </
Section>                                </
Section>                                </Section>
                                </Section>                                </
Section>                                </Section>
                                </Section>                                </
Section>                                </Section>                                </
Section>                                </Section>                                </
Section>

```

Areas to watch:

- * Improper looping when it doesn't involve repeating content
- * Using *for* content or recipient processing
- * Standard Read Loop: Allows for additional Variable Rule usage
- * Table Rule: Does not allow for additional object usage during the loop process.
- * No EOF checks: Performance hit

Revision Unit

Use Revision Units to divide the document into sections that can be handled individually by xRevise. Using Revision Units improve the way the Revise Carry Forward utility works, so using them should be considered for any document that may be subject to negotiated revisions. For more information, see Using Revision Units with Section Rules in the xDesign_user_Guide.pdf.

- * OP Recipients must start in a new revision unit .
- * All xRevise documents (including both master and subdocuments) should have at least one revision unit defined .
- * Recommend explicit creation / naming for all revision units (do not let xPression create/ name for you) .
- * If you rename a document, your defaulted revision unit name can change .
- * This can affect carry forward activities .
- * During execution, any time a variable changes within a revision unit, this variable becomes static .
- * Data in loops can become static .
- * This can concatenate a variable into the revision unit name to avoid this becoming static .
- * This can affect carry forward activities .

- * Recommend defining a new revision unit before changing the data value if you don't want the variable(s) to become static .

- *

Images

- * Use the recommended types found in our eBooks for CompuSet

- *

EPS

- *

IMG (Xerox)

- *

JPEG

- *

PCL

- *

TIFF

- *

Use the recommended types found in our eBooks for xPublish

- *

JPEG (.jpg, .jpeg)

- *

Group 4

- *

GIF (.gif)

- *

EPS (.eps)

- *

PDF (.pdf)

- *

PNG (. png)

- *

TIFF (. tif) (xPression supports B/W, Gray, RGB Palette, RGB TIFF, and CMYK TIFF)

- *

Windows Bitmap (.bmp)

- *

Documentum Image - Image from EMC Documentum Docbase (only supported in xDesign, xPresso for Adobe InDesign, and xPresso for Word 2007)

- *

Filenet Image - Image from Filenet (only supported in xDesign)

- *

External Images

- *

Best used from ECM, can use LAN, but be careful!

- *

Printer Resident Images

- *

Faster processing

- *

Greater control

- *

Smaller output file sizes

- *

Does allow for proofing locally with images.

- *

Importing new images

- *

Image Utility or within the Content – xPublish

- *

Image Utility (CSET Tools)

- * Within the Content – CompuSet
- * Areas to watch:
- * Know what engine you are using
- * Ensure the files you use come from your marketing/creative services team. I can't stress this enough!
- * Make sure the images are properly sized for the use
- * Match the printer resolution or it is a waste of processing and space.
- * External Images
- * That have no control
- * Volatile location
- * Printer Resident Images:
- * Final testing requires output as the proofing version from preview may not be the same.
- * Know what doesn't work:
- * Animated GIF
- * EPS with TIFF Proof/Preview
- * Compressed TIFF
- * CMYK TIFF
- * If you view all images loaded in your xPression database and find it has a suffix of "_1" or "_1_1" or longer, then a circular or reverse migration has occurred. The system upon migration found the same name and added a counter to the image to maintain its unique property assuming you wanted this.
- * To remove all images with counters to reset the system back to the original set, you must:
- * Ensure all copies are in fact the exact same. Do this by creating a new document and add all images (with names). Once verified...
- * Contact EMC DS PS to configure and run a utility in your environment.
- * When attempting to load/import images, be sure to use the Image Utility (not within the Content).
- * The content method adds the hourglass – before timeout.
- * The content method adds the Word effect – Word issues, security, access, etc.
- *

Others

- * Check for unneeded or overused sections rules (should be at least 3 rules in a section rule, unless needed for recipient processing) .
- * Check if the document uses User Exits. If User Exits are present, perform a code review to make sure they are optimally coded .
- * Only apply items (such as Markers) where needed (not on every high - level section) .
- * Replace all "smart quotes" with "straight quotes" while developing / copying / pasting. Smart quotes are not supported in xPression and can cause issues.
- * Items to consider when creating a table:
- * Empty table cells should be combined together whenever possible.
- * All space settings using line breaks (¶) should be replaced with MS Word paragraph indent and spacing features (Format menu # Paragraph...)
- * Do not allow the table width to exceed the defined width of the page.

- * Maintenance vs. Performance
- * There is generally a trade off on maintenance vs. performance.
- * A document tuned for optimal performance may not be easy to maintain.
- * A document that is tuned for ease of maintenance may not perform well.
- * Gather customer requirements for both items and find a “happy medium”. Compromise is usually necessary for high volume
- *

Troubleshooting

The most straightforward way to find a design issue would be review the document tree structure directly in design tools. An experienced designer would follow some best practices such as flatness instead of deep nesting, use Universal Content instead of include everything in a huge document, etc. Some classic design issue can also be easily observed such as using a GOTO with a label outside of the current section.

