

The Apache OpenOffice 3.4 features can be split into two stages: changes already made in the OpenOffice.org beta, and additional changes made at Apache since incubation started. During both stages many small bugs were fixed and we can't honestly summarize the huge help provided by our development community.

These features were already part of the OpenOffice.org 3.4 beta. The 3.4 beta was built from the OOO340 code line which was cloned from DEV300m106. This code line is the base for the Apache OpenOffice code.

Startup is faster since the program now has enough initial knowledge about its components so that it doesn't have to start each of them.

The ODF 1.2 specification allows optional use of W3C-specified algorithms in the encryption (password-protection) of ODF 1.2 documents.

For Apache OpenOffice 3.4, the default "Save with Password" continues to use SHA1 digests and Blowfish encryption, the one combination that is supported for interchange of encrypted documents across ODF 1.0/1.1 and ODF 1.2 supporting application.

In addition, documents encrypted using AES256 are now recognized and decrypted when the correct password is provided.

It is also possible to select AES256 for encrypting "Save with Password" documents by setting both of these configuration options:

- Set configuration option org.openoffice.Office.Common/Save/ODF/UseSHA1InODF12 to "false".
- Set configuration option org.openoffice.Office.Common/Save/ODF/UseBlowfishInODF12 to "false".

A macro to set the options on or off is available at <http://wiki.services.openoffice.org/wiki/User:TJFrazier/Encryption>.

The configuration options have no effect when saving in ODF 1.1/ODF 1.0 format.

See <http://docs.oasis-open.org/office/v1.2/csprd03/OpenDocument-v1.2-csprd03-part3.html> for precise details of the ODF 1.2 provisions.

The TIME spreadsheet function returns values between 00:00:00 and 23:59:59.999... inclusive and wraps values equal to or greater than 24:00:00.

In the Spreadsheet GUI, the term "DataPilot" has been replaced with the term "Pivot Table".

PivotTable (formerly known as DataPilot) is no longer limited in the number of fields supported.

Previously, in Spreadsheet CSV export, all content from text cells was quoted with the selected text delimiter. Now this is optional and happens only if the new check box "Quote all text cells" is selected. By default, the text delimiter is added only if the cell content contains the field delimiter, text delimiter or multiple lines. When a CSV file is loaded and saved again without editing the filter settings, the import option "Quoted field as text" is used for the "Quote all text cells" option. The default for "Quoted field as text" is now "off" (but if it's changed in the import dialog, that is still saved as new default in the configuration).

Better interoperability with other applications supporting the import of CSV (Comma Separated Values) files as the style for exporting strings is configurable now.

In the Pivot Table layout dialog, it is now possible to add more than 8 fields to the "Row Fields", "Column Fields", and "Data Fields" areas, and more than 10 fields to the "Page Fields" area.

###

The MaxRowScan setting allows the user to define how many rows should be scanned to find the correct type of a column. In the past only the first row was scanned.

Math: new option to save only used symbols for each formula

In order to save on document file size (namely in Writer documents with many formulas) there is now a new option to save only those symbols that are used in a formula. By default this option will be active. Geek info: However there is a small drawback: Previously, if you worked with different office installations, documents with formulas could be used to implicitly import **ALL** user-defined symbols in Math from one office installation to the next, even if none of them got used in a formula. Since now only the actually used symbols get saved, only those can be implicitly imported into a different office installation. To get the old behavior, simply turn off this new option.

Formula: Automatic baseline for Math objects

There is a new check-box named 'Math baseline alignment' available in the 'Formatting Aids' tab of the Writer and WriterWeb option pages. When this option gets checked then **ALL** Math OLE objects already existing in the Writer document that are anchored 'as character' (and only those with this anchor type) will be automatically aligned to match the baseline of the formula with the one of the surrounding text. Also new Math objects inserted later on will be automatically aligned as well. As long as this option is active, manual movement of Math objects that are anchored 'as character' is not possible, and thus vertical alignment in the 'Object' context dialog is now disabled as well. For new documents this option is set by default. When loading old documents it is not set. In any case, once a document gets saved again, the current value of this option will be saved along with the document. This automatic baseline alignment was implemented by Michal Spisiak as part of this year's Google Summer of Code; the mentor for this was Eric Bachard. Thanks to both of them!

