Docmosis Tornado v2.9.2 Release Notes

Jun 2022

Environment Changes

#	Change
1	

Functional and API Changes

#	Change
1	Improvement to template-merging to bring "default" style paragraph settings of sub-template into "Standard/Normal" style of sub-template. This fixes issues of sub template styles changing during merging. Specifically, where sub templates with custom styles that are derived from Normal/Standard have implied dependencies on the "default" style.
2	User console now provides a search filter for finding templates by name or by folder: When used to filter by a name:
3	User console now allows selection of "TXT" output format instead of "DOC" format. Console allowed formats are now PDF, DOCX, ODT, TXT.
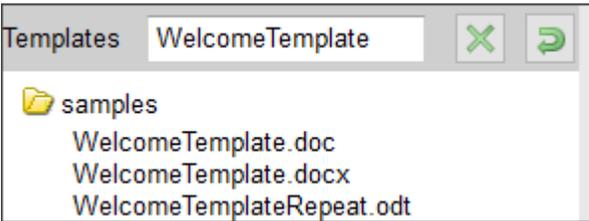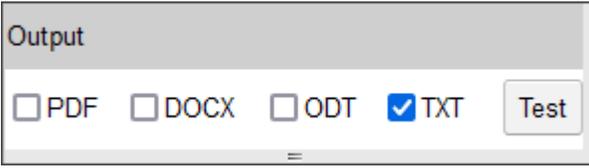
Bug Fixes / Technical Changes

#	Change
1	Fixed default logging configuration that was hiding low level issues (such as bad SSL configurations)
2	Remove logging of credentials used for AWS and Azure template stores.
3	Fixed out of memory error when listing larger numbers of templates (over 1000) with AWS S3 as source of templates.

#	Change
4	Fixes mail dispatch to not set null mail header (which could cause a failure in some mail gateways)
5	Updated OpenAPI configuration to use configured Tornado port rather than default
6	Fixed two template-comments on the same line causing an analysis error that was not helpful.
7	Updates to template-merging to remove blank space where the rendered sub-template is a blank result.

See below for older release notes.

Docmosis Tornado v2.9.1 Release Notes

Apr 2022

Environment Changes

#	Change
1	The access key, if configured in the Tornado server, can now optionally be supplied to API calls (REST service endpoints) as header in the request. This supports systems (such as Power Automate) that can apply such tokens globally to all requests.

Functional and API Changes

#	Change
1	

Bug Fixes / Technical Changes

#	Change
1	Fixed potential remote code exploit in specifically crafted DOCX templates.

See below for older release notes.

Docmosis Tornado v2.9.0 Release Notes

Feb 2022

Environment *Changes*

#	Change
1	Support for LibreOffice 7.3
2	Support for Java versions up to 17 (minimum Java version is now 8)
3	`log4j.config.file` parameter is no longer supported and logs a warning if used. Superseded by `java.util.logging.config.file` and Java Util Logging configuration.

Functional and *API Changes*

#	Change
1	Security Updates – dependencies have been updated to bring Tornado up to the latest in security awareness. In particular: - log4j has been removed (since downstream dependencies still have security warnings) - Tomcat engine updated - Many downstream dependencies updated or removed using OWASP guidelines
2	QR codes are now supported. The implementation is similar to the existing barcode implementation but with some further simplifications.
3	IMB (Intelligent Mail Barcode) format is now supported.
4	Template sourcing from AWS S3 and Azure Blob Storage are now official. The console now supports configuring these as a source of templates.
5	New template-variables to obtain information about the current template being rendered: `<<$templateName>>` - the name of the template `<<$templateFolder>>` - the folder containing the template `<<$templatePath>>` - both the above combined
6	Barcodes can now use Docmosis template features. This allows barcodes to be dynamically constructed using expressions and variables.

#	Change

7	OpenAPI support is now provided. The openAPI definition can be examined by the console:
	
	IRL Status URL WADL OpenAPI
	
	Or by using the URLs:
	`http://<location>/api/openapi.json`
	`http://<location>/api/openapi.yaml`

8	A new form of combining templates ("Coordinator Templates") has been created. This form of combining templates is suitable for document-level type appending, where content is not inserted into a document but instead appended after. This is more appropriate for certain use-cases and reduces complications around headers and footers. The new "coordinator" template type provides "instructions" on how to combine a set of other templates. For PDF, this can "stitch" the final artifacts into one PDF document.

9	Multiple templates/outputs can now be specified in the render call to have several templates rendered at once. Previously only one name could be specified. Specifying multiple templates will result in multiple documents being returned in zip format except in the case of PDF, where the output can alternatively be combined into a single document automatically.
	
	Multiple template names are separated by a semi-colon in the `templateName` parameter. Eg:
	`templateName=coverLetterTemplate.docx;letterTemplate.docx`
	and the `outputName` can specify a single (combined) pdf:
	`outputName=myLetter.pdf`
	or a zip of each document
	`outputName=myCoverLetter.pdf;myLetter.pdf`

Bug Fixes / Technical Changes

#	Change
1	The backward slash (\ ) character is supported for template paths in addition to the forward slash (/).
2	Document properties can now use <<field>> notation to substitute during document population. This includes custom properties. Note, this functionality is only fully supported with Libre Office version 7.3.
3	Static barcodes (ie defined only by the template and not dependent on any data) now populate even if no data at all is passed to the render process. Previously, some data (even unrelated) had to be present.
4	Template merging – improvements have been made to improve the retention of styles when merging templates without the need for work-arounds.
5	Render output names now support extended characters such as Japanese character codes.

See below for older release notes.

Docmosis Tornado v2.8.6 Release Notes

July 2021

Environment Changes

#	Change
1	New parameter allows Keep Alive to be set on http connections. This helps avoid idle-disconnection in envrionments with aggressive firewalls (Azure). `    keepAlive=true|false` This can be specified in the environment: or on the command line. By default this is disabled.

Functional and API Changes

#	Change
1	none

Bug Fixes / Technical Changes

#	Change
1	Launching has been improved to short-circuit quickly when there is no configuration (eg first-run).

See below for older release notes.

Docmosis Tornado v2.8.5 Release Notes

July 2021

Environment Changes

#	Change
1	none

Functional and API Changes

#	Change
1	none

Bug Fixes / Technical Changes

#	Change
1	Improved support for EMF and WMF images.
2	Fixed logging of bad license key information which was hidden when mis configuration occurred.

See below for older release notes.

