

Ant aneb „Ferda Mravenec, práce všeho druhu“

Nástroj na sestavování projektů (aplikací) podobný programu make, který se používá u programů v C či C++. Program Ant je volně k dispozici (tzv. The Apache Software License) na <http://jakarta.apache.org/>. Program je napsaný v jazyce Java a používá se v mnoha projektech realizovaných v Javě (např. tomcat, JDOM, jBOSS,).

V porovnání s programem make má Ant následující výhody:

- je nezávislý na platformě (pokud se nevyvolávají příkazy operačního systému, což obvykle není potřeba na rozdíl od programu make),
- má přehlednější syntaxi zapsanou v souboru s XML formátem,
- při sestavování aplikací v Javě je rychlejší než make,
- obsahuje specifické konstrukce používané pro sestavování aplikací v Javě (podpora jar souborů, podpora EJB, generování dokumentace javadoc, ...)

Ant se obvykle spouští z příkazové řádky, je možné ho i integrovat do mnoha vývojových nástrojů, např.:

- JBuilder
- Visual Age
- NetBeans, Forte for Java
- JEdit

Jak Ant pracuje?

Každý projekt má vlastní sestavovací (build) soubor, pro který se obvykle používá jméno **build.xml**, neboť tento název je defaultní a tudíž se nemusí specifikovat na příkazové řádce (parametr *-buildfile soubor*). Každý sestavovací soubor obsahuje jednotlivé **cíle (targets)**, které odpovídají jednotlivým aktivitám při sestavování projektu – např. překlad či vytvoření distribuce. Cíl, který se má realizovat se specifikuje na příkazové řádce – pokud se cíl neuvede, provede se defaultní cíl zapsaný v sestavovacím souboru. U každého cíle může být specifikováno, na kterých cílech je závislý (např. vytvoření distribuce bude závislé na překladu).

Každý cíl se provádí maximálně jednou za jedno vyvolání příkazu Ant. V případě, že vznikne chyba, přeruší Ant svoji činnost a nepokračuje s dalšími úlohami a cíli.

Každý cíl se skládá z jednotlivých úloh, které se provádějí sekvenčně za sebou. Některé úlohy se provádějí pouze v případě, že je to potřeba – např. zdrojový soubor se překládá pouze v případě, že je novější než odpovídající soubor .class

Jednoduchý příklad

V tomto příkladu předpokládáme následující adresářovou strukturu:

kořenový adresář projektu	<i>soubor build.xml a obvykle soubor README,</i>
- adresář src	<i>v tomto adresáři a podadresářích (dle balíků, package) jsou zdrojové texty jednotlivých tříd,</i>
- adresář lib	<i>v tomto adresáři jsou knihovny .jar používané v projektu,</i>
- adresář classes	<i>přeložené třídy (soubory .class),</i>
- adresář dist	<i>do tohoto adresáře se vytváří distribuce (obvykle .jar soubor),</i>
- adresář doc	<i>adresář s dokumentací, obvykle obsahuje podadresář api s dokumentací vygenerované pomocí javadoc,</i>

Ukázka souboru build.xml následuje:

```
<project name="CestovniKancelar" default="dist" basedir=". ">

  <!-- set global properties for this build -->
  <property name="src" value="src"/>
  <property name="build" value="classes"/>
  <property name="dist" value="dist"/>

  <target name="init"
    description="Create the build directory structure">
    <mkdir dir="${build}"/>
  </target>

  <target name="compile" depends="init"
    description="Compile the java code from ${src} into ${build}">
    <javac srcdir="${src}"
      destdir="${build}"/>
  </target>

  <target name="dist" depends="compile"
    description="Create the distribution jar file">
    <mkdir dir="${dist}/lib"/>
    <jar jarfile="${dist}/lib/CestovniKancelar.jar" >
      <fileset dir="${build}"/>
    </jar>
  </target>

  <target name="run" depends="dist"
    description="Run CestovniKancelar application">
    <java classpath="${dist}/lib/CestovniKancelar.jar"
      classname="CestovniKancelar"
      fork="yes"/>
  </target>

  <target name="clean"
    description="Delete the ${build} and ${dist} directory trees">
    <delete dir="${build}"/>
    <delete dir="${dist}"/>
  </target>
</project>
```