Math symbol catalog now supports UTF-32 characters

Draw/Impress default enhancements:

- Better defaults for Impress & Draw snap.
- Changed default for "copy when moving".
- Better defaults for grid settings in Impress & Draw
- Better defaults for bullet spacing and indentation
- Animate outline shapes 'By 1st level paragraphs' as default
- New default colors of drawing objects
- Setting default shadow distance for shapes to 0.2cm
- Enhancement to the "mouse as pen" feature in impress slideshow
- Default full width for text in shapes
- Change outline default bullet symbol order from 'Oo-' in 'O-O-O'
- Shortcut for inserting comments changed

Better UI Defaults for Draw and Impress

See for details:

[Better Bullet UI](#)

[Better Grid UI](#)

Draw/Impress: Setting pixel resolution when exporting graphics

PDF File Size Reduction

PDFs containing monochrome bitmaps are smaller now

Password entry in PDF export dialog changed

During PDF export one can enter two passwords ("open" and "permission" password). These can now be entered in one pass instead of two separate dialogs.

Base: Mozilla Address Book support removed, in favor of Seamonkey

Since the Mozilla product has long been superseded by the SeaMonkey suite, and the profile formats of both changed incompatibly, the Mozilla Address Book support in OpenOffice.org has been removed, in favor of support for the SeaMonkey address books.

Base: List box controls: "Bound field" is allowed to be 0

The "Bound field" property of a list box control, specifying which column of the list boxes result set should be used for data exchange, can have the value "0" now.

Base: object trees: selecting entries by typing their name

The tree lists in Base's main application window, now support selecting entries by typing their name.

Base: Scan rows for correct type information for Text databases

The MaxRowScan setting allows to define how many rows should be scanned to find the correct type of a column. In the past only the first row was scanned.

[Base new features detailed description and Screenshots](#)

Stylist / Navigator / Options Dialog: selecting entries by typing their name

The tree lists in the application's Navigator, Stylist ("Styles and Formattings"), and in the options dialog, now support selecting entries by typing their name.

Chart: Resizeable Legend

The legend within charts can be resized now.

Chart: Date Axis

Date axis are available now for category charts.

Time Axis supported in Chart



Improved Printing on Unix

Printing via PDF (if the system supports it) allows object transparency to be handled directly by the printer subsystem

Print file format changes on Linux systems

Printing on Unix systems can now produce PDF files instead of PostScript.

New Gstreamer multimedia framework on Linux systems

Gstreamer is available now to use to playback multimedia content in OpenOffice documents (e.g. audio and video in Impress presentations). Note that Gstreamer (and the appropriate gst-plugins) need to be installed in system.

Enhanced RTF export

Now it supports nested tables, more character properties, sections, graphics for Wordpad, bookmarks, fields, drawings and OLE objects.

Remove binfilter export filters

Further development/bug fixes for DOCX import filter

OOo/Mac running headless as daemon

Various BugFixes

New Features for Apache OpenOffice 3.4

These features are the result of the work at Apache on OpenOffice since the OpenOffice.org 3.4 beta:

License Change

The License has changed to our simpler, non-restrictive, [Apache License 2](#). The Apache License will make adoption, distribution and modification of the software easier for all users and developers. Of course the software remains free, and the source code with its updates will continue to be available through the Apache Software Foundation servers.

Fonts and other Misc. Updates.

In accordance to the new license all the software components have been updated to use licenses that are in line with ASF guidelines. No critical functionality was lost, and we did update many libraries that will work better on most situations. Many internal libraries and headers were updated with modern versions that provide the same or superior functionality. On selected platforms (Mac OS, Linux and others) the "Liberation" fonts were replaced with the [improved](#) Google Chrome fonts.

We strongly recommend that all developers and distributors update to this release to avoid risks resulting from inconsistencies (IP and security issues) found in previous versions of the OpenOffice.org suite.

Calc: Faster LP solver.

The previous Linear Programming solver has been replaced with the faster and better designed CoinMP from the Computational Infrastructure for Operations Research ([COIN-OR](#)) project.

Apache Tomcat updated.

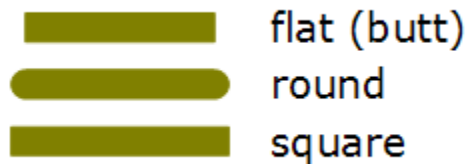
The Apache Tomcat servlet was updated from the older version 5.0.30 to version [5.5.35](#) which brings in years of performance and security improvements without changing the API. We plan to continue updating components without causing any regression on the current code.

Support for Line Cap Property

Now you can add a cap to the end of a line. Such caps are not only known in the ODF1 .2 standard but in HTML5 and SVG too. Also other Office Suites provide caps in different styles to be added to thick lines.