Docmosis Tornado v2.8.4 Release Notes

May 2021

Environment *Changes*

#	Change

1	New parameters allow Mail System Configuration to be fully configured from the command line / Docker configuration:
	`mailServerConnectRetryMaxTimes`
	`mailServerConnectRetryMinWaitMillis`
	`mailServerConnectRetryMaxWaitMillis`
	`mailServerConnectRetryRebuildTransport`
	`mailServerSendRetryMaxTimes`
	`mailServerCustomHeadersAdd`
	`mailServerCustomHeadersName`

2	Introduce a file-system lock file to ensure another Tornado launched on a different port could not break the Tornado working caches.
3	Outbound http connections now set the user-agent to DocmosisTornado/<version> to allow Tornado to be identified.
4	Updated connectivity to Amazon AWS (http pool increased from 50 to 100). This only applies where connectivity to Amazon AWS is configured.

Functional and *API Changes*

#	Change
1	Corrected `isBlank()` template function to return "`true/false`" not "`1.0/0.0`". This incorrect behavior was introduced in Tornado version 2.8.0

2	New settings can be applied when generating PDF documents:
	`pdfRestrictPassword`
	`pdfRestrictPrinting`
	`pdfRestrictEditing`
	`pdfRestrictCopy`
	`pdfRestrictAllowAccessibilty`
	These are new parameters for the "/render" REST service.

3	Template changes are now detected and processed when the source template timestamp changes, even when the time stamp of the template becomes older. This means that when a newer template is replaced by an older template, the older template now becomes active.

#	Change
4	Updates to template-merging (`ref:` and `refLookup:`):
	- Manual insertion of a small leading line in the sub template is no longer required in most scenarios
	- Processing of sub templates cannot override the style in the master/parent templates. This was possible where sub-templates had a "default-font" setting (as opposed to the Normal/Standard style) and LibreOffice versions before 7 are used.

Bug Fixes / Technical Changes

#	Change
1	Tweaks to defend against badly constructed zip files.
2	Fixed a bug where the style of a sub template could affect the style of another sub template in a tree of template-merges due to a miscalculation of unique style names.
3	Fixed unhelpful error message when processing images from URLs that return a blank image.
4	Fixed unhelpful error message when render invoked with null data parameter.
5	Updated text type detection to allow for BOM header

See below for older release notes.

Docmosis Tornado v2.8.3 Release Notes

Dec 2020

Environment *Changes*

#	Change
1	Beta integration with MinIO for template storage is available. Contact Docmosis support if you would like to evaluate these features.
2	Sample data now generated for <<html:>> template fields
3	Render data now default excluded from logging at all levels. It can be enabled for logging with new configuration setting: `log.data=true` and it will be logged at INFO level.
4	Support for Libre Office 7 now present. Currently LO 6.2.8.2 is still recommended general purpose version.
5	Improvements have been made to processing of large numbers which under Java 15 could be rendered as 1e15 for example rather than a long string of digits.

Functional and API Changes

#	Change
1	Added Content-Length header to render server http response
2	Image substitution from URL data sources better detects error with retrieved image. If invalid, normal error processing will be applied. Previously, the image would be blank.
3	Added a check for a certain type of template pattern that can crash LibreOffice version currently available. Such templates will now be rejected.
4	Allowed the extra-long version of the hyphen (char code 8212) character to be used in field names as a hyphen. This is because Word likes to replace the hyphen with the long or extralong version whilst editing.

Bug Fixes / Technical Changes

#	Change
1	Updated template-merge processing to improve preservation of styles from sub-templates. In cases where the template style exists in both the master template and sub-template, in some cases the sub-template style could be removed so the master template style applied.
2	Fixed issue leaving a temp file in the temp folder during processing where templates were being merged.
3	Fixed issue where empty table cell tags could cause a stack overflow error (only occurred in unusual/invalid templates)

#	Change
4	Fixed issue with some text-format output documents having an invalid byte order mark.
5	Fix issue with artificial sample data elements were generated for Conditional Column (cc_xx) template fields.
6	Fixed issue with char() template function throwing an error if given blank data to process.
7	Fixed issue where expressions comparing '1F' and '1D' type of strings would compare them as floating point numbers meaning 1F=1D which is incorrect for strings.
8	Fixed issue with template squote() function raising an error when passed blank/null data.

See below for older release notes.

Docmosis Tornado v2.8.2 Release Notes

Apr 2020

Environment *Changes*

#	Change
1	Beta integration with AWS S3 and Azure Blob Storage for template storage is available. Contact Docmosis support if you would like to evaluate these features.
2	A new parameter (installSamples) has been added to allow the installation of the sample template(s) to be turned off. Command line: `java -DinstallSamples=false …` Environment: `set DOCMOSIS_INSTALLSAMPLES=false` By default, Tornado installs sample templates into a "samples" folder at startup.

Functional and API Changes

#	Change
1	Field parsing has been enhanced to allow data fields to be optionally specified with brackets around the name and when doing so, hyphenated names in data can now be used: `<<[net-value]>>` will look for data key "net-value".
2	New Template directive <<list:reset>> can be used to reset the numbering of numbered lists when inside repeating sections. There are use cases where the repeated copies of the list should not continue the numbering from the previous iteration.
3	New template function char(int) which allows characters to be specified by Unicode, html or hexadecimal code.

Bug Fixes / Technical Changes

#	Change
1	Outbound email using the render storeTo:mailto directive validation changed to allow modern email address domains which are longer than .com such as name@x.property.
2	The check-period for changes to templates has been adjusted to 1 second (was 15 seconds in v2.8.1). This default is suitable for all environments (dev to prod).
3	Fixed an issue where combining template-merging with multiple output formats would fail to process.
	Fixed issue with template squote() function raising an error when passed blank/null data.

See below for older release notes.