Generally speaking, an error message on UI or exception from log could be a sign of issue that may not related to document design. This could be a system bug or runtime error caused by misusing. However that does not mean it cannot be avoided by re-design the document.

*

Collect Information

xDesign provides diagnostics tools that you can use to collect useful information to help identify xDesign problems. For the first diagnose without code and runtime environment, the most useful information may include:

- * Business Document Template, which is an xml format snapshot of the document, useful when the document is not convenient to be opened in xDesign.
- * Assembly List, which is stored as an XML file record of the replacements, content items, and other information xPression Design uses to build the document for that customer. It can be used to troubleshoot an unsuccessful assembly. Assembly lists are also helpful when debugging problems where xDesign and xPression Batch produce different results. You can look at the assembly list to see the names of the content items that qualified for each assembly and determine if the assemblies are picking up the exact same content items or if xDesign did not replace a variable in a content item with the replacement value.
- * Document Instance captures preview data that is sent from the server for xDesign to display. To get MSO HTML, just preview as Word. MSO HTML can be very useful to identify if the issue is a client issue or from server side, as well as for most of xDesign fidelity issue.
- * xPression.log is always needed.

Speaking from experience, when an issue comes into engineer, the most powerful method would be debugging with a running server. The materials required for debug include:

- * Pdp including document, category, and publish profile if used
- * Customer data
- * External content if there is any
- * Word template used by xDesign
- * Output file
- *

Troubleshooting Checklist for xDesign

Problem

Solution

Some hidden DSC tags on form fields get duplicated once the content items get approved , causing xEditor to crash when the containing document's work item is opened for edit.	The root cause may not relate to approval but modified hidden code in xDesign Word template. After cleaning up some bad content items, the issue appears to be resolved.
Variables non-instantiated have their name printed on the output document	UMerge.VariableDebugMode in customerdata.properties : If UMerge.VariableDebugMode = true, it will print variable name if the variable name is null. If UMerge.VariableDebugMode = false, it will NOT print variable name if the variable name is null. It is better to initialize the variables at the beginning of the document, so every variable will be defined.
Unable to open xPression 4.5 produced .docx document in Word 2007	The table used in the template has some style issue which caused the incompatibility with Word 2007. Recreating the table resolved it.
Incorrect value returned by variable rule located in a footer which should be updated inside a loop	Resolved by usage of Frames .
When a variable is selected as "HTML formatting " in xdesign , < is not working.	For the text on <finished>, Please use < ; finished > ; in the customer data.
While protecting the content item in xDesign bullets are changing to different formats	This is a limitation in MS Word; EMC suggests that you only use direct formatting in this scenario.
Table Rule cannot read secondary DSG	The table to query for the table rule should be set as the primary table in data source group
Date Format Overridden by strange format after optional paragraph added	As a work around - the user could (from xDesign) show hidden text of their content and delete the variable with the issue (plus associated hidden text: DLSFMT, etc...) and re-insert the variable again.
Variable values does not change in xRevise when	All the text pieces are merged into one RU. If the customers want to keep the variables change for every page, they need to set up RU for every page in xDesign.
External PDF file is not generated by xPression	Issue was resolved by upgrading Adobe Reader version to 10x and installing the Japanese font package.
Header rows doesn't repeat in new page if the last table row has different number of column as header's	Workaround - create 3 rows instead of 1, make 1st and 3rd as 8 columns, merge the second row as one column
error message 6096: failed to display content in xDesign	Correctly setup the temp folder for xPression server temp files generating. it is in ofp.properties in xPression home and key/value is xPressionTempDir =C:\\xPression\\CompuSet\\temp\\
Image appears lighter in xPression	Re-created the image to be a Bi-level png format image
Font is attempted to be substituted but it is not used in the document -	Use a font tool to rename the font family to something different so xPression will not see it as part of the same font family.
Do we support page breaks inside Word universal content?	All page-level information is removed for an inline Universal Content. To keep the page-level information like section breaks, you need to insert an Universal Content content item instead of using inline Universal Content. If

How to make page N of M continue across subdocuments or Universal Content.	<p>you select not to continue the master layout, then all the page-level information of the Universal Content will be retained.</p> <p>Use Universal Content instead of Subdocument with the following setting:</p> <ul style="list-style-type: none"> * Layout: Content with Layout * Start UC From: Next Page * Page Numbering: Retained * TOC Definition: None <p>Not supported by xPression.</p>
Is the Word Field - SectionPages - supported in xPression for xPublish output?	Not supported by xPression.
When using external image references in xDesign; should the image coming in stay the same size as it is outside of xPression or would you expect the image to take on the size of the placeholder within the Content Item?	When you insert an external image, you get a placeholder, which allows you to size the image.
xAdmin OP variable mapping overrides xDesign document assembly mapping	Work as designed. If an output variable is mapped to both the master document and the subdocument data sources, only the value contained in the subdocument's data source will be returned. If you need to retrieve the value of both the master and the subdocument's data source, create two different output variables each mapped to its respective data source.
