Docmosis v2.2.2 Release Notes

January 2011

New Features

#	Change
1	**New Break Fields** Docmosis now provides 4 fields as a neater way to insert page and column breaks. The first forms will produce a page and column break (for multi-column documents) respectively: *«pageBreak»* *«columnBreak»* The second forms are for use within repeating sections and will insert the corresponding break until the last iteration of the section: *«pageBreakNotLast»* *«columnBreakNotLast»* The new fields can be used instead of or in conjunction with actual page and column breaks in your template. Please see example4 in the download bundle which shows an example use of *«pageBreakNotLast»*.
2	**Image Lookup Extended** Image lookup has been extended to match the facilities of textual data including nesting (eg my.images.image1) and variables ($myImage)
3	**Corrupt Templates are Detected Earlier** Previously if badly corrupted documents were used as templates (for example empty files or incorrect format) then Docmosis would take minutes to identify this. Now such templates are recognised sub-second.
4	**Full-Justification Support for Multi-Line Data** Carriage returns/end-of-line sequences in data now result in the end of a paragraph in the resulting document. This is particularly useful when using full-justified paragraph formats which would previously display overly-spaced lines.

API Changes

The following API changes should be noted

Class / Interface	Change
DataProviderBuilder	New addJavaObject(Object, String, boolean) method has been added to allow "forgiving" mode to be specified during Java reflection for data.
DataProviderBuilder	addJavaObject(Object) has been deprecated since it can mask other data and such behaviour is not obvious.

Bug Fixes / Technical Enhancements

#	**Change**
1	Improved processing and reporting of errors in template tables where rows marked as heading rows are used.
2	Corrected logging of class name where Reflection fails to find any method related to the template field.
3	Fixed issue where images were not substituting in headers under some conditions

Docmosis v2.2.1 Release Notes

May 2010

New Features

None

API Changes

The following API changes should be noted

Class / Interface	Change
DataProviderBuilder.addJavaObject(Object)	This method has been documented as dangerous due to its ability to hide other data. addJavaObject(Object,String) is the safe alternative.

Bug Fixes / Technical Enhancements

#	Change
1	Reflection now populates correctly from non-List Collections such as Set (eg TreeSet) and Queue.

Docmosis v2.2.0 Release Notes

April 2010

New Features

#	Change
1	**Template Merging** Docmosis now allows templates to be referenced by other templates and the templates will be merged at processing time. This allows templates to have common content to be separated out into shared templates. For example, if documents have a standard header layout, they can reference a common header template using one of the two new Docmosis fields. This might look like: *«ref:header.doc»* or *«refLookup:headerTemplate»* where the first field would pull in the template called *header.doc* into the current template and the second field would look up the key "*headerTemplate*" in the data provider to get the name of the template to pull in. See the Docmosis Template Guide for more information and *example5* in the download bundle for an example.
2	**New Render Methods** The DocumentProcess class has new overloaded render methods to assist in simplifying document generation. Most notably is the presence of a new Boolean parameter to override the default behaviour of cleaning up the DataProvider after the render. Reusing the data can be helpful if making separate render calls to produce separate documents rather than a single call to produce a zip archive.
3	**Reduced IO** Disk and Network IO has been reduced in the case where the Converters are on the same host as the core engine.
4	**Improved Error and Diagnostic Handling** a) 32/64 bit incompatibilities are detected and suggestions are reported. This is helpful when using a 64 bit Java and there is no 64 bit OpenOffice (such as on Windows and MacOSX). b) Handshaking between the Docmosis core and the converters will report versioning issues, status and environmental differences.
5	**Improved docmosis.properties** The example *docmosis.properties* file has been re-organised to show the critical properties first, better documentation and show some of the other important properties.