Docmosis Tornado v2.8.1 Release Notes

Mar 2020

Environment Changes

#	Change
1	none

Functional and API Changes

#	Change
1	Functions that specify locale (such as `dateFormat` and `numFormat`) can now specify it also using common notations with language and country. Eg `en_US` for US English or `de_CH` for Swiss German.
2	Template function `numFormat()` has a new additional parameter: `formatIsLocalized`. It specifies whether the given `format` string is using characters specific to the given locale, rather than default characters. For example, the default character for year is 'y', but if using the French locale with `formatIsLocalized=true`, it is 'a'. This parameter defaults to true (preserving current behavior).
3	Template function `dateFormat`() has two new parameters: `ouptutFormatLocalized` and `inputFormatLocalized` which respectively indicate whether the given `outputFormat` and `inputFormat` use characters specific to the given `outputLocale` or `inputLocale`. For example, the default character for year is 'y', but if using the French locale for output, with matching `outputFormatIsLocalized=true`, it is 'a'. These parameters both default to `false`.
4	New template-functions to assist with diagnostics: `locale([country or language, name or code])` – report the ID for a locale matched with the given parameter `localeInfo([country or language, name or code or ID])` – report diagnostics for the matched locale `localeDatePattern(pattern, locale)` – report the given pattern in localized format in the given locale.
5	Minor updates to the Tornado GUI – template list now shows folders first and non-case-sensitive ordering, the Template Refresh and Open buttons are always visible without scrolling, provide visual feedback when fetching sample data etc.

Bug Fixes / Technical Changes

#	Change
1	Implemented template-locking to allow concurrent reading and writing of a template to be serialized automatically. This resolves issues where templates are in use during attempted updates.

#	Change
2	Improved processing a corner case in error handling in dev mode not correctly writing the error into a paragraph leading to a corrupted document.
3	Improved Locale lookup to be deterministic under different versions of Java – always finding the same locale if available for the same lookup term.
4	Fixed an issue with render-parameter validation failing when tornado being used in case-insensitive mode for data.
5	Improved processing a corner case in error handling in dev mode not correctly writing the error into a paragraph when inside template-comments leading to a corrupted document.
6	Added logging of supported SSL protocols and ciphers at startup.
7	Added ability to use MD5 checksums rather than timestamps when comparing templates from the source with the cache.
8	Fixed template-expression processing issue where the closing round-bracket character inside quotes was being counted as parameter boundary.
9	Internal Tomcat engine updated to version 7.0.100 from 7.00.65.

See below for older release notes.

Docmosis Tornado v2.8.0 Release Notes

Dec 2019

Environment Changes

#	Change
1	A new parameter (address.listen) has been added to control the network interface that Tornado listens on. This can be passed via the command line or via environment variables: Command line: `java –Daddress.listen=127.0.0.1 …` Environment: `set DOCMOSIS_ADDRESS_LISTEN=127.0.0.1` By default, Tornado listens on all network addresses

Functional and API Changes

#	Change

1	The Tornado API URLs have changed from. If you were using for example:
	
	`https://myTornado/`**`rs`**`/render`
	
	it is now:
	
	`https://myTornado/`**`api`**`/render`

2	The `/getTemplateStructure` service end point returns more detail about the template now and the response format has changed to accommodate the extra information. The response now returns details of data-lookup fields within expressions, set-variable elements and template-reference elements. Please see the Web Services Guide for details about the response.
3	The `/getSampleData` service end point has been updated to provide sample data for fields within template expressions (<<{ xxx }>>) and set-variable fields <<$m=abc>>.

#	Change

4	New template function dateAdd to allow simple adjustments to dates. For example add a day to myDate:
	
	`<<{dateAdd(myDate, 1, 'day')}>>`
	
	subtract 3 weeks from myDate
	
	`<<{dateAdd(myDate, -3, 'weeks')}>>`
	
	The units dateAdd allows are millis, seconds, minutes, hours, days, weeks, months, years and can be specified as singular (day) or plural (days).
	
	See the Template Guide for more detail.

5	New template function dateDiff to compute the difference between two dates. For example, to compute d2-d1 in days: `<<{dateDiff(d1, d2, 'days')}>>` The units dateDiff allows are millis, seconds, minutes, hours, days, weeks, months, years and can be specified as singular (day) or plural (days).
6	New template function numToText which will write out numbers in English text. For example: `<<{numToText(1024)}>>` will output "one thousand and twenty four". `<<{numToText(302.24)}>>` will output "three hundred and two point two four". See the Template Guide for more detail.
7	New template function ordinal which will write out numbers in English text. For example: `<<{ordinal(1)}>>` will output "1st". `<<{ordinal(2, 'suffix')}>>` will output "nd". `<<{ordinal(23, 'long')}>>` will output "twenty third". See the Template Guide for more detail.
8	New template function numToDollars which will write out numbers in Dollars and cents. For example: `<<{numToDollars(1024)}>>` will output "one thousand and twenty four dollars". `<<{numToDollars(32.24)}>>` will output "thirty two dollars and twenty four cents". See the Template Guide for more detail.

#	Change
9	New template function mapi() which is the same as the "map" function but ignores case when comparing values. See the Template Guide for more detail.
10	New template function isNumber() will determine whether the input value is a number. This can allow conditional processing based on whether the data is numeric or not. For example: <<cs_{isNumber(invoiceNumber)}>> Process numeric invoice <<else>> Process non-numeric invoice <<es_>>
11	New template-variable: $nowUTCFormat – provides the date format used for the $nowUTC value. which allows the new dateAdd() and dateDiff() functions to base calculations on the current date/time.
12	When rendering documents to text format, the output is now UTF-8 format with no "byte order mark".
13	Added support for the FODT document format for use as templates and output documents.
14	Added support for Tables inside shape objects for Word templates.

Bug Fixes / Technical Changes

#	Change
1	Updated authentication system to redirect on success to a calculated url which allows for Load Balancers (with SSL termination) and proxies.
2	Fixed template processing issue with simple empty 1 line nested rs and cs failing to validate
3	Fixed issue where deeply nested 1 line template structure: <<rs_>><<cs_>><<cs_>><<cs_>><<cs_>><<es_>><<es_>><<es_>><<es_>><<es_>> type of structure would fail to analyze.
4	Added yyyy-MM-dd'T'HH:mm:ss.SSS'Z' as a default input date format
5	Fixed issue where bottom borders (and probably top borders) were not processing properly when conditional columns where used and repeating rows were meant to dictate the border.
6	Fixed expression processing regarding negation function and functions like isBlank() which would fail to process.
7	Fixed issue with parsing where internal content (soft page breaks) were causing skip sections to overlap for an <<rs_>><<cs_>> on one line scenario and would fail to analyze successfully.
8	Improvements have been made to processing the styles of sub-templates to preserve more of the original sub-template design.

