

Práca s triedou %Installer

V tomto pokračovaní seriálu vám ukážeme použitie ešte stále pomerne čerstvej novinky, určenej na používateľskú úpravu inštalácie Caché. S príchodom verzie 2009.1 sa súčasťou Caché stal balík % Installer, ktorej hlavnou úlohou je poskytnúť komfortné API na tvorbu upravených inštaláčnych skriptov Caché. Mnohí z vás si inštaláciu Caché upravujete tak, aby okrem servera inštalovala aj aplikáciu kód a predkonfigurovala runtime prostredie aplikácie. Z vlastnej skúsenosti viete, že možnosti Caché v tejto oblasti neboli kvotieaké. To sa teraz podstatne mení.

Ako balík % Installer funguje? Základnou triedou na prácu je % Installer.Installer. Táto trieda poskytuje rozhranie pre vašu vlastnú inštaláčnu triedu, ktorú vytvoríte tak, že nebude dedič od žiadneho predka. Vaša trieda však musí obsahovať ľubovoľne nazvaný element XData, ktorý musí mať deklarovaný [XMLNamespace = INSTALLER]. Tým kompilátor spozná, že ide o inštaláčnu triedu, a vygeneruje správny kód na pripojenie a spustenie pri inštalácii Caché. Na správnu kompiláciu nesmiete zabudnúť pridať do deklarácie triedy direktívu `import % occlude`.

Čo všetko sa dá v kontajneri XData definovať? Namespace INSTALLER obsahuje prvky <Manifest>, <Var>, <Namespace>, <Configuration> a mnohé ďalšie. Tieto prvky definujú konfiguráciu inštalácie. Ešte sa im budeme venovať neskôr. Okrem konfiguráčnych prvkov existujú aj prvky <Invoke> a <Import>, ktoré umožňujú definovať metódy, ktoré sa majú vykonať v priebehu inštalácie. Okrem kontajnera XData musíte v inštaláčnej triede deklarovať a implementovať metódu `setup ()`. Poďme sa teraz pozrieť na jednotlivé prvky a metódu `setup ()` bližšie.

Prehľad vybraných elementov kontajnera XData:

| Element | Opis |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Manifest | Koreňový prvok XML stromu |
| Namespace | Povinný - potomok prvku Manifest, obsahuje definíciu novo vytváraného/modifikovaného názovového priestoru. Má atribúty Create, Code (dataset s rutinami) a Data (dataset s globálmi). |
| Configuration | Povinný - potomok prvku Namespace, skupinový prvok. |
| Database | Povinný - potomok prvku Configuration, obsahuje informáciu o vytváranom/modifikovanom datase. Má okrem iného tieto atribúty: Dir, Create, Resource, Blocksize, Collation, Initial-Size a iné. |
| Var | Voliteľný prvok - potomok prvku Manifest, definuje premenné použiteľné v rôznych častiach dokumentu XML. |
| Import | Voliteľný - potomok prvku Namespace, definuje súbor s globálmi, rutinami či triedami, ktorý sa má načítať. |
| Invoke | Voliteľný - potomok prvku Namespace, definuje metódu triedy, ktorá sa má vykonať po importe rutín, globálov či tried. |

Na to, aby sa spustil vygenerovaný kód opísaný v kontajneri XData, musíme definovať už spomínanú metódu `setup ()`. Jej implementácia musí vyzeráť takto:

```
ClassMethod setup(ByRef pVars, pLogLevel As
%Integer, pInstaller As %Installer.Installer)
As %Status [ CodeMode = objectgenerator, Internal ]
{
  #; Let XGL document generate code for
  this method.
  Quit ##class(%Installer.Manifest).%Generate(%compiledclass,%code, nazevXDataKontajneru)
}
```

