Development Snapshot Builds

Snapshots builds are announced periodically on the Apache OpenOffice dev mailing list.

More frequent dev builds can be found in the Buildbot output area.

IMPORTANT: The builds linked on this page are intended to be used for **final testing** of release candidates by community volunteers. They have no release quality and should not be installed in a production environment. Ongoing DEVELOPMENT builds (committed changes but not allocated to a specific release) and SNAPSHOT builds (consideration for release candidate) can be found in the Buildbot output area. DEVELOPMENT builds can be unstable and are expected to have bugs.

Please report problems revealed by installation and testing of a snapshot as soon as possible. Found issues should be submitted in the issue tracker. Issues, along with subsequent issue-investigation and -resolution updates, are announced automatically on the ooo-issues mailing list.

The current SVN Revision Number of the repository is used to identify snapshots built from the Subversion Repository that is current as of that revision. The revision numbers are shown in the reporting of individual commits on the ooo-commits mailing list. The commits made between successive snapshots can also be determined by reviewing SVN logs, down to the revision history of individual source-code files.

NOTE: The provided *unofficial* development builds here should always be based on the same revision number to align and coordinate our early QA efforts. It is not intended to keep older builds at this time for reference.

Anchor | AOOSnapshotSource

Anchor | AOOSnapshotfullsets

Anchor | AOOSnapshotMSP

Used build flags for development snapshots

MacOS

I prepare my source tree to have a fresh clean tree without any output. I update the sources to the revision number that we have agreed on and use the following switches to build the snapshots. I use pre-built versions of dmake and epm. And I use the pre-built mozilla libs.

```
#!/bin/sh
### 4.1.2 Script:
### 4.1.2 Script: echo "### Configure"
### 4.1.2 Script: ./configure \
### 4.1.2 Script:              --with-build-version="$(date +"%Y-%m-%d %H:%M:%S (%a, %d %b %Y)")" \
### 4.1.2 Script:              --enable-verbose \
### 4.1.2 Script:              --enable-category-b \
### 4.1.2 Script:              --enable-wiki-publisher \
### 4.1.2 Script:              --enable-bundled-dictionaries \
### 4.1.2 Script:              --without-stlport \
### 4.1.2 Script:              --with-dmake-path=/Users/jsc/dev/tools/bin/dmake \
### 4.1.2 Script:              --with-epm=/Users/jsc/dev/tools/bin/epm/epm \
### 4.1.2 Script:              --with-openldap \
### 4.1.2 Script:              --with-junit=/Users/jsc/dev/tools/junit/junit-4.11.jar \
### 4.1.2 Script:              --with-packager-list=/Users/jsc/dev/svn/aoo-build-pack.lst
### 4.1.2 Script:              --with-jdk-home=/Library/Java/JavaVirtualMachines/jdk1.7.0_67.jdk/Contents/Home \
### 4.1.2 Script:              --with-ant-home=/Users/jsc/dev/tools/apache-ant-1.9.3/dist \
### 4.1.2 Script:              --with-lang="kid ast bg ca ca-XR ca-XV cs da de el en-GB en-US es eu fi fr gd gl
he hi hu it ja km ko lt nb nl pl pt pt-BR ru sk sl sr sv ta th tr vi zh-CN zh-TW"
### 4.1.2 Script:
### 4.1.2 Script:
#
# Build-script for AOO 4.1.x on OSX 10.12
#
# System Setup:
#  XCode 8.3.3 (Updated w/ https://github.com/devernay/xcodelegacy.git)
#
# Local Changes:
```