#	Change
9	Updated image processing to detect the case where an image draw frame (bookmarked for Docmosis) has multiple images inside. This caused corrupt output since it is not expected so now it is raised as an error.
10	Added a step in the sequence of document creation which improves the accuracy of PDF output in some corner cases.
11	Updated image bookmark processing to allow for another quirk of docx processing (SVG tags inside frame and before adjacent bookmark).
12	Detection of SVG placeholder in ODT template causing duplicate images in template with latest versions of LibreOffice. This raises an error in the template during analysis now indicating the problem with the placeholder image.
13	Allow data to specify images from URLs with a second prefix (since the typo is common): [imageURL:…] (in addition to existing [imageUrl:…].
14	Updated library loading to automatically enable the "custom loader" if Java 8 or later is detected.
15	Fixed issue with Zip processing which could fail to process templates with EMF or WMF images.

See below for older release notes.

Docmosis Tornado v2.7.3 Release Notes

July 2019

Docmosis core remains at version 4.4.1

Tornado Changes

#	Change

	Turned render queue-depth defence off by default. This was rejecting requests when a queue of more than 20 renders was reached. It can be enabled and configured with the following settings:
	
	 ```render.queue.defence.enabled=true (default=false)```
	```render.queue.perServer.user.queuedRendersMax=n (default=20)```

Web Service API Changes

#	Change

New Core Engine Features

#	Change

Bug Fixes / Technical Changes

#	Change
1	

See below for previous 2.7 release notes.

Docmosis Tornado v2.7.2 Release Notes

Feb 2019

Docmosis core remains at version 4.4.1

Tornado Changes

#	Change

Web Service API Changes

#	Change

New Core Engine Features

#	Change

Bug Fixes / Technical Changes

#	Change
1	Fixed a bug where calls to the Render service ("/render") using a multipart or url-encoded form content type would ignore accessKey configuration in Tornado settings.
2	Reworked logging initialization so that the log.level parameter can be used to override logging even when a custom logging configuration file has been specified. Previously the log file configuration would override the log.level setting.
3	Updated new parameter / environment variable processing to allow a blank value to be passed to remove any pre-existing setting.

See below for previous 2.7 release notes.

Docmosis Tornado v2.7.1 Release Notes

Dec 2018

Docmosis core updated from version 4.4.0 to 4.4.1

Tornado *Changes*

#	Change
1	All tornado parameters can now be passed from the command line and by using environment variables. This allows Tornado to be completely configured (or adjusted) by scripts or the operating environment without needing to manually interact with the web console. Related are new optional command line args to Tornado: a. `java –jar docmosisTornado.war` **`help`** - print the help including all options b. `java –jar docmosisTornado.war` **`clearConfig`** - to wipe any saved configuration

Web Service API Changes

#	Change

New Core Engine *Features*

#	Change
1	New template functions: - replaceStr(source, searchFor, replaceWith [, ignoreCase]) - replaceFirst(source, searchFor, replaceWith [, ignoreCase])

Bug Fixes / Technical Change*s*

#	Change
1	Improved converter bootstrapping process to prime the converter with a more realistic document. This avoids the increased processing time for a first-time-used converter.
2	Fixed issue where table-cell colouring wasn't being applied (when driven by data eg '<bgcolor="#ff0000"/>') to table cell when field providing the style information was number or bullet listed
3	Fixed issue with DOCX image substitution from DOCX templates that could result in the image not being sized correctly and the text 0.00.0 being displayed.
4	Fixed issue where <<cc_>>, <<rr_>>, <<cr_>> tags might be ignored if list-styled (bullet or numbered) in the template.
5	Fixed an issue with TableAnalysis which could mis-track which cell is being processed across the row.

#	Change
6	Adjusted error handling to not report bad section matches when ending a list if errors have already been reported. This stops two errors being reported for one problem.
7	Fixed initialization issue where preferences could not be saved via the console if the previous preferences were bad or the key had expired.
8	Improved search for default Libre Office install when properties are defaulting for first use.
9	Improved validation and tool tips for Office Dir to better validate and guide user.
10	Fixed end of line characters with packaged startTornado.sh script for linux.

See below for previous 2.7 release notes.

Docmosis Tornado v2.7 Release Notes

Oct 2018

Docmosis core updated from version 4.3.0 to 4.4.0

Tornado Changes

#	Change
1	**Support for Java up to version 11** Note: Java 11 will give Warnings during Tornado Startup such as: `WARNING: An illegal reflective access operation has occurred` These can be ignored for the time being as they won't affect Tornado's execution. A future release of Tornado will resolve these issues when related libraries become Java 11 compliant.
2	**Support for LibreOffice 6**

Web Service API Changes

#	Change
1	**New end point "/status"** The "`/status`" end point goes further than the "`/ping`" service, confirming the engine is ready to render documents.
2	**New render response header "X-Docmosis-Document-Errors-Detected"** HTTP response header "X-Docmosis-Document-Errors-Detected" is set when rendering in Dev mode. This allows the client to determine if the rendered document is known to have errors.

New Core Engine Features

#	Change
1	**Improved Error Handling in Dev Mode** Top level errors now written closer to the location of the error rather than top of document.

#	Change

2	## Improved XML and JSON Sample Data Generation
	
	Generated JSON data is now more compact.
	
	Sample data generation now supports nested names for fields. For example:
	
	`<<a.b.c>>`
	
	Generates the sample data:
	
	`{"a": {"b": {"c": "value1"}}}`

3	## New sentenceCase(), isBlank(), ifBlank(), and squote() Functions
	
	Use `sentenceCase()` to convert text data to sentence case format:
	
	`sentenceCase('this is a sentence.') => This is a sentence.`
	
	Use `isBlank()` or `ifBlank()` to test and act based on blank data values:
	
	`isBlank('') => true`
	
	Use `squote()` to replace double quotes with single quotes in a string literal or data to allow quote-escape:
	
	`squote('This sentence"s quotes.') => This sentence's quotes.`

4	## New Built-In Template Variables `$num` - the current counter into the data. `$num1, $num2...` – provide the absolute index into the repeating sequence starting at zero (i.e. 1,2,3,4,5,6...). `$itemidx1, $itemidx2...` – provide the absolute index into the repeating sequence starting at zero (i.e. 0,1,2,3,4,5...). `$nl` - a new line character. `$nowMS` – the current UTC time in milliseconds since epoch `$nowUTC` - current UTC time as an ISO 8601 string `$quot` - a single-quote character
5	## Improved Numbered List Processing Now handles more complex cases of: - conditioning out some numbered items and resuming numbering - lists continue after sub-lists rendered - allow lists to continue from master template into sub-template using a new "<<list:continue>>" directive