Popis částí souboru build.xml

V sestavovacím souboru je vždy popsán jeden projekt pomocí tagu <project>. Tento tag obvykle obsahuje tyto parametry:

name – jméno projektu
default – defaultní cíl, tj. ten ke kterému se Ant snaží dojít pokud neuvedete při spouštění jiný.

basedir – základní adresář, odkud vycházejí všechny cesty používané při popisu sestavování.

Uvnitř projektu jsou definovány globální parametry projektu (tag <property>) a jednotlivé cíle (tag <target>). U globálního parametru se vždy definuje jméno (parametr name) a hodnota (parametr value). Počet a typ parametrů u jednotlivých cílů závisí na typu cíle. Parametry jednotlivých cílů jsou popsány v dokumentaci.

Používání programu ant

Ant se spouští z dávkového souboru – příkazem ant. Obvykle se spouští v adresáři, ve kterém je sestavovací soubor a jako parametr se zadává název cíle, který se má provést (vždy se provedou i všechny cíle, na kterých je tento cíl závislý). Lze zadat i různé parametry – úplný seznam lze získat po zadání příkazu ant -help.

Následující výstup je získán po zadání příkazu ant v situaci, kdy se změnil jeden zdrojový soubor. Ve výstupu jsou vidět jak všechny realizované cíle, tak všechny realizované úlohy v rámci jednotlivých cílů (vypisují se pouze ty úlohy, které provedou nějakou akci – v následujícím výpisu není např. uvedena úloha mkdir v rámci cíle init, neboť příslušný adresář již existuje).

```
Buildfile: build.xml

init:

compile:
    [javac] Compiling 1 source file to C:\cestovka2\build

dist:
    [jar] Building jar: C:\cestovka2\dist\lib\CestovniKancelar.jar

BUILD SUCCESSFUL

Total time: 14 seconds
```

Spuštění programu ant s parametrem -projecthelp vypíše jednotlivé úlohy definované v buildfile. Následuje ukázka výstupu:

```
Buildfile: build.xml
Default target:

    compile  Compile the java code from ${src} into ${build}

Main targets:

clean      Delete the ${build} and ${dist} directory trees
compile    Compile the java code from ${src} into ${build}
init       Create the build directory structure
dist       Create the distribution jar file
run        Run CestovniKancelar application
```

Další časté cíle

Vedle cílů z ukázkového sestavovacího souboru se často objevují i následující cíle:

- generování dokumentace javadoc:

```
<property name="docDir" value="doc" />
<target name="javadoc" depends="compile"
  description="generate javadoc from all .java files">
  <delete dir="${docDir}/api"/>
  <mkdir dir="${docDir}/api"/>
  <javadoc sourcepath="${src}"
    destdir="${docDir}/api"
    classpath="${build}"
    sourcefiles="${src}/*.java"/>
</target>
```

Přehled úloh programu ant

Následující přehled obsahuje většinu úloh, které jsou k dispozici v základní instalaci programu ant.

