

BlueLinQ 対応の実際

2002年 9月 4日
マーズ株式会社 大塚 芳徳

Sun Cobalt BlueLinQ

BlueLinQ とは？

ソフトウェアの配布、更新の自動通知メカニズムをもち、WEB ブラウザからソフトウェアパッケージのインストール、更新、削除を行うことができる機能です。



パッケージ (pkg) の構造 1

パッケージとは？

rpm (Red Hat Package Manager) 形式のアーカイブと Cobalt 独自のスクリプトを TAR + GZIP で固めたものです。



パッケージ (pkg) の構造 2

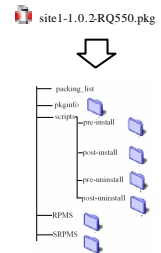
ディレクトリ構成

packing_list (ファイル)
パッケージの情報を記述します。

pkginfo
ロケール毎のリソースなどを設置します。

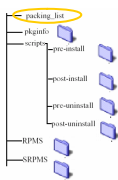
scripts
スクリプトを設置します。

RPMS
rpm を設置します。



packing_list

```
[Package -- Version=1.0]
Vendor:      mars
VendorTag:  [[vendor]]
Name:        site1
NameTag:     [[name]]
Version:     {ant:site1-version}
PackageType: complete
Options:     uninstallable,refreshui,refreshcse
LongDesc:   [[longDesc ]]
ShortDesc:  [[shortDesc ]]
Copyright:  [[copyright]]
InfoURL:    [[infoUrl ]]
Splash:     pre-install
Product:    {ant:pkg:product}
Depend:     CobaltOS >= 6.0
RPM:        site1-{ant:site1-version}-BLQ.i386.rpm
[/Package]
```



pkginfo - locale

[pkginfo] - [locale] - [<lang>] - messages.mo

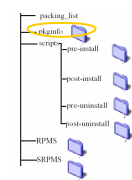
.po ファイルの書式は以下のとおりです。

```
msgid "<stringId >"
msgstr "Internationalization language."
注) 日本語もいけます。。
```

.mo ファイルは msgfmt というツールを使ってフォーマットします。

```
$ msgfmt -e messages.po -o messages.mo
```

とすると messages.mo というファイルができます。



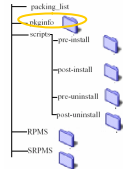
pkginfo - splash

[pkginfo] - [splash] - [pre-install] - index.cgi

通常、下記の 1 - 3 のプロセスでインストールが行われますが、splash ディレクトリを追加することで、2 の部分に、独自のインストールプロセスを構築することができます。

1. パッケージの概要表示
2. 使用許諾
3. インストール

1 のパッケージの概要表示の後、「インストール」を選択すると、システムにより自動的に index.cgi または index.php に submitURL と cancelURL の 2 つの GET パラメータと共にリクエストが転送されます。開発者はこのパラメータを頼りに、CGI を作成します。



7

pkginfo - scripts

rpm のインストール、アンインストールの前後に実行されるスクリプトを配置するディレクトリです。

インストールボタンの押下

pre-install ディレクトリ内のスクリプトの実行

rpm のインストール

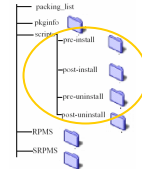
post-install ディレクトリ内のスクリプトの実行

アンインストール ボタンの押下

pre-uninstall ディレクトリ内のスクリプトの実行

rpm のアンインストール

post-uninstall ディレクトリ内のスクリプトの実行



8

BlueLinQ インストールデモ

デモ中

9

WEB STYLE SITE1 とは

- サブレットコンテナ、RDBMS を使用する Java で記述された WEB アプリケーション。
- HTML に独自テンプレートを使用した WEB STYLE という技術を使用。

10

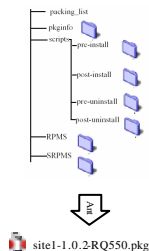
Jakarta Ant による pkg 開発

• Jakarta Ant とは？

java と XML でビルドの自動化を行う make に似たツール。

• 利点

java のプロジェクトと親和性が高い。
tar, gzip, rpm (optional) が使用でき、pkg を開発する上で申し分ない。
Colbat のシリーズの違いによる対応を自動化。



11

build.xml (sample)

```
<?xml version="1.0" encoding="EUC-JP"?>
<project name="site1" default="all" basedir=".">
  <property name="version" value="1.0.3" />
  <property name="jreversion" value="1.3.1_04" />
  ...
  <target name="pkg-RQ4">
    <mkdir dir="${pkgworkdir}" />
    <copy todir="${pkgworkdir}/RPMS" overwrite="yes">
      <fileset dir="${archive.dir}/mysql" includes="MySQL.*-${mysqlversion}-RQ4.i386.rpm"/>
    </copy>
    <replace file="${pkgworkdir}/packing_list" token="[ant:site1-version]" value="${version}" />
    <tar tarfile="${pkgworkdir}/${site1}-RQ4.tar" basedir="${pkgworkdir}" excludes="*.pkg" />
    <gzip zipfile="${pkgworkdir}/${site1}-RQ4.pkg" src="${pkgworkdir}/${site1}-RQ4.tar" />
    <delete file="${pkgworkdir}/${site1}-RQ4.tar" />
    <move todir="${dst.dir}" file="${pkgworkdir}/${site1}-RQ4.pkg" overwrite="yes" />
  </target>
</project>
```

12

苦労した点

- 日本語の情報が少ない

英語がスラスラ読めれば問題ないのですが。。

- Cobalt のシリーズによる違い。

同じ拡張子(.pkg)でも RaQ4 と Qube3 はほとんど違う構造をしている。

同じ構造の(.pkg)でも Qube3 と RaQ550 は微妙に違う。(pkg 内で使用するリソースの文字エンコーディングや、最初にインストールされているアプリケーションなど)