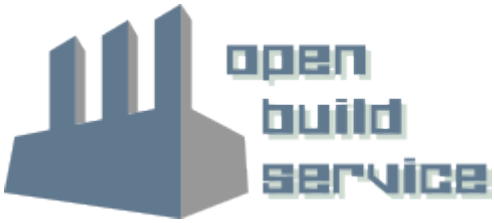
Open Build Service: creazione pacchetti

Introduzione



L'Open Build Service (OBS)[(obs_site>)] è un sistema realizzato da Suse per centralizzare la gestione della creazione di pacchetti per la loro distro e altre.

Le caratteristiche principali di questo sistema sono le seguenti:

1. gestione dei pacchetti raggruppati in progetti, staccati l'uno dall'altro
2. compilazione dei sorgenti in ambienti isolati e sempre congruenti
3. creazione automatica repository pacchetti
4. versioning dei file di specifica e dei sorgenti utilizzati, in maniera simile a CVS/SVN
5. API web per poter automatizzare certi aspetti delle build tramite chiamate HTTP

Quello che questo sistema non è in grado di effettuare (almeno per il momento) è l'esecuzione di build che prevedano lo scaricamento di qualsiasi tipo di informazione da internet, in quanto gli ambienti all'interno dei quali vengono effettuate le build sono isolati dalla rete per garantire la coerenza delle build.[(obs_isolated)]

Ogni sistema OBS presenta tre servizi agli utenti:

1. l'interfaccia di gestione web (la webui), normalmente su HTTPS
2. i repository dei pacchetti realizzati, normalmente su porta 82
3. la web API, esposta normalmente su porta HTTPS [(obs_webapiport)]

Di queste componenti verrà effettuato un excursus nelle prossime sezioni.

Attualmente il sistema OBS è in uso per produrre i pacchetti Centos/Redhat di eXtraWay Enterprise Edition.

Ambienti installati

Attualmente sono installati 2 ambienti OBS in 3DI:

- obs-test.bo.priv, che contiene un'istanza OBS v2.6 sulla quale sono stati fatti i primi test di compilazione di extraway enterprise per Centos7;
- obs.bo.priv, che contiene un'istanza OBS v2.6, sulla quale sono stati copiati i pacchetti realizzati su obs-test e sulla quale si è proseguito con la creazione di nuovi pacchetti.

Entrambi i sistemi sono installati all'interno di due macchine virtuali senza grosse potenzialità computazionali, per cui si caldeggia, in vista di compilazioni consistenti, di istanziare un certo numero di OBS worker su altre macchine client più potenti e non sottoposte ad altri carichi per parallelizzare le build.