Ant	provedení cíle v jiném build souboru, používá se v případě, že existují relativně samostatné subprojekty
AntStructure	vygeneruje DTD soubor, který obsahuje všechny známé úlohy
Apply	spustí příkaz z operačního systému,
Available	nastaví specifikovanou proměnnou (property) na true, pokud je dostupný zadaný soubor, či zadaná třída či zadaný zdroj JVM
Chmod	mění přístupová práva k souboru (v Unixu)
Condition	nastaví proměnnou na základě podmínky, je to zobecnění úlohy Available
Copy	kopíruje soubory či adresáře
CVS	spolupracuje se systémem pro správu verzí (CVS)
Delete	vymaže jednotlivé soubory či celé adresáře včetně podadresářů
Ear	vytváří tzv. Enterprise application archive (ear), obsahující .jar a .war soubory
Echo	vypíše text na standardní výstup či do souboru
Exec	spustí příkaz operačního systému
Fail	ukončí sestavování s chybovou hláškou
GenKey	generuje klíč pro digitální podpis
Get	zkopíruje soubor ze zadaného URL
Gzip, Gunzip	komprimování, dekomprimování souboru
Jar	vytvoření archivu .jar
Java	spustí aplikaci v Javě
Javac	přeloží zdrojový soubor programu v Javě do .class
Javadoc	vygeneruje dokumentaci v HTML formátu ze zdrojových souborů
Mail	odešle zprávu elektronickou poštou
Mkdir	vytvoří adresář(e),
Move	přesune/přejmenuje soubor(y),
Parallel	umožňuje využít více vláken pro spouštění úloh, používá se např. pro testování serverů (server je v jednom vlákně, test se spustí ve druhém

	vlákně)
Patch	umožňuje aplikovat diff soubory na zdrojové texty
Pathconvert	upravuje jména souborů do formátu vhodného pro aktuální platformu
Property	nastavuje vnitřní proměnné, které lze použít v dalších úlohách, proměnné lze též načíst ze souboru
Record	zapisuje výstup (log) z programu ant do souboru
Replace	zamění výskyty řetězce za jiný v zadaném souboru
Rmic	vyvolá rmic překladač pro vytvoření stubů a skeletonů pro třídy RMI
SignJar	doplní digitální podpis k jar souboru,
Sleep	čeká určitou dobu
SQL	vykonává SQL příkazy vůči databázi
Style	aplikuje XSLT předpisy na XML soubory a vytvoří příslušné výstupy
Tar	vytváření archivů tar
Taskdef	umožňuje začlenit vlastní úlohy (specifikuje se jméno třídy, která se má vykonat)
Touch	vytvoří prázdný soubor pokud neexistuje, nastaví čas modifikace souboru na aktuální (obdoba příkazu touch v unixu)
Tstamp	vytvoří vnitřní proměnné programu ant, které obsahují údaje o aktuálním čase,
Unjar	rozbalení archivu jar
Untar	rozbalení archivu tar
Unwar	rozbalení archivu war
Unzip	rozbaluje ZIP archivy
Uptodate	nastavuje vnitřní proměnné programu ant na základě porovnání aktuálnosti dvou souborů
War	vytváří Web Application Archive z jednotlivých souborů
Zip	vytváří ZIP archivy

Součástí distribuce programu ant je též archiv tzv. volitelných úloh, z nichž některé jsou uvedeny v následujícím přehledu. Pro spuštění některých z nich je potřeba doinstalovat ještě dodatečné programy (např. junit.jar pro testy JUnit).

Cab	vytváří archivy CAB používané firmou Microsoft
.EJB úlohy	úlohy pro vytváření Enterprise Java Beans
.NET úlohy	podpora C# a dalších .NET technologií
FTP	umožňuje pracovat se soubory na ftp serveru
JDepend	spouští nástroj JDepend nad zdrojovými programy pro měření kvality návrhu a závislostí mezi balíky,
Jlink	umožňuje vytvářet archivy ze souborů v různých archivech,
JUnit	spouští testy vytvořené pomocí JUnit
JUnitReport	generuje HTML výstup z testů pomocí JUnit
MimeMail	umožňuje odesílat el. poštou soubory
PropertyFile	umožňuje upravovat parametrické soubory Javy (property files)
Rpm	vytváří .rpm soubory používané v distribucích v některých verzích Linuxu
Sound	podpora pro přehrávání zvukových souborů
Telnet	podpora protokolu telnet (přihlašování na vzdálených počítačích),
XmlValidate	kontroluje, zda jsou XML soubory v pořádku (odpovídají .xsl souborům) či aspoň správně naformátované