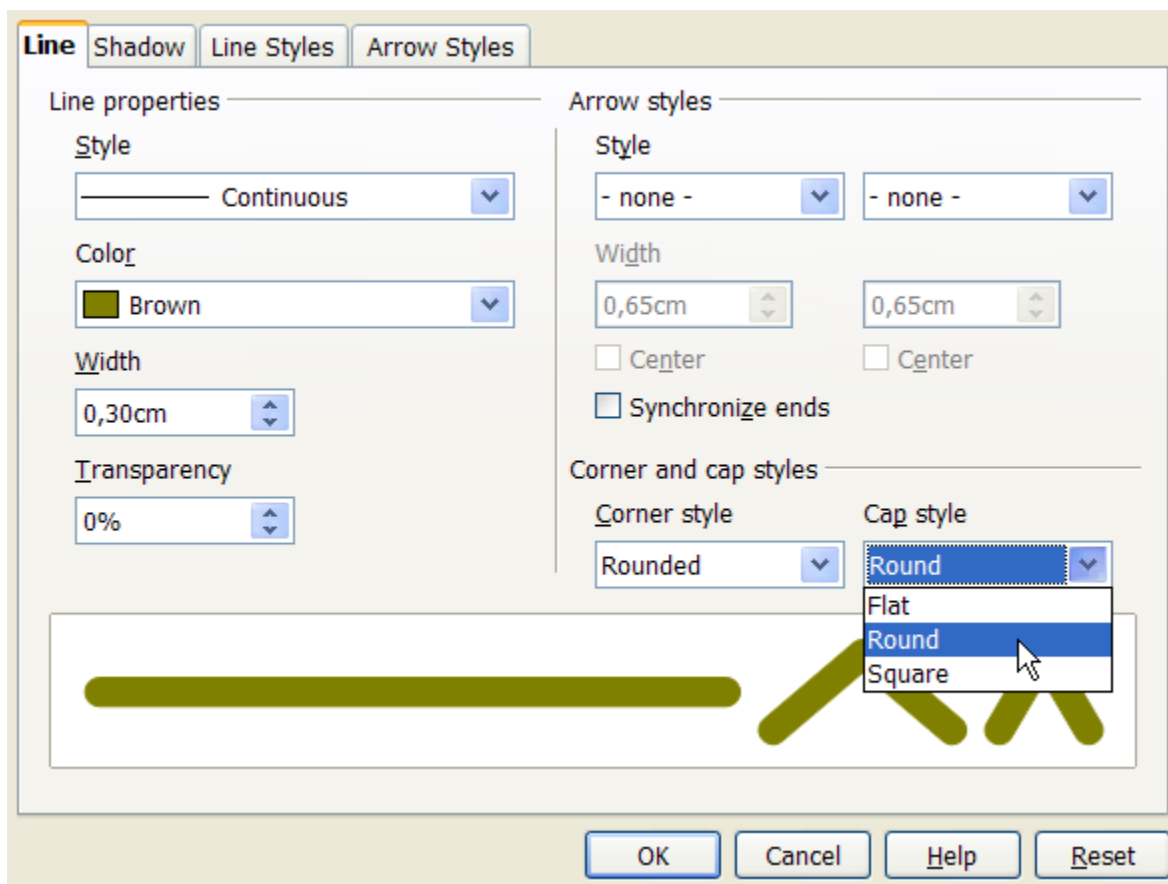
Three styles exist:

- without a cap, called 'Butt' in programming and 'Flat' in UI (as in MS Office)
 - with a round cap, called 'Round'
 - with a rectangle cap, called 'Square'
- The property value 'Flat' corresponds to the old behavior and is now the default.



The caps are added to the lines, so that the total length of the lines increases with two-times a half line width.

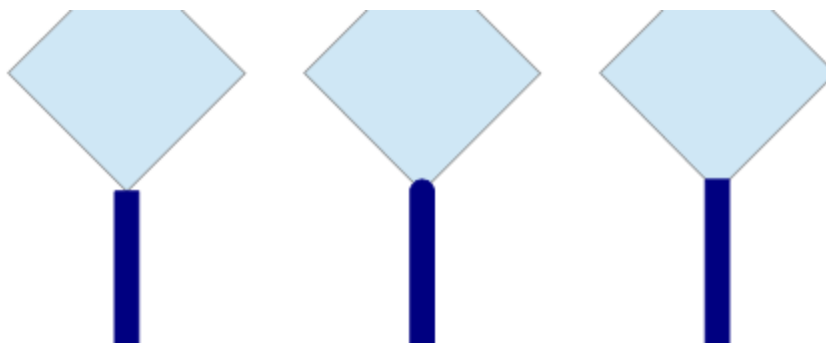
Select the value from a drop-down-list in the line property dialog, just beside the settings for corner style. The new property is only available in contexts, where the corner style is active too.