Bug Fixes / Technical Changes

#	Change
1	Added ability use SSL for remote converters.
2	Improved the way URLs work to include context for items displayed in the console (e.g. ping, status, wadl). This resolves some issues around when deploying Tornado to different Tomcat contexts.
3	Improved template-functions to not raise errors when given null values.
4	Improved rendering of merged documents by renaming styles to remove conflict of styles.
5	Updated further functions to handle null/empty params better
6	Updated remote converter to preserve the extension of the files being transferred. This fixes an issue with HTML injection where the file encoding is not correct if Libre Office imports the file and doesn't realize it is html.
7	Added <<list:continue>> directive to allow sub-templates to specify they want to continue numbering from a master template.
8	Improved list processing to track the lists rendered so that stripped lists (e.g. conditional sections) don't break linkages between lists.
9	Updated GetTemplate* services to log the exception trace when a Not Found occurs since it's quite a normal error.
10	Fixed several getX() services that could download a file with a "/" in the content-disposition.
11	Fixed issue in Tornado configuration which expects the template directory to be writable when saving configuration.
12	Updated tornado to be able to be configured with a "log.dir.override" setting to log outside of the configured working dir (helpful in Docker environments).
13	Fixed bug in TemplateContext constructor which wasn't correctly removing duplicate "/" characters.
14	Fixed bug in SimpleJSONTemplateStructureProcessor.finish() which wasn't keeping the adjustment to the indent variable.
15	Fixed issue with SimpleTemplateTable.hashCode() which was not correctly hashing the array.
16	Updated repeating section processing to correctly handle more complex cases of repeating and variable references. This also fixes a corner case problem with <<pageBreakNotLast>> not working after an internal <<rs_a.b.c>>blah<<es_>> upset the indexing.
17	Fixed Thread leak in MailSystem. When Transport instances are being cached AND a TransportListener is attached, the JavaMail system spawns a thread per Transport for each Thread a Transport was created in.
18	Fix to writing in-document errors into header and footer (some where hiding).
19	Updated bookmark and image processing to be able to use a bookmark that appears just after the image frame rather than wrapping the image. This helps with imports of Docx files in LO5 and looks like it helps with DOC files where the image has a different wrap mode.
20	Corrected processing of [F-L] type range directives which were doing the wrong thing under some cases.
21	Updated stepping and indexing processes - allows more flexible ranges e.g. [1,2,1,1,2].
22	Improved error messages from template-functions - provide function name in message.

#	Change
23	Fixed issue resulting in corrupt documents when new line characters in data were mixed with html-like (<b>, <u>, etc.) directives.
24	Fix to HTML injection where <<html:xxx>> fields being last element in a paragraph could leave "repairme" message.
25	Improvements have been made to the auto-detection of the LibreOffice/OpenOffice locations for the first-runs of Tornado when no configuration has been created.

Docmosis Tornado v2.6 Release Notes

Feb 2018

Docmosis core updated from version 4.3.0 to 4.3.1

Tornado Changes

#	Change
1	none

Web Service API Changes

#	Change
1	none

New Core Engine Features

#	Change
1	

Bug Fixes / Technical Changes

#	Change
1	Fixed issue with image processing particularly for docx import into LibreOffice 5. An image at the start of the table as the first element in a document could sometimes fail to be recognized as an image for replacement.

Docmosis Tornado v2.6 Release Notes

Nov 2017

Docmosis core updated from version 4.2.0 to 4.3.0

Tornado Changes

#	Change
1	Sample data creation has been improved to recognize barcodes and generate a barcode rather than an image display.

Web Service API Changes

#	Change
1	**Render can now stream result base64 encoded in payload** The "`/render`" end point has a new parameter: `streamResultInResponse` which can be set to "`y`" or "`true`" when streaming the resulting document back. Setting this flag will mean that the document will not be returned as a binary stream, instead it will be returned in the JSON or XML response. The JSON/XML response will include a key "`resultFile`" which is the base64 encoded document.
2	**New end-point "/getSampleData"** The new end-point can generate sample data for the selected template in XML or JSON format.

New Core Engine Features

#	Change

1	**Number Formatting more control over Locale**
	
	The "`numFormat`" template now supports a new optional parameter to be able to enable / disable the application of the specified Locale when parsing the supplied data:
	
	`<<{numFormat(value,  outputFormat, locale, applyLocaleToInput)}>>`
	
	The default behavior is to apply the locale to both the input and output, however this is incorrect in some circumstances.
	For example:
	
	`<<{numFormat('0.0257', '#.##0,##', 'nl')}>>`
	
	Incorrectly produces "257" because the "nl" locale is applied when parsing 0.0257 into a number. This can now be corrected by disabling the parse-phase conversion:
	
	`<<{numFormat(0.0257, '#.##0,##', 'nl', false)}>>`
	
	So the correct result ("0,03") is produced.

Bug Fixes / Technical Changes

#	Change
1	Improved template-expression processing: - processing of unary operators (NOT in particular) - processing of more complex logical expressions - booleans handled better.
2	Improvements to streams dealing with UTF-8 content
3	Changed behaviour of DataProviderBuilder.addJSONFile(File) to assume UTF-8 encoding of file contents.
4	Improved shutdown process with regards to closing the stderror stream and possible blocking scenario.
5	Fixed issue in "getTemplateStructure" service that was not cleaning up resources in a timely manner.
6	Improved document processing when merging templates and injecting HTML content to handle more complex cases involving page layout changes and large tables. The result is the layout is more consistent with the templates involved.
7	Improved issue with LibreOffice 5 installations on Windows platforms where OpenGL processing could cause DocX processing in particular to fail.
8	Improved the "numberFormat" template function to allow the locale to now be OPTIONALLY applied to the input values. This is required
9	Improved barcode processing to report errors if barcode is longer than 200 chars.

#	Change
10	Fix to another corner case of plain text fields not being recognized where expressions where using the ">" greater than operator.
11	Template processing improved with fixes for complex scenarios where errors in templates were unable to be reported and would result in a general failure to process the template instead of correctly indicating the error in the template.
12	Fixed issue where blank "fillin" fields in Word templates were being raised as a general error with no helpful specifics.
13	Fixed template processing error where a table contained repeating rows then a conditional row directive but with no rows inside the condition.
14	Improved "null" error message when a problem occurs processing a template so that implementation can be corrected.