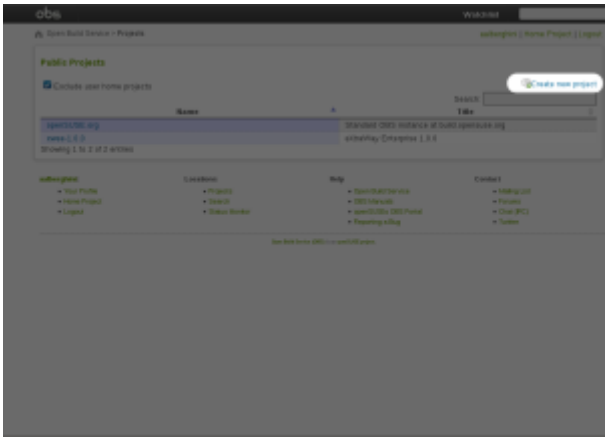
Operazioni fondamentali

Creazione di un nuovo progetto

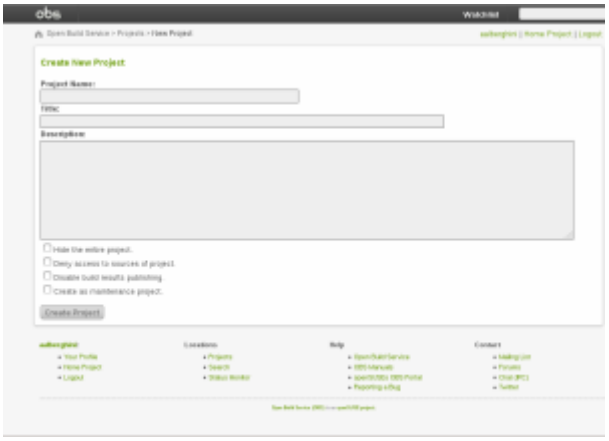
Da webui

1. Dopo aver effettuato l'accesso, selezionare `All Projects`

2. Selezionare `Create new project`



3. Inserire gli estremi del nuovo progetto, ovvero un nome, un titolo ed una descrizione (opzionale) e cliccare su `Create Project`



Tramite OSC

1. Immettere il seguente comando

   ```
   osc -A ALIAS meta prj -e [project]
   ```

   dove ALIAS indica l'alias configurato in `~/.oscrc` da utilizzare per puntare all'host corretto
2. Verrà aperto un editor di un file XML che contiene i metadati del nuovo progetto (o di un progetto già esistente, nel caso il nome indicato esista già).
3. Inserire i dati del nuovo progetto e salvare.

Creazione di un nuovo pacchetto

Checkout di un intero progetto

Checkout di un pacchetto

Deploy di un pacchetto

Deploy di un pacchetto - CURL

Procedura di rilascio per eXtraWay Enterprise

Ad ogni nuova release di eXtraWay Enterprise, i passi da seguire sono i seguenti:

- creare un nuovo tarball di sorgenti mediante lo script makexweepackage.sh
- caricare il tarball creato su \\storage\SRC_REPO\3di\xwee\xwee (per mantenere uno storico)
- aggiornare il file .spec relativo ad xwee, modificando la versione in modo da puntare al nuovo tarball
- aggiornare la sezione changelog in coda al file .spec con la data corretta ed il dettaglio delle modifiche al pacchetto
- deploy dei nuovi sorgenti (sia .spec sia sorgenti veri e propri di xwee) sul corretto progetto OBS

Script makexweepackage

Questo script è stato realizzato per creare un tarball di sorgenti omnicomprensivo della distribuzione di eXtraWay Enterprise, poiché il progetto è stato spezzato in due parti (core e modules) a livello di sorgenti.

La struttura che viene approntata all'interno del tarball è già quella attesa dagli script .spec utilizzati per la creazione dei pacchetti RPM.

REQUISITI per l'utilizzo

È necessario specificare il proprio utente CVS per poter effettuare i vari checkout mediante la variabile CVSUSER. Tale variabile può essere specificata in due modi:

- mediante variabile d'ambiente, eg.

  ```
  CVSUSER=pinco-pallino-3di ./makexweepackage.sh
  ```

- modificandone la definizione nello script

Ovviamente l'utente specificato dovrà avere permesso di accesso al percorso /opt/cvs/xw sul CVS, ovvero dovrà far parte del gruppo **xwdevel** sul server CVS.

Lo script richiede diversi parametri per funzionare, i quali permettono sostanzialmente di specificare con precisione quale versione scaricare di entrambi i progetti.

Esempi di utilizzo dello script

Scaricamento di entrambi i progetti in versione HEAD:

```
./makexweepackage.sh -h
```

Scaricamento di entrambi i progetti ad una particolare data YYYYMMDD:

```
./makexweepackage.sh -d YYYYMMDD
```

Scaricamento di entrambi i progetti al tag XYZ:

```
./makexweepackage.sh -t XYZ
```

Scaricamento del modulo core come HEAD e del modulo modules al tag XYZ:

```
./makexweepackage.sh -xh -mt XYZ
```

Scaricamento del modulo core come HEAD e del modulo modules alla data YYYYMMDD:

```
./makexweepackage.sh -xh -md YYYYMMDD
```

Scaricamento del modulo core al tag XYZ e del modulo modules alla HEAD:

```
./makexweepackage.sh -xt XYZ -mh
```