```
#   MacPorts:
#     o apache-ant (1.9.9)
#     o gnutar (1.29) (symlink gnutar to gtar)
#     o perl5 (perl5.22)
#     o p5-archive-*
#     o p5-lwp-*
#     o p5-xml-parser
#
#   Oracle JAVA 1.7 JDK
#
#   /usr/local:
#     o dmake
#         http://sourceforge.net/projects/oooextras.mirror/files/dmake-4.12.tar.bz2
#         ./configure --prefix=/usr/local ; make install
#     o epm
#         http://www.msweet.org/files/project2/epm-4.3-source.tar.gz
#         $ ./configure --prefix=/usr/local ; make install
#
#     o openssl (1.0.2j)
#         https://www.openssl.org/source/openssl-1.0.2j.tar.gz
#         $ export MACOSX_DEPLOYMENT_TARGET=10.7
#         $ ./Configure darwin64-x86_64-cc no-shared --prefix=/usr/local --openssldir=/usr/local ; make install
#
#   Env:
#     LIBRARY_PATH=/usr/local/lib
#     C_INCLUDE_PATH=/usr/local/include
#     CPLUS_INCLUDE_PATH=/usr/local/include
#     MACOSX_DEPLOYMENT_TARGET=10.7
#     PATH=/opt/local/bin:/opt/local/sbin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:.
#
#   Notes:
#     o openssl required for serf; we build non-shared so all
#       linkings are to the static libs
#
#     o JDK 1.7 seems to work better; and it's always best to
#       have just *one* version of te JDK installed.
#
if [ ! -d ../main -o ! -d sal ] ; then
        echo "CHDIR into AOO's main/ directory first!"
        exit 1
fi
if [ -z "$JAVA_HOME" ] ; then
        JAVA_HOME=$(/usr/libexec/java_home -v 1.7)
        export JAVA_HOME
        echo "Setting JAVA_HOME to $JAVA_HOME..."
fi
if [ -z "$ANT_HOME" ] ; then
        ANT_HOME=/usr/local/share/java/apache-ant
        export ANT_HOME
        echo "Setting ANT_HOME to $ANT_HOME..."
fi
if [ ! -e external/unowinreg/unowinreg.dll ] ; then
        echo "Downloading unowinreg.dll..."
        curl -o external/unowinreg/unowinreg.dll http://www.openoffice.org/tools/unowinreg_prebuild/680
/unowinreg.dll
fi
LANGS="ast bg ca ca-XR ca-XV cs da de el en-GB en-US es eu fi fr gd gl he hi hu it ja kid km ko lt nb nl pl pt
pt-BR ru sk sl sr sv ta th tr vi zh-CN zh-TW"
if [ ! -e configure -o configure.in -nt configure ] ; then
        echo "Running autoconf..."
        autoconf || exit 1
fi
./configure   \
    --with-build-version="$(date +"%Y-%m-%d %H:%M:%S (%a, %d %b %Y)") - `uname -sm`" \
        --with-vendor="Apache OpenOffice Community Build" \
        --enable-verbose \
        --with-openldap \
        --enable-category-b \
        --enable-bundled-dictionaries \
        --enable-wiki-publisher \
        --with-junit="/usr/local/share/java/junit.jar" \
```

```
        --with-jdk-home="$JAVA_HOME" \
        --with-ant-home="$ANT_HOME" \
        --with-epm=/usr/local/bin/epm \
        --with-dmake-path=/usr/local/bin/dmake \
        --without-stlport \
        --with-package-format="installed dmg" \
        --disable-systray \
        --with-alloc=system \
        --with-lang="${LANGS}"
./bootstrap || exit 1
source ./MacOSXX64Env.Set.sh || exit 1
cd instsetoo_native
time perl "$SOLARENV/bin/build.pl" --all -P2 -- -P2 || exit 1
cd util
dmake ooolanguagepack || exit 1
dmake sdkoo_en-US || exit 1
date "+Build ended at %H:%M:%S"
```

Windows

Visual Studio 2008 Professional installed (from MSDN subscription available for Apache committers).