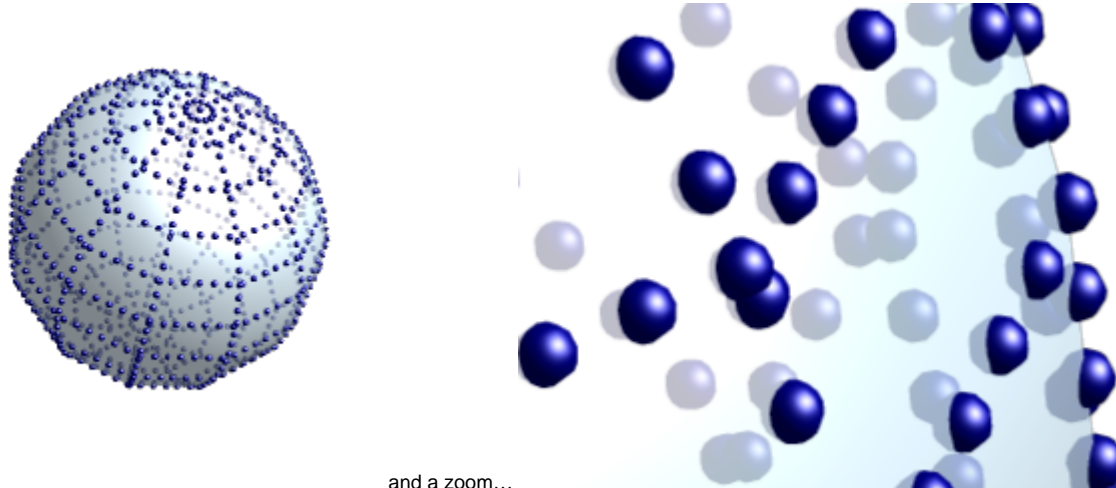
If a line is dashed, the single dashes get caps too. Hereby a dot is treated as dash. You can style not only pure lines and curves, but the border of graphic objects as well.



The next example shows the new property applied to a connector. Left side without cap, in the middle with 'round' cap, and on the right side with 'square' cap.



The caps are available for 3D-objects too, when you turn on lines and make them thick. The example shows lines, which are styled to look "dotted".



and a zoom...

Linecaps defined in svg-graphics are supported, so that those graphics look the same as they do in modern browsers.

Support for shear transformations for GraphicObjects ([Blog entry](#))

GraphicObjects which get created when inserting graphics in Draw/Impress and Spreadsheet support now not only rotation but also shear, slant and distort. The visualization during interactions was improved, also the break for vector-based GraphicObjects to draw objects was improved. Writer has its own GraphicObjects; the ones from the other applications can be copied to it as workaround.

Support for attributes and transformations for OLEObjects ([Blog entry](#))

OLEObjects (OLE is for Object Linking and Embedding) in Draw/Impress and Spreadsheet support now all draw attributes and geometrical transformations. They can have line style, fill style, shadow and text. They support all transformations, e.g. rotations and shear. This is handy for e.g. having a mathematical formula shown rotated by 90 degrees or adding a border to a chart. Break to draw objects is also enhanced.

Enhanced crop support for GraphicalObjects

Crop for GraphicObjects now works correctly together with horizontal and/or vertical mirroring in all applications.

Support for Scalable Vector Graphics (SVG) ([Blog entry](#))