Docmosis Tornado v2.5 Release Notes

Jun 2017

Docmosis core updated from version 4.1.4 to 4.2.0

Tornado *Changes*

#	Change
1	Email support for TLS and SSL have been added allowing many more email gateways to be used directly by Tornado.
2	HTTP and HTTPS configured ports are now tested before attempting to start the server and a more useful error message is provided.
3	Fixed an issue causing a warning to be written at start-up if the user doesn't have access to the Windows Registry outside the user's area.
4	The dummy-data function has been improved to identify hyperlink fields and the new "op:" fields and create dummy data with the correct key.

Web Service API *Changes*

#	Change
1	The Render function now supports url-encoded-form payloads supporting a greater variety of client applications.
2	There is a new end point "/renderForm" which expects to read data as normal http form elements (unlike the "/render" endpoint which expects a "data" element to provide all the data). See the Web Services Guide for more information.

New Core Engine *Features*

#	Change
1	### Comments in Templates The templates can now contain comments. There are two sets of delimiters: <<## and ##>> and <</* and */>> See the Template Guide for more information.

#	Change
2	**Optional Paragraph Fields** A new field type (<<op:xxx>>) will strip the containing paragraph when no data is present for the field. This can be handy for fields like "addressLine2" which are often blank and should be removed as simply as possible. This can now be done using <<op:addressLine2>>. See the Template Guide for more information.
3	**PDF Form Fields can be Pre Filled** Templates (ODT format templates only) can create Fillable PDF form fields and pre-populate them. This means that completed or semi completed electronic PDF forms can be generated. See the Template Guide for more information.

Bug Fixes / Technical Changes

#	Change
1	Fixed issue where dynamic images in repeating sections would not render when using versions of Libre Office after version 5.0.
2	Fixed issues where Libre Office versions after 5.0 would not process a template that was open and modified.
3	Improved diagnostics logging for 32-bit / 64-bit mismatch between Java and Libre Office
4	Improved error message when missing section ends are detected
5	Increased plain text default max length from 150 to 1024 characters to support long template expressions
6	Fixed corner-case "null" error when processing complex tables with varying numbers of columns in rows that are repeating.
7	Fixed some character stream outputs which were not writing in UTF-8 explicitly.
8	Improved date parsing to fall-back if performing "Strict" matching and strict matching fails. This provides date parsing that succeeds correctly even when ambiguities are present.
9	Fixed HTML injection issue where inserting blank HTML into a table would leave a "repairme" indicator
10	Improved document loading to explicitly disable macros in templates when loading.

Docmosis Tornado v2.4 Release Notes

Feb 2018

Patch release to improve Tornado stability.
Docmosis core updated from version 4.1.1 to 4.1.2

Tornado Changes

#	Change
1	None.

Web Service API Changes

#	Change
1	None.

New Core Engine Features

#	Change
1	None.

Bug Fixes / Technical Changes

#	Change
1	Added new response header to Render service. When rendering in Dev Mode (devMode=y), the response will include the header: `X-Docmosis-Document-Errors-Detected` with a value of `true` or `false`. This means that in Dev Mode, when a document that contains errors is streamed back, the response HTTP header will indicate whether errors are present in the document. This is not relevant to Production Mode (devMode=n, which is the API default) since if an error is detected no document will be returned.

Docmosis Tornado v2.4 Release Notes

Dec 2017

Patch release to improve Tornado stability.
Docmosis core updated from version 4.1.0 to 4.1.1

Tornado Changes

#	Change
1	None.

Web Service API Changes

#	Change
1	None.

New Core Engine Features

#	Change
1	None.

Bug Fixes / Technical Changes

#	Change
1	Improved converter pool reliability when issues occur launching/repairing/shutting-down the converters. The pool is now resilient to unexpected failures during these steps. In corner cases, particularly in single-converter configurations, this could stop document production.

Docmosis Tornado v2.4 Release Notes

Oct 2017

Patch release to improve Tornado stability.
Docmosis core remains at version 4.1.0

Tornado Changes

#	Change
1	Added email field to "status" test bed of Tornado console to allow emails to be sent as part of testing..

Web Service API Changes

#	Change
1	None.

New Core Engine Features

#	Change
1	None.

Bug Fixes / Technical Changes

| #	Change
1	Improved email system configuration, connection and dispatch

Docmosis Tornado v2.4 Release Notes

Aug 2016

Docmosis core updated from version 4.0.4 to 4.1.0

Tornado Changes

#	Change
1	Barcodes are now available in Tornado: - Code 128 - Code 39 - ITF 14 Adding a barcode is easy and quite configurable. Please see the template guide.
2	Docx Output option added to the GUI.
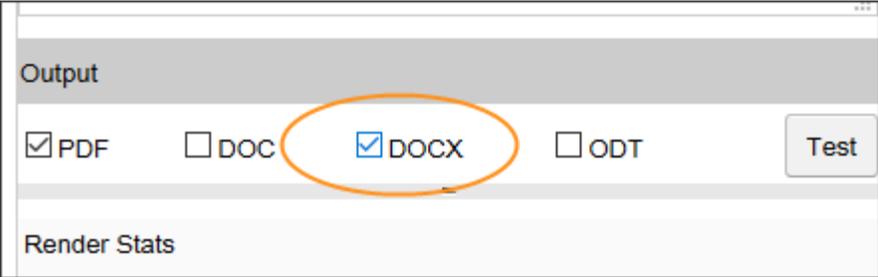
3	Images can be injected from URLS in data. Docmosis will fetch the image and stream into the document. This feature is also security controlled so that only white-listed URLs can be used to fetch images.
	
