

OpenRTM-aist C++インストーラのアップグレード動作確認

1.1.0-RELEASE版がインストールされている環境へ1.1.1-RELEASE版をインストールしてアップグレードできる動作を確認できた。

(参考までに: PythonとJavaの各1.1.0-RELEASE版はアップグレードできない)

2015/01/06の作業報告より

確認結果: 以下の順に操作 (vc10 32bit環境でのテスト)

- 1) 1.1.0版をインストール
- 2) 1.1.1版をインストール
古いバージョンを削除してからインストールしている。
途中で確認メッセージ等はない。
- 3) 続けて1.1.1版をインストール
インストーラのメニューは、Change, Repair, Removeの3択になる
テストはキャンセルで終了させた。
この後、次の確認のために1.1.1版をアンインストール。
- 4) 新規で1.1.1版をインストール
問題なし。

この動作実現のために設定したこと

ちょうどTortoiseSVNをアップグレードしたばかりだったので、手元にあったバージョン違いのmsiファイルをwxsに展開して中身を調べる。
TortoiseSVN-1.8.8.25755-x64-svn-1.8.10.msi
TortoiseSVN-1.8.10.26129-x64-svn-1.8.11.msi

msiファイルをwxsに展開する処理は、darkを使う
(c:\Program Files (x86)\WiX Toolset v3.8\bin\dark.exe)
パッチスクリプトにしておく使いやすい。

```
----- dark.bat
@echo off
set WiX_BIN_DIR="c:\Program Files (x86)\WiX Toolset v3.8\bin"
set name=%~n1
echo %1
echo %name%

%WiX_BIN_DIR%dark.exe %1 %name%.wxs
-----
```

使い方は、引数として展開したいmsiファイル名を指定する。
> dark.bat aaa.msi
こうすると、aaa.wxs ファイルが出力される。

この2つのmsiは、Product GUIDを変更し、UpgradeCode GUIDを同一を使用している。また、MajorUpgradeタグを使っている。

これを参考にし、msi作成時のOpenRTM.wxsへ下記コードを追加する。

```
----- OpenRTM.wxs を追加
<MajorUpgrade AllowDowngrades="yes" Schedule="afterInstallValidate" />
<Property Id="SELFRTMFOUND" Secure="yes" />
<Property Id="WIXUI_INSTALLDIR" Value="INSTALLLOCATION" />
<Upgrade Id="$(env.UPGRADE_GUID)">
  <UpgradeVersion Minimum="$(env.OPENRTM_VERSION)" Property="SELFRTMFOUND" OnlyDetect="yes" IncludeMinimum="yes" />
</Upgrade>
</Product>
</Wix>
-----
```

MajorUpgrade タグ

- ・AllowDowngrades: デフォルトは"no"なのだが、"no"にするとDowngradeErrorMessageを定義する必要がある。UpgradeVersionタグで対象バージョンを限定するので、ひとまずここは"yes"にしておく。
(TortoiseSVNをまねる)
- ・"afterInstallValidate"はScheduleのデフォルト値。
removes the installed product entirely before installing the upgrade product.

UpgradeVersion タグ

- ・IncludeMinimum="yes" は Minimumで指定されたバージョンより低いか等しいバージョンを探す
- ・OnlyDetect="yes"は検出するが削除はしない

このように設定してアップグレード動作を実現できたが、実現できなかった時の設定も失敗例として記録に残す。

失敗例 1

- ・ProductタグのUpgradeCode GUIDだけを同一とし、Product GUIDは変更したら新旧バージョンが共存してしまった
- ・メジャー・アップグレードと判断されたようだ
- ・確かに、メジャー・アップグレードの条件のVersion属性とPackage GUIDを変更することも満たしていた

失敗例 2

- ・バージョンが変わるマイナー・アップグレードなので、Product GUIDとUpgradeCode GUIDを同一、Version属性とPackage GUIDを変更した。
Versionは1.1.0と1.1.1、Package GUIDは*指定で自動取得するので異なる。

動作結果：以下のメッセージダイアログが表示され処理が止まる。以前のバージョンを手動で削除しない限りアップグレードできない。

メッセージ：

別のバージョンの製品が既にインストールされています。

このバージョンのインストールを続行できません。

既にインストールされているバージョンの製品を構成、または削除するには、コントロールパネルの[プログラムの追加と削除]アイコンを使用します。