The relevant configure options are:

```
--without-junit \
--without-stlport \
--enable-win-x64-shellext \
--enable-category-b \
--enable-bundled-dictionaries \
--with-mozilla-build=c:/mozilla-build \
```

Linux

Linux build flags for AOO >= 4.0.1

Linux packages were built on CentOS 5 with the following flags:

```
./configure   \
--with-build-version="$(date +"%Y-%m-%d %H:%M") - `uname -sm`" \
--enable-verbose \
--enable-category-b \
--enable-crashdump=yes  \
--enable-wiki-publisher  \
--enable-opengl  \
--enable-dbus  \
--enable-gstreamer \
--with-package-format="rpm deb" \
--with-dmake-url=https://sourceforge.net/projects/oooextras.mirror/files/dmake-4.12.tar.bz2 \
--with-epm-url=https://sourceforge.net/projects/oooextras.mirror/files/epm-3.7.tar.gz \
--enable-bundled-dictionaries \
--without-junit \
--with-perl-home=$HOME/perl \
--with-ant-home=$HOME/src/apache-ant-1.8.3 \
--without-stlport \
--with-system-stdlibs \
--with-lang="ast bg ca ca-XV cs de da el en-GB en-US es eu fi fr gd gl he hi hu it ja km ko lt nb nl pl pt pt-
BR ru sk sl sr sv ta th tr vi zh-CN zh-TW"
```

To speed up the building process I use prebuilt mozilla (built by myself on the same environment).

Solaris

I (Mouette Yang) build in the platform Solaris 10 u10 8/11 x86, and use SolarisStudio12.3 Compiler:http://www.oracle.com/technetwork/server-storage/solarisstudio/overview/index.html
Some need build environment tools is placed here:http://www.openfoundry.org/of/download_path/osscd/3.4.0m1_Rev1227012/BuildAOO-Env-GNUtools_solaris_i86pc.pkg
And need subversion can be download in here:http://www.blastwave.org/

Env variables is set:

```
export CC=/opt/solarisstudio12.3/bin/cc
export CXX=/opt/solarisstudio12.3/bin/CXX
export LD_LIBRARY_PATH=/opt/solarisstudio12.3/lib:/usr/sfw/lib
export PATH=/usr/gnu/bin/:$PATH:/usr/sfw/bin
```

And my configure options are:

```
./configure \
--disable-mozilla \
--disable-cups \
--with-system-python \
--enable-presenter-console \
--enable-wiki-publisher \
--enable-minimizer \
--with-lang=[lang options...] \
--with-junit=[junit.jar's path] \
--with-build-version=[Date and revision]
```