	JSON data example:
	
	`"image1":"[imageUrl:http://xmpl.co/images/test.png]"`
	
	Whitelisting eg:
	
	`docmosis.external.resources.whitelist=http://xmpl.co/images;http://e2.com/other`

Web Service API Changes

#	Change
1	None.

New Core Engine **Features**

#	Change
1	## Docx Enabled by Default. Docx processing is now enabled by default. This requires LibreOffice which has DOCX support, or the ODFConverter to be used if using Open Office.
2	## TIFF Image Support. TIFF images are now supported in the templates and data streams.
3	## New toAlpha(), toAlpha2(), toRoman() Functions. New number formatting functions toAlpha, toAlpha2, toRoman have been added to allow Docmosis to support numbering itself. This means that Docmosis can provide numbering in different formats based on the current index: ``` <<rs_items>> <<{toAlpha($itemnum)}>>). This is an item <<es_>> ``` Would produce output like: ``` a). This is an item b). This is an item c). This is an item d). This is an item … ``` `toAlpha()` maps numbers to a, b, c…, y, z, aa, bb, cc, dd etc. For example: ``` toAlpha(1) => a toAlpha(26) => z toAlpha(27) => aa toAlpha(28) => bb ``` `toAlpha2()` maps numbers to a, b, c…, y, z, aa, ab, ac, ad etc. For example: ``` toAlpha2(1) => a toAlpha2(26) => z toAlpha2(27) => aa toAlpha2(28) => ab ``` which is the same as `toAlpha()` except when hitting double letters. `toRoman()` maps numbers to Roman Numerals. For example: ``` toRoman(1) => i toRoman(26) => ii toRoman(27) => xxvii toRoman(28) => xxviii ```

Bug Fixes / Technical Changes

#	Change
1	Fixed bug in `FileUtilities` where unzipping a zip file with sub folders was failing because the sub-folders were not being created.
2	Improved field detection where plain text fields near the bottom of a page could be left unrecognized.
3	Fixed bug where injecting html that was simply a table would leave the "`repairme`" text behind.
4	Fixed issue where table borders and other style information was lost for some simple tables where `<<rr_>>` or `<<ref:>>` fields existed, but no tables in the document had lookup fields.
5	Reduced diagnostics logging for the `ExpressionFunctionAdapter` to make sure that when a processing error occurs, the message doesn't contain the package and class of the exception. This makes the message more meaningful to the user.
6	Updated evaluator to not spit out same message about overridden functions every construction.
7	Fixed NPE in table analysis where conditional column was spanned. This could show up as an error with message "null".

Docmosis Tornado v2.3 Release Notes

Apr 2016

Docmosis core updated from version 4.0.3 to 4.0.4

Tornado Changes

#	Change
1	GUI changes for Tornado console: - Added "ping" hyperlink to show how to access PING service - Added "Status" hyperlink to show how to access STATUS service - Added "WADL" hyperlink to allow WADL to be accessed easily See details for the API calls below

Web Service API Changes

#	Change
1	New "ping"web service endpoint added: eg. http://localhost:8080/rs/ping This service can be called to determine whether the Tornado web server is running. It is a simple minimal check. The service returns: http: 200 on success.

2	New "status"web service endpoint added:
	  eg. http://localhost:8080/rs/status
	This service can be called to determine more detailed Tornado operation including whether it is ready to render documents and current utilisation. The service returns the following JSON response summarizing operation:
	`{`
	`  "detail": {`
	`    "converterCountInUse":"0",`
	`    "converterCountOffline":"0",`
	`    "converterCountOnline":"4",`
	`    "converterCountTotal":"4",`
	`    "uptimeSeconds":"20"`
	`  },`
	`  "message":"ready",`
	`  "ready":"true"`
	`}`

New Core Engine Features

#	Change
1	Field parsing has been improved to allow the period character (".") to appear in template string literals. Eg. `ab.cd`.
2	Date Rendering has been improved such that if an input date string includes a timezone, that same timezone will be applied to the output value. Previous behavior is that the local timezone would be applied when outputting the resulting date/time.

#	Change
3	Improved startup processing to work when running as a Windows Service.
4.	Fixed issue where table borders could be rendered incorrectly when using repeating rows with a "step directive" eg. `<<rr_xx:stepN>>`.

Bug Fixes / Technical Changes

#	Change
1	Fixed field parsing to identify a special case of plain-text-field that could cause the field not to be recognized by Docmosis and ignored.

Docmosis Tornado v2.2 Release Notes

Dec 2015

Docmosis core updated from version 4.0.2 to 4.0.3

Tornado Changes

#	Change
1	GUI improvements for Tornado console: - Test button is always visible - Its now easy to resize the display to make the data area larger to work with

2	Rendering updated to always use "localhost" when invoking the render. This fixes the issue where the server is unable to reach back to itself via the public hostname / address.
	Two new configuration settings are available to override this configuration:
	`docmosis.tornado.render.useHostName=localhost`
	`docmosis.tornado.render.useUrl=http://my.host.com:8080/`

3	Added command line settings for control over logging: - `log.level=debug|info|error` to specify the level of logging to the console and log files - `log4j.config.file=path` to set the log4j logging configuration file - `java.util.logging.config.file=path` to set Java Util logging configuration file

Web Service API Changes

#	Change
1	<no changes>

New Core Engine Features

#	Change
	<no changes>

Bug Fixes / Technical Changes

#	Change
1	XML sample data creation no longer creates invalid duplicate keys.
2	Bug fix where template processing errors for conditional sections inside bullet/numbered lists where the conditional section end is in a different paragraph.

Docmosis Tornado v2.1 Release Notes

Nov 2015

Docmosis core updated from version 4.0.1 to 4.0.2

Tornado *Changes*

#	Change
1	The release has been updated to a ZIP file containing various materials including the program, documentation and license information.
2	GUI improvements for Tornado console: - Feedback when render is executing - When admin password is set, session expiration now redirects to login page - XML Sample Data creation no longer produces duplicate "keys"
3	Added command line settings for control over logging: - `log.level=debug|info|error` to specify the level of logging to the console and log files - `log4j.config.file=path` to set the log4j logging configuration file - `java.util.logging.config.file=path` to set Java Util logging configuration file

Web Service API *Changes*

#	Change
1	<no changes>

New Core Engine *Features*

#	Change
	<no changes>

Bug Fixes / Technical *Changes*

#	Change
1	Fixed issue where session expiry would give unhelpful error if admin password is configured. Now automatically redirects to login.
2	Much logging has been moved to debug level so the default info level logging is more succinct.
3	Http caching disabled for Tornado console to aid with browser cache related problems.
4	Fixed bug stopping the restart button from being enabled.
5	Render now always uses `localhost` to execute the render rather than non-deterministically choosing a local address to use.

Docmosis Tornado v2.0 Release Notes

Oct 2015

Docmosis core updated from version 3.3.0 to 4.0.1.