Scaricamento del modulo core alla data YYYYMMDD e del modulo modules alla HEAD:

```
./makexweepackage.sh -xd YYYYMMDD -mh
```

Scaricamento del modulo core al tag XYZ e del modulo modules al tag ZYX:

```
./makexweepackage.sh -xt XYZ -mt ZYX
```

Scaricamento del modulo core al tag XYZ e del modulo modules alla data YYYYMMDD:

```
./makexweepackage.sh -xt XYZ -md YYYYMMDD
```

Scaricamento del modulo core alla data YYYYMMDD e del modulo modules al tag XYZ:

```
./makexweepackage.sh -xd YYYYMMDD -mt XYZ
```

Scaricamento del modulo core alla data YYYYMMDD e del modulo modules alla data YYYYMMDD2:

```
./makexweepackage.sh -xd YYYYMMDD -mt YYYYMMDD2
```

Codice dello script

makexweepackage.sh

```
#!/bin/bash -x
# Author: Alan Alberghini 2015, Roberto Tirabassi 2016
# Version: 1.1


PRGNAM=xwee
DATE="$(date +%Y%m%d)"
VERSION=${VERSION:-$DATE}
OUTDIR=${OUTDIR:-/tmp}
CWD="$(pwd)"

set -e

##### CVS INFORMATION ##################

CVSUSER=${CVSUSER:-rtirabassi-3di}

XWEE_CVSROOT=:ext:"${CVSUSER}"@cvs.3di.it:/opt/cvs/xw
XWEE_MODS_CVSROOT=:ext:"${CVSUSER}"@cvs.3di.it:/opt/cvs/xw

XWEE_CVSNAME=eXtraWay.EE.core
XWEE_MODS_CVSNAME=eXtraWay.EE.modules

XWEE_TAG=${XWEE_TAG:-xwee_20151008}
XWEE_MODS_TAG=${XWEE_TAG:-xwee_20151008}


###### END OF CVS INFORMATION ###########

# Checkout project from CVS
# Arguments:
#       $1: CVSROOT
#       $2: MODULE
#       $3: HEAD | TAG | DATE
#       $4: the specific tag or date, depending on $2

function project_checkout() {


        CVSROOT=$1
        MODULE=$2

        cat << EOF
Checking out project $MODULE with this information:
CVSROOT:        $CVSROOT
VERSION:        $3
VERSION_SPEC:   $4
EOF

        case $3 in
        HEAD)
                CVSROOT="${CVSROOT}" cvs co "${MODULE}"
                ;;
        TAG)
```

```
                CVSROOT="${CVSROOT}" cvs co -r "${4}" "${MODULE}"
                ;;
        DATE)
                CVSROOT="${CVSROOT}" cvs co -D "${4}" "${MODULE}"
                ;;
        *)
                echo "Are you trying to bug me?"
                exit 1
                ;;
        esac
}


function print_usage () {
        cat << EOF
Usage:
        $(basename $0) OPTIONS
        Where options can be:

        -h                      Downloads head revision for both projects
        -t                      TAG  Downloads tag TAG for both projects
        -d                      DATE Downloads version due on date DATE for both projects
        -xh | -xt TAG | -xd DATE
                                Downloads head, tag or date revision for the xwee project
        -mh | -mt TAG | -md DATE
                                Downloads head, tag or date revisione for the modules
project
        -v                      Set a version tag for the output archive instead of the
default date (YYYYMMDD)
EOF
}


if [ $# -eq 0 ]
then
        print_usage
        exit 0
fi

XWEE_VERSION=""
XWEE_MODULES_VERSION=""
XWEE_VERSION_SPEC=""
XWEE_MODULES_VERSION_SPEC=""

# Simple parameters parsing
while getopts "h t:d: v: xh xt:xd: mh mt:md:" OPTION
do
        case "$OPTION" in
        h)
                XWEE_VERSION=HEAD
