

Parent Pom

# Table of Contents

1. POM struktúra .....	2
1.1. parent-pom .....	2
1.2. parent-icellmobilsoft-pom .....	2
1.3. parent-oss-pom .....	3
2. Releaselési folyamat .....	4
2.1. Deploy .....	4
2.2. Release .....	4
2.2.1. #1 Ellenőrzés .....	4
2.2.2. #2 Tisztítás .....	5
2.2.3. #3 SCM verzió módosítás .....	5
2.2.4. #4 Nexus feltöltés .....	6
2.3. Release problémák .....	6
3. Migrációs leírások .....	7
3.1. v1.1.0 → v1.2.0 .....	7
3.1.1. Újdonságok .....	7
3.2. v1.2.0 → v1.3.0 .....	8
3.2.1. Újdonságok .....	8
3.3. v1.3.0 → v1.4.0 .....	8
3.3.1. Újdonságok .....	8
3.4. v1.4.0 → v1.5.0 .....	9
3.4.1. Újdonságok .....	9
3.4.2. Bugfix-ek .....	10

## Java projektek fő pom.xml őse.

Általános maven fő pom.xml, amit minden i-Cell Mobilsoft Zrt. projekt tartalmaz.

A projekt arra célra készült hogy i-Cell Mobilsoft által készített projektek azonos általános beállításokkal működjenek.

Összegyűjti a közös beállításokat és azonosítókat, így elkerülhető a fejlesztői beállítások összeakadása és kisebb lehet a projekt fő pom.xml mérete ahol csak általános beállítások vannak. Emellett segít frissen tartani a plugineket, bugok esetén közös workaround megoldásokat és céges azonosításokat.

### *Minimum feltételek*

- Java 8+
- Maven 3.3.0+

# Chapter 1. POM struktúra

A projekt tartalmaz több pom.xml fájlt, mely mindig célzottabban közelíti meg a beállításokat.

## 1.1. parent-pom

Ez a legfelső pom.xml fájl. Ebből indul ki minden többi pom.xml.

*parent-pom.xml*

```
<groupId>hu.icellmobilsoft.pom</groupId>
<artifactId>parent-pom</artifactId>
```

Tartalma csak teljesen általános elemekből áll, icellmobilsoft cégre utaló beállítások:

*Beállítások:*

- UTF-8 karakter kódolás
- dátum formátum
- organization
- developers

*Definíciók:*

- maven plugin:
  - verziók
  - workaroundok
  - általános beállítások
- maven verzió minimum követelmény

*Profilok:*

- jdk8 - aktiválódik hogyha JDK 8 -as környezettel kompiláljuk a projektet
- jdk11 - aktiválódik hogyha JDK 11 -es környezettel kompiláljuk a projektet
- jdk17 - aktiválódik hogyha JDK 17 -es környezettel kompiláljuk a projektet
- jdk21 - aktiválódik hogyha JDK 21 -es környezettel kompiláljuk a projektet
- parent-pom-release - kézzel kell aktiválni hogyha a 'parent-pom'-ból új release-t készítünk

## 1.2. parent-icellmobilsoft-pom

Ez egy 'gyerek' pom, ami örökli a parent-pom összes beállításait, és kiegészíti azt az icellmobilsoft domain specifikus beállításokkal.

*icellmobilsoft/pom.xml*

```
<groupId>hu.icellmobilsoft.pom</groupId>
```

```
<artifactId>parent-icellmobilsoft-pom</artifactId>
```

Tartalma minimális, csak kiegészíti a fő pom.xml-t:

*Beállítások:*

- Repository (plugin repository is):
  - icellmobilsoft nexus
  - jboss maven
  - maven central
- Distribution Management
  - Snapshot tárhely
  - Release tárhely

## 1.3. parent-oss-pom

Ez egy 'gyerek' pom, ami öröklí a parent-pom összes beállításait, és kiegészíti azt az icellmobilsoft open source specifikus beállításokkal.

*oss/pom.xml*

```
<groupId>hu.icellmobilsoft.pom</groupId>  
<artifactId>parent-oss-pom</artifactId>
```

Tartalma az aktuális public maven central publikálásához szükséges beállításait célozza meg. Ezek közé tartoznak a következők:

*Beállítások:*

- license
- pluginManagement
- release profil
- distributionManagement
- gpg aláírás
- sonatype public nexus plugin kezelés

# Chapter 2. Releaseelési folyamat

Mivel a projekt őrsként szolgál a többi projekten, így azt figyelembe véve nem "klasszikus" release maven beállításokkal kell ellátni. Az ok hogy ennek a projekt beállításait leörökölné a cél projekt, és az nagyon nem kívánt eseményekhez vezetne. Tehát minden release beállítást egy **parent-pom-release** maven profile alá kell rejtteni.

## 2.1. Deploy

Hogyha egy SNAPSHOT verziót akarunk kiadni a céges nexus szerverre, a következő parancsokat kell használni:

```
mvn -B -Pparent-pom-release clean deploy source:jar
```

- *mvn* maven executable parancs
- *-B* batch futás - ne kérjen be konzolból megerősítéseket a maven
- *-Pparent-pom-