Tornado Changes

#	Change
1	Added the ability to generate XML sample data based on the template (new button on the GUI)
2	Render requests are now done in DEV mode. This means errors will be injected into the document to help with locating problems. The render REST API supports the choice of dev/prod mode.
	There is a new REST API endpoint "/convert" which can be used to convert (not populate) files between formats including spreadsheets and drawings.
	GUI Updates for the web console including: - Text wrapping - Display 200 templates in browser tree - New scroll bars - XML dummy data button

Web Service API Changes

#	Change
1	render endpoint allows `storeto` to be empty as well as not specified
2	web services - updated render end point to set header `X-Docmosis-PagesRendered` when successful.
3	web services - updated `getTemplateStructure` to correctly detect `TemplateNotFoundException` and return as a 400 failure with the correct message
	web services - updated render to have new parameter "`ignoreUnknownParams`" to allow systems to pass unreasonable params. This is because one customer has a system they call to issue the request to Docmosis. It adds it's own params and the customer can't avoid that.
	web services - added PING endpoint ("`/ping`")
	web services - allowed "+" character in output file names
	New endpoint: `convert` for converting documents without requiring data or templates. This endpoint is considered experimental. Contact Docmosis support if you would like access to this service.
	Image data can reference images by fetching from URLs.
	Updated render service where response message was being constructed in JSON rather than XML or choosing between XML and JSON correctly.

#	Change

If / Else / Else-If Support in Conditional Sections

Simple "else" looks like this:
```
<<cs_true>>
true
<<else>>
false
<<es_>>
```

"else-if" is written like this:
```
<<cs_isPerson>>
 I have a person
<<else_isPlant>>
 I have a plant
<<else>>
 I have something I didn't expect
<<es_isPerson>>
```

Conditional Sections Support the new Expression Syntax (described below)
Eg:
```
<<cs_{val < 10.0}>>
Low value = <<val>>
<<else_{val > 100.0}>>
High value = <<val>>
<<else>>
Nominal value = <<val>>
<<es_>>
```

New Expression Engine

A new expression engine has been added that improves computational capabilities of templates. Operators and functions can be applied to String and numeric data. The following lines summarize the way this affects the templates.

Expressions in Docmosis templates are still delimited by the braces ("{" and "}") characters. Expressions can now include:

- Precedence using the brackets "(" and ")" characters
- Mathematical expressions
- Mathematical and String functions
- Boolean logic

Direct Expression Evaluation

Fields which are expressions can be used to insert data into documents. For example:
```
<<{1 + 2 + 3}>>
<<{round(val/100,2)}>>%
<<{titleCase(firstName + ' ' + lastName)}>>
```

#	Change

Assignment of Expressions to Variables

Variables can now be assigned the results of expressions: << $m={expr} >>

```
Eg. <<$m={round(1+ceil(2*3.5))}>>
    round(1+ceil(2*3.5)) = <<$m>>
```

Boolean Logic

```
<<cs_{val1 || (val2 && val3)}>>
val1 is true or both val2 and val3 are true
<<es_>>
```

Maths functions

Typical math functions are supported.
```
    Eg. <<{max(4.522, 4.5)}>>
```

The round() function has been extended to support an optional precision:
```
    <<{round(1.236)}>>
    <<{round(1.236, 2)}>>
```
The precision also pads:
```
    <<{round(1.2,5)}>>
```

String Functions

charAt, compareTo, compareToIgnoreCase, concat, endsWith, equals, equalsIgnoreCase, indexOf, lastIndexOf, length, replace (character replacement), startsWith, substring, toLowerCase, toUpperCase, Trim, map, titleCase, split,

eg:
```
    <<{equals('a','b')}>>
    <<{indexOf('abc','b')}>>
    <<{startsWith('this is', 'this')}>>
    <<{titleCase('joe blogs')}>>
```

Formatting Functions

numFormat and dateFormat functions have been created to perform numeric and date formatting functions. These functions are based on the similarly named FieldRenderers that already existed in Docomsis.
```
    numFormat(<value>, <format>[, <locale> ])
    <<{numFormat(value1, '###,###.00')}>>

    dateFormat(<value>[, <output format>[, <input format> ]])
    <<{dateFormat(value1, 'dd/MM/yy', 'dd-MMM-yyyy')}>>
```

Operators

The well known operators are supported:
```
    ( ) + - * / % + - = == != < <= > >= && || !
```

XML population has been updated to allow "\r" to result in paragraph insertion (in addition to "\r\n" and "\n"). This helps XML data processing where the xml contains this type of new lines.

#	Change
	Logging is by default quieter now with more logging information moved to DEBUG/FINE level.
	Hyperlink processing has been updated to be able to use variables and variables set from expressions. The hyperlinks are now expected to be of the format `<<link:xxx>>` but the `<<link_xxx>>` format is still supported.
	Sub-templates can now set template-variables that are visible to the "master" template and subsequently processed templates.

Bug Fixes / Technical Changes

#	Change
1	Fixed issue where no data being given by client results in an error
2	New license key format
3	Fixed bug with JSON image dummy data provision which didn't work for ODT templates
	JSON dummy data structure processor improved to handle case where image and text field have same key.
	Updated Tomcat to 7.0.65 (from 7.0.27) which fixes runtime JSP compilation errors under Java 8
	Fixed issue where spaces around data input via web console would stop data being recognized as JSON format.
1	Improved hyperlink insertion to deal with being given blank data. This was corrupting DOC output.
	Updated field parsing to handle more cases where set-variable fields occurring in redundant paragraphs (blank lines) was causing parsing errors.
	Improved injection of errors into the document (in "dev" mode): - Errors not being visible in the document - stopping template errors highlighting in red
	Improved JSON data provision for anonymous arrays (arrays that have a list of values without keys). Internally, the new key "" is used for items that are anonymous which fits current data provision perfectly - and is semantically a good fit.
	Improvide population of <<rs_$this>> and <<rs_$current>> to work with JSON anonymous arrays of arrays.
	Updated analysis to detect when overlapping sections are caused by an over-zealous image bookmark and raise a better error message. This occurs when the bookmark captures more than just the image so other content could be accidentally removed from the result.
	Updated field processing in numbered and bullet lists. Removes the default assumption that a bullet list with a field in it is intended for repetition and makes it much more "normal".
	Increased the default window size for xml processing for templates with large/complex paragraphs particularly in docx format.
	Fixed issue where hidden ("_GoBack") bookmarks created by new versions of word (Mac) were causing blank paragraphs to be left behind.
	Updated error handling for when a start or end tag is in a list (but not the matching tag) - causing the related other tag to be lost/remote

#	Change
	Fixed issue where looping over including sub-templates more than 2 levels deep would incorrectly determine a cycle in template referencing and raise an error.
	Updated number renderer to be able to take a second parameter which identifies the locale to use when parsing/formatting the date.