And other need package and developer snapshot's modify, i write a script, can auto deal with these things until build finish:

```
#!/usr/bin/tclsh8.5
package require platform
set confarea "/DocumentArea/scripts"
#Write a file "AOO-LANG", place it in $confarea , it define language list, it will modify
#languages list in file "main/instsetoo_native/util/pack.lst" and configure's parametre "--with-lang"

set buildarea "/BuildArea/ooo"
#Define AOO source place.

set os [platform::identify]
set buildbin "main/solenv/bin/build.pl"
set svnrev [lindex [lindex [split [exec /opt/csw/bin/svn info $buildarea] \n] 9] end ]
set builddate [clock format [clock seconds] -format "%Y-%m-%d %H:%M:%S"]
set dev_parametre ""

#=======Define AOO Build Type====================
#option :normal,devsnap
# "normal" are normal installation sets
# "devsnap" are Development Snapshots

set buildtype "devsnap"
#================================================

#=======Define Special Platform Parametres=======
set solaris10_special "--disable-mozilla --disable-cups --with-system-python \
  --enable-presenter-console --enable-wiki-publisher --enable-minimizer"
set solaris11_special ""
set linux_special ""
#================================================
if {$os=="solaris2.10-ix86"} { set os_parametre $solaris10_special }
if {$os=="solaris2.11-ix86"} { set os_parametre $solaris11_special }

#=======Define Compiler and System Env===========
if {$os=="solaris2.10-ix86" || $os=="solaris2.11-ix86"} {
  set CC "/opt/solarisstudio12.3/bin/cc"
  set CXX "/opt/solarisstudio12.3/bin/CC"
  set PATH "/usr/gnu/bin:/usr/bin:/usr/sbin:/usr/sfw/bin:/usr/ccs/bin"
  set LD_LIBRARY_PATH "/opt/solarisstudio12.3/lib:/usr/sfw/lib"
  set platenv "SolarisX86Env.Set.sh"
```

```
}
#================================================

#=======Define Language==========================
set lf [open "$confarea/AOO-LANG" r]
set langlist [lrange [split [read $lf] \n] 0 end-1]
close $lf
set lang_parametre "\"$langlist\""
foreach i $langlist {
  append lanlist1 "$i,"
  append lanlist2 "$i|"
}
set lanlist1 [string range $lanlist1 0 end-1]
set lanlist2 [string range $lanlist2 0 end-1]
#================================================

#===========Autres================================
set autre_parametre "--with-junit=/BuildArea/junit.jar "
#Add other need parametres,ex junit's place
#================================================

#===========Define Build Type====================
if {$buildtype=="normal"} {
  set buildwork "cd $buildarea/main/instsetoo_native\n"
  append buildwork "LD_LIBRARY_PATH=$LD_LIBRARY_PATH $buildarea/$buildbin --all"
}

if {$buildtype=="devsnap"} {
  set buildwork "cd $buildarea/main/instsetoo_native\n"
  set utilpath  "$buildarea/main/instsetoo_native/util/"
  file copy -force $utilpath/makefile.mk $utilpath/makefile.mk-bak
  file copy -force $utilpath/pack.lst $utilpath/pack.lst-bak
  set fm [open $utilpath/makefile.mk r]
  set pakmk [split [read $fm] \n]
  close $fm
  set fm [open $utilpath/makefile.mk w]
  foreach i $pakmk {
    if {$i=="ALLTAR : openoffice sdkoo_en-US ure_en-US"} {
      puts $fm "ALLTAR : updatepack"
    } else {
      puts $fm $i
    }
  }
  close $fm

  set fp [open $utilpath/pack.lst w]
  puts $fp "OpenOffice_Dev_multilang    unxlngi6.pro,unxlngx6.pro,unxmacxi.pro,unxsoli4.pro,unxsols4.pro,
wntmsci12.pro   $lanlist1   openofficedev"
  puts $fp "OpenOffice_Dev_SDK    unxlngi6.pro,unxlngx6.pro,unxmacxi.pro,unxsoli4.pro,unxsols4.pro,wntmsci12.
pro   en-US    sdkoodev"
  puts $fp "OpenOfficeDevLanguagepack   unxlngi6.pro,unxlngx6.pro,unxmacxi.pro,unxsoli4.pro,unxsols4.pro,
wntmsci12.pro    $lanlist2   ooodevlanguagepack"
  close $fp

  append buildwork "LD_LIBRARY_PATH=$LD_LIBRARY_PATH $buildarea/$buildbin --all"

}
#================================================

#=======Generate Build Script====================
set f [open $buildarea/buildaoo.sh w]
puts $f "#!/usr/bin/bash"
puts $f "cd $buildarea/main"
puts $f "CC=$CC CXX=$CXX PATH=$PATH LD_LIBRARY_PATH=$LD_LIBRARY_PATH \
  $buildarea/main/configure $os_parametre --with-lang=$lang_parametre \
  $autre_parametre $dev_parametre --with-build-version=\"$builddate - Rev.$svnrev\""
puts $f "sed 's/sh/bash/' $buildarea/main/fetch_tarballs.sh > tmp"
puts $f "cat tmp > $buildarea/main/fetch_tarballs.sh"
puts $f "bash $buildarea/main/bootstrap"
puts $f ". $buildarea/main/$platenv"
puts $f $buildwork
```

```
close $f
#================================================

#============Start Build=========================
set mf [open /tmp/mstat w]
puts $mf "Building AOO ..."
close $mf
cd $buildarea/main/instsetoo_native
exec -ignorestderr sh $buildarea/buildaoo.sh > $buildarea/buildlog
set mf [open /tmp/mstat w]
puts $mf "Idle"
close $mf
#================================================
```

Define language list in this file "/DocumentArea/scripts/AOO-LANG"
(if $confarea is /DocumentArea/scripts):

```
ast
bg
cs
da
de
el
en-GB
en-US
es
eu
fi
fr
gd
gl
hi
hu
it
ja
km
ko
lt
nb
nl
pl
pt
pt-BR
ru
sk
sl
sr
sv
ta
th
tr
vi
zh-CN
zh-TW
```