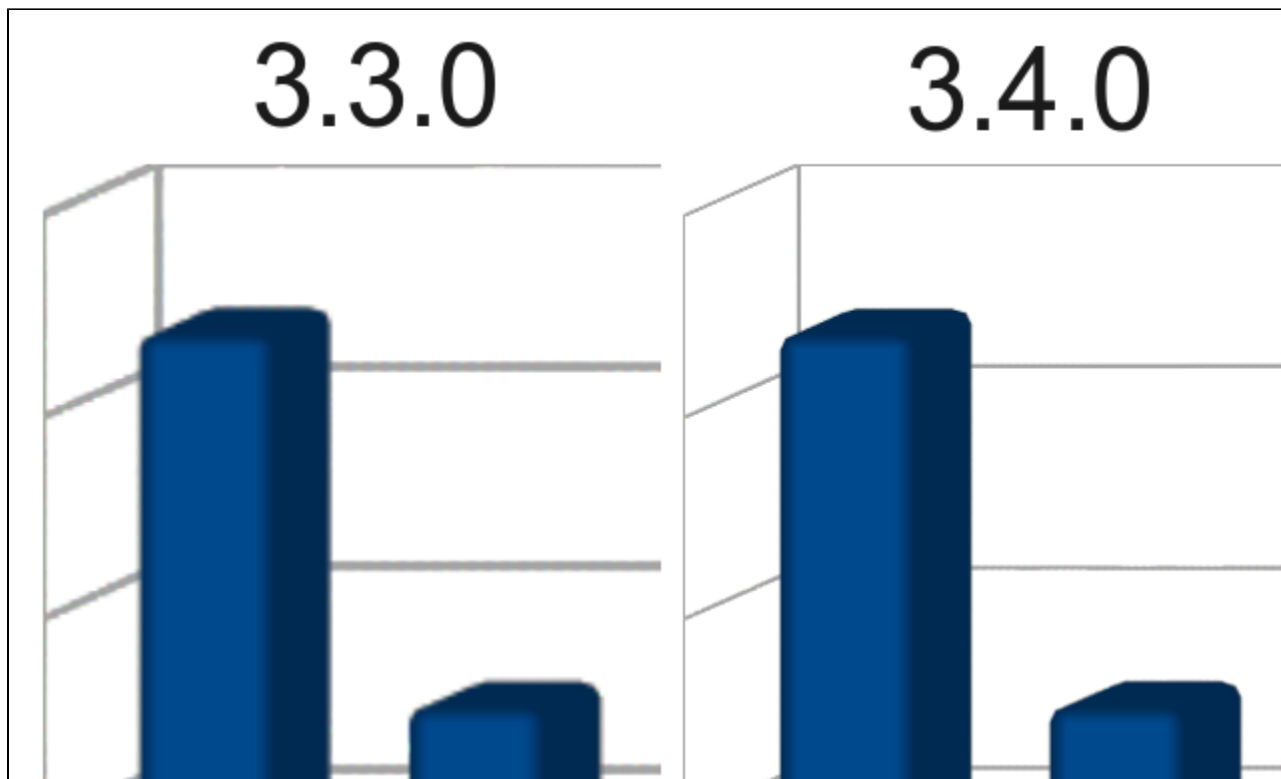
SVG is now supported as content for GraphicObjects in all applications. The new implemented generic SVG interpreter supports SVG format 1.1. The geometric content is internally processed as vector data in all usages, e.g. PDF export and printing, which guarantees good visualization quality. SVG graphics can be broken to draw objects and be processed further. A blog entry can be found [here](#).

Support for Multimage in ODF

For SVG support it was necessary to allow multiple image representations for one GraphicObject to be present in the ODF file format, e.g. a pixel graphic and the original SVG is written in case of a GraphicObject with SVG content. This allows it to stay compatible with other and older ODF supporting applications. The number of images for one GraphicObject is not limited: applications using this may choose the image with the format which supports its purpose best, e.g. prefer to use a pixel graphic for ODF viewers. OpenOffice 3.4 uses a weighting function which prefers pixel images with transparency over such without and vector formats over pixel formats.

Enhanced chart visualization

The visualization quality of charts has been enhanced by using a new mechanism for more direct visualization. This increases speed, reduces memory usage and enhances chart visualization in all visualizations, including PDF export and printing.



Writer: Asian Language Word Count

The word count of Asian languages is now correct. Previous releases were counting terms instead of words when counting words of Asian language text. Notice that file word count will not be automatically updated for existing text documents. You need to make a change to the document in order to force an update of the file word count.

New regular expressions (regex) engine

The existing regular expression engine has been replaced by the ICU engine. This solves several bugs in "Find and Replace" with the old implementation and speeds up the search.

The new engine also offers better standards compliance especially regarding Unicode, which also means that some non-standard syntax extensions like `\<` and `\>` for word boundary matching are now deprecated. For the convenience of a smooth upgrade experience they get emulated by the `\b` operator though.

See [ICU regex syntax](#) for expressions supported in the new regular expression engine. It is recommended to stay on the common ground though and [regex flavors](#) gives an overview where this common ground is. The modified syntax applies to macros too: regular expressions in macros that were relying on the deprecated syntax should be converted to use widely supported regex expressions.

Users of Japanese scripts should be aware that several non-trivial transliterations could behave slightly differently, especially when transliteration rules like "ProlongedSoundMark", "IterationMark", "Ignore-Width", "BaFa", "SeZe", "HyuByu", "landEfollowedByYa" or "KiKuFollowedBySa" might be involved.

New Color Picker dialog

The old color picker which is used in the office code for over a decade has been replaced by a new, more intuitive one which offers a wide variety of color selection possibilities. This feature was developed by Christian Lippka; kudos to him for offering it under Apache License. A picture says more than words.

Old	New
-----	-----