External PDF content for xPublish document doesn't publish correctly (overlapping characters) after going through LZW compression method Headers and Footers not properly working when document protection set	<p>Take the Word document and open it directly in Word and saved it as PDF then attached it to the xPression document. It resolved the issue.</p> <ol style="list-style-type: none"> 1. Put the continuous section break after the content. 2. Initially, this textpiece only have one section. After you insert a continuous section break, there will be two sections. You need to make sure the header/footer is added to the second section. (You probably need to create new page in the section 2, after setting the header/footer, delete this new page)
Unplanned extra newline before subdoc section in documents published from xRevise / xEditor	This is a special case. The content item before the subdocument is applied paragraph merge marker. But when generating work item, all the contents before the subdocument will merged into a RU. The special comment on paragraph will be removed. The workaround is move the subdocument's content into master document for this case.
xPression ignore trailing space in custom format replacement(Insert variable replacement field and during insertion apply Custom string format replacement like {FORMAT: *, })	<p>A blank ending a format can never be trusted. Too often, Word adds them spuriously and we have to compensate. Try FORMAT:*, X</p> <p>Since all the characters were read for the *, the X is just skipped.</p>
Alignment of table borders in xResponse work items is messed up when edited in online TCE editor.	Tables with editable regions aren't officially supported in xDOE . One possible workaround is to split the table into two separate content items

bullet is changed when reopening text piece in xDesign

The reason is character set of customer's Oracle is WE8ISO8859P15, the resolution is to install oracle by setting Character Set to AL32UTF8.

*

Troubleshooting Checklist for xWord

Problem

The legacy package is not showing on ECM Dashboard

OOM when loading a large size customer data XML file.

The legacy package from xWord 3.0SP1 or before is not supported in the conversion.

Where is the conversion log ?

There is no converted document opened after clicking "Check out and Edit".

The xWord4.2 packages are automatically converted when clicking "Check out and Edit" button.

TOC output could be displayed as Field Codes or collapsed.

The value of variable missed in .docx output when it's placed in Text Box control in template.

Undo can't be done with CTRL+Z

When Paragraph is created with only one UC which has criteria and it's not satisfied in the output, an extra empty line would be presented.

When opening a dialogue, it doesn't show up, and the program freezes.

Solution

The login user may not have the access right to the expected xWord4.2 package category. Go to xAdmin and check the access right in the category.

Split the XML file if memory issues occur.

If there is 3.0SP1 source template, please save it with xWord4.2 as a xWord4.2 package first.

The conversion log is in the xWordDesigner.log under a folder like "C:\Users\[User]\AppData\Roaming\xPresso\4.5SP1\Unity Designer\logs", you can find the conversion error and warning log in it.

If there is no converted document opened, there is some fatal errors happened during conversion. Please go to xWordDesigner.log to find the detail of the error message.

If user wants to check out the xWord4.2 package and keep the format unchanged, please use the "Legacy actions" in ECMDashboard and there are "check out"/"Check in"/"Cancel check out" actions and user can use xWord4.2 add-in to editing the template.

TOC field could be displayed as codes or values, depends on options setting, Word Options->Advanced->Show document content section, and check/uncheck Show field codes instead of their values. You could also right-click the TOC field and switch between the options via Toggle Field Codes. If the view of value is not expected still, you may need to Update Field in the context menu.

The Text Box in template should be larger to accommodate the variable value in the output.

Do NOT modify default shortcut keys in Microsoft Word. Otherwise, some shortcut keys in xWord may not function as expected.

If you encounter issues with shortcut keys, reset all shortcut keys for your xWord document.

Adding Content Group as the container of UC and setting criteria on the Content Group rather than UC.

Delete data file xWord Designer.data under "C:\Users\[User]\AppData\Roaming\EMC Document Sciences\xWord Designer\".

xWord add-in failed to load after opening Word even reinstalling xWord.

Clear up cached data file xWord Designer.data under

"C:\Users\[User]\AppData\Roaming\EMC Document Sciences\xWord Designer\Work"

Extra line can't be removed following a UC.

Show paragraph marks in Word, highlight the Paragraph mark using keyboard, in the Font dialog, and check Hidden in the Effects section.

*

Troubleshooting Checklist for xInDesign

InDesign takes particular value only from Document structure to be avoid :

first XML data set and puts it for all pages Content Loop A (xpath = /A/B/C, self-adaptive=true) instead of appropriate data sets .

----- Content Loop B (xpath = /A, self-adaptive=true)

----- Variable C (xpath =/A/B/C/D)

The document structure is not consistent with the customer data xpath structure , which would make server in vague and don't know how to assemble the document: it don't know which principle would be following, by document structure or by xpath s tructure.

Since Content Loop B is redundant because server will pull out the first record of /A always , so a content group is more preferred since it would be efficient than using loop.