                XWEE_MODULES_VERSION=HEAD
                ;;
        t)
                XWEE_VERSION=TAG
                XWEE_MODULES_VERSION=TAG
                XWEE_VERSION_SPEC="$OPTARG"
                XWEE_MODULES_VERSION_SPEC="$OPTARG"
                ;;
        d)
                XWEE_VERSION=DATE
                XWEE_MODULES_VERSION=DATE
                XWEE_VERSION_SPEC="$OPTARG"
                XWEE_MODULES_VERSION_SPEC="$OPTARG"
                ;;
        v)
```

```
                VERSION="$OPTARG"
                ;;
        xh)
                XWEE_VERSION=HEAD
                ;;
        xt)
                XWEE_VERSION=TAG
                XWEE_VERSION_SPEC="$OPTARG"
                ;;
        xd)
                XWEE_VERSION=DATE
                XWEE_VERSION_SPEC="$OPTARG"
                ;;
        mh)
                XWEE_MODULES_VERSION=HEAD
                ;;
        mt)
                XWEE_MODULES_VERSION=TAG
                XWEE_MODULES_VERSION_SPEC="$OPTARG"
                ;;
        md)
                XWEE_MODULES_VERSION=DATE
                XWEE_MODULES_VERSION_SPEC="$OPTARG"
                ;;
        *)
                echo "Are you trying to bug me?"
                exit 1
                ;;
        esac
done

if [ -z "$XWEE_VERSION" -o -z "$XWEE_MODULES_VERSION" ]
then
        echo "You're missing a version specification:"
        echo "XWEE_VERSION=$XWEE_VERSION"
        echo "XWEE_MODULES_VERSION=$XWEE_MODULES_VERSION"
        exit 1
fi

export CVS_RSH=ssh

TMPDIR=$(mktemp -d)

cd "$TMPDIR"

# Checkout xwee.core
project_checkout "$XWEE_CVSROOT" "$XWEE_CVSNAME" "$XWEE_VERSION" "$XWEE_VERSION_SPEC"
cd "${XWEE_CVSNAME}"/src
# Checkout xwee.modules inside xwee.core/src
project_checkout "$XWEE_MODS_CVSROOT" "$XWEE_MODS_CVSNAME" "$XWEE_MODULES_VERSION"
"$XWEE_MODULES_VERSION_SPEC"
# Remove CVS metadata folders
find . -type d -name CVS -exec rm -rf {} \; || true
# Remove test targets
#find . -type d -name test\* -exec rm -rf {} \; || true
# Move xwee.modules folder contents inside src/
mv  -t . "${XWEE_MODS_CVSNAME}/"*
rm -r "${XWEE_MODS_CVSNAME}"

# Add an info file for the sources
cat  > checkout.info << EOF
Checkout date: $DATE

$XWEE_CVSNAME version: $XWEE_VERSION version_spec: $XWEE_VERSION_SPEC
```

```
$XWEE_MODS_CVSNAME version: $XWEE_MODULES_VERSION version_spec: $XWEE_MODULES_VERSION_SPEC
EOF

cd ..
mv src "${PRGNAM}-${VERSION}"
tar vcjf "${OUTDIR}/${PRGNAM}-${VERSION}.tar.bz2" "${PRGNAM}-${VERSION}"
cd "$CWD"
rm -rf "$TMPDIR"
```

Gestione pacchetti

Scaricamento pacchetto tramite OSC

Deploy tramite OSC

Deploy tramite CURL

Riferimenti esterni

Info generali

[(obs_site> Sito ufficiale del progetto)]

[(obs_tutorial> Tutorial iniziale)]

[(obs_webapiport>Nelle versioni precedenti la 2.6 di OBS, la porta per la web API era la 444)]

[(obs_isolated>http://stackoverflow.com/a/33789070)]

Setup di un OBS Worker

Modalità di delivery dei pacchetti/repository

Documentazione per lo sviluppo

Home page del progetto su Github

OSC

Documentazione su OSC

Tutorial su OSC

Workers

Immagine docker per un OBS Worker