API Changes

The following API changes should be noted

Class / Interface	Change
DocumentProcessor	New renderDoc() methods allowing easier access to re-using the same DataProvider between calls. Previously

	only one method allowed this to be specified (within the ConversionInstruction)

Bug Fixes / Technical Enhancements

#	**Change**
1	Repeating Table Rows now allow for conditional rows and hence keep border and background styling features
2	Improvements to rendering errors into the resulting document in corner cases where error was not reported
3	Improvements to ODT (OpenOffice Writer) template processing for end of section detection.
4	Improvements to JavaDoc information

Docmosis v2.1.1 Release Notes

February 2010

Bug Fixes / Technical Enhancements

#	**Change**
1	OpenOffice 3.2 support OpenOffice has a great new release which has no serious bugs as far as Docmosis is concerned. The previous two production releases (3.0.1 and 3.1.0) had bugs that were not ideal for typical Docmosis use. OpenOffice 3.2 changes a few things under the hood so Docmosis had to upgrade to match.
2	New options to control the way Docmosis loads OpenOffice jars and native libraries. This allows Docmosis to launch the Converters itself rather ("embedded converters") rather than this requiring a separate script (eg runConverter.sh). This means setup is simpler for smaller systems. The launching works from within more Web Application Servers than before, such as JBoss5, Glassfish etc. Note: using "embedded converters" implies you are running everything on the same machine as opposed to a load-distribution configuration. To use embedded converters, update see the information in the example *converterPoolConfig.xml* file. The primary new property allowing control over the library loading is: *docmosis.openoffice.useCustomLoader=true|false* This property defaults to *false* but if you have linkage errors, particularly when using embedded converters, you can try setting this to *true* to load the libraries in a different fashion.
3	The property *template.store.location* can now be left blank, in which case Docmosis will create a temporary area to work with for the template cache.

Docmosis v2.1.0 Release Notes

November 2009

New Features

#	Change
1	**Hyperlink Insertion** Docmosis can now insert active hyperlinks into documents. A link placeholder is inserted into the template using a name with a "link_" prefix. For example a field *«link_myWebSpace»* will look up the data provider using the key "myWebSpace" and render the result as a hyperlink. Further, the data provided can be delimited using a pipe symbol ("|") to name the link differently from the address. For example, given the field above, a data value of "http://www.mywebspace.com/bluk" would be rendered as a hyperlink and display the text "http://www.mywebspace.com/bluk" in the document. If the data value was "mywebspace|http://www.mywebspace.com/bluk", the link would be rendered into the document displaying the text "mywebspace". See *example1* in the download bundle to see it in action.

API Changes

The following API changes should be noted

Class / Interface	Change
DataProviderBuilder	addFile() methods now allow character encoding to be specified.
StoreHelper	New storeTemplate() method that can take an InputStream as the source of the template rather than just file-based templates.
DropStoreHelper	New process methods to allow Zip and Jar files of templates to be processed directly, or from URLs to Resources from a class loader.

Bug Fixes / Technical Enhancements

#	Change
1	Expressions Enhanced - general improvements i. default boolean true test cs_{isFriend()} was previously invalid, but now evaluates isFriend() as a boolean as expected. ii. null tests supported cs_{getAlpha()=null} iii. size() capability broadened now also applies to any non-reflective data sources iv. isEmpty() capability added cs_{getFriends().isEmpty()}

2	Fixed issue where rs_, cs_ and es_ tags at the first line of a template page could result in extra blank lines in output documents.
3	Added toString() to TemplateAnalysis implementation to allow dumping of analysis information.
4	Fixed issue with nested anonymous er_ tags failing template registration.
5	Fixed issue with second and subsequent Tables Of Content/Tables of Figures etc not being updated correctly.
6	Improved handling of java.sql.Time data type.
7	Subtle fixes to template caching and warnings for large templates
8	Fixes to DataProviderBuilder.addJavaObject(name, Object) so that it works with Collection/Array data types and allows them to be referenced directly by the name given.
9	The connection to OpenOffice will complain with a specific message about unsupported Java version if the OpenOffice API is not compatible with the version of Java in use.
10	Improved population of non-ascii characters to support multiple languages and symbology.