Jednoduchá inštaláčna trieda využívajúca opísaný prvok môže vyzeráť napríklad takto (krátené):

```
<XData INSTALL [ XMLNamespace = INSTALLER ]
{
  <Manifest>
  <!-- VARIABLES -->
  <Var Name="SOURCE_PATH"
Value="c:\kutac\install\cache2010\" />
  <Var Name="FILE_LOCALE"
Value="loc_ycsw.xml" />
  <Var Name="FILE_LOADER"
Value="kutac.loader.rsa" />
  <Var Name="NEW_NS" Value="KUTAC" />
  <Var Name="SYS_NS" Value="%SYS" />
  <Var Name="SET_SUPERUSER" Value="1" />
  <Var Name="SET_UNKNOWN_USER"
Value="1" />
  <Var Name="SET_UPGRADE_ALL" Value="1" />
  <Var Name="IS_UPGRADE"
Value="##class(%SYS.Namespace).Exists("%NEW_NS")' />

  <!-- nova instalace -->
  <If Condition="%IS_UPGRADE"=0">
  <Log Topic="KUTAC.Demo" Text="Provadim
cistou instalaci" Level="1" />
  <!-- aplikacni namespace -->
  <Namespace Name="%NEW_NS"
Create="yes" Code="%NEW_NS" Data="%NEW_NS"
Ensemble="0" />
  <Configuration>
  <Database name="%NEW_NS"
Create="yes" Dir="%MGRDIR" %NEW_NS" PublicPermissions="" Resource="%DB_%NEW_NS" />
  <Var Name="URL"
Value="/csp/##{$ZCVT("%NEW_NS","L")}' />
  <Var Name="DIR"
Dir="%CSPDIR" ##{$ZCVT("%NEW_NS","L")}' />
  <CSPApplication Url="%URL" Directory="%DIR"
Resource=""
Description="%NEW_NS CSP Interface"
Recurse="true"

CookiePath="%URL" AuthenticationMethods="32" />
  <Configuration>
  </Namespace>
  <Namespace Name="%NEW_NS" Create="no"
Ensemble="0" />
  <!-- Nastaveni databaze CACHELIB -->
  <If Condition="%SET_CACHELIBS">
  <Log Topic="KUTAC.Demo"
Text="Kutac.Installer.SetCACHELIB (0)" Level="0" />
  <Invoke Class="Kutac.Demo" Method="SetCACHELIB" CheckStatus="true" />
  <Arg Value="0" />
  </Invoke>
  </If>
  <!-- Nastaveni lokalu -->
  <If Condition="%SET_LOCALE">
  <Log Topic="KUTAC.Demo"
Text="Kutac.Installer.SetLocale (%SOURCE_PATH) %FILE_LOCALE" Level="0" />
  <Invoke Class="Kutac.Demo" Method="SetLOCALE" CheckStatus="true" />
  <Arg Value="%SOURCE_PATH" %FILE_LOCALE" />
  </Invoke>
}
```

V praxi pri tvorbe zložitejších inštaláčnych schém najskôr zistíte, že hoci značky XML vám dávajú pomerne veľkú voľnosť, aj tak pre niektoré úlohy nestačia (napr. definovanie používateľských účtov a rolí, nastavenie NLS), takže asi často využijete prvky <Import> a <Invoke>.

Testovanie a ladenie obsahu triedy: Je samozrejme, že trieda s takýmito zásadnými schopnosťami musí mať možnosť byť pred nasadením otestovaná. Na testovanie môžete použiť príkazový riadok Caché, z ktorého spustíte metódu `setup ()` buď bez odovzdaných vstupných parametrov, alebo s nimi. Hoci takto:

```
%SYS> Set vars(SourceDir)="c:\kutac\installer\cache2009"
%SYS> Set vars(Updated)="Yes"
%SYS> Do ##class(KUTAC.Demo).setup(.vars,3)
```

Na to, aby inštaláčny skript Caché (na ľubovoľnom operačnom systéme) zavolať váš inštalátor,

musíte triedu exportovať zo Studia do súboru s názvom `DefaultInstallerClass.xml` v inštaláčnom adresári Caché. Ak chcete pri inštalácii použiť alternatívny názov súboru s inštaláčnou triedou alebo odovzdať parametre získané z operačného systému pri práve vykonávanej inštalácii, musíte inštaláciu Caché vykonať z príkazového riadka, a nie dvojklikom na ikonu inštaláčného súboru (iba Windows). Tu je alternatívna inštalácia pod Windows:

```
setup_cache.exe
INSTALLERMANIFEST="c:\kutac\installer\MyInstaller.xml"
INSTALLERMANIFESTPARAMS="SourceDir=c:\mysourcedir,Updated=Yes"
A zde pro Unix/Linux:
ISC_INSTALLER_MANIFEST=c:\MyStuff\MyInstaller.xml
ISC_INSTALLER_PARAMETERS=SourceDir=c:\mysourcedir,Updated=Yes
cinstall
```

Ako vidíte, unixové systémy využívajú premenenné prostredia miesto argumentov príkazu `cinstall`. Tento príspevok je vzhľadom na rozsiahlosť témy len veľmi stručný výrez a nezahŕňa zďaleka všetky elementy a možnosti inštalátora. Jeho cieľom je uviesť čitateľa do témy a naznačiť, kadiaľ sa možno uberať pri zostavovaní vlastnej inštaláčnej triedy.

Ak inštalujete na Windows, môžete ešte sami pokračovať tým, že pomocou nástrojov typu `InstallShield` alebo `Orca` si sami upravíte používateľské rozhranie inštalátora Caché, napríklad môžete vybrať niektoré dialógy a nahradiť ich východiskovými hodnotami či, naopak, pridať vlastné dialógy a inštalátor úplne upraviť na svoj obraz. Toto však značne presahuje rozsah článku a prípadné pokusy nechám úplne na vás.



DAN KUTÁČ, InterSystems

V nasledujúcej časti sa budeme venovať dynamickému odbaveniu vlastností a metód tried.

Ďalšie zaujímavé diely seriálu *Tipy a triky s Caché*, ktoré vychádzajú už od roku 2005, nájdete na stiahnutie tu: <http://www.intersystems.com/education/university/serialy/typy-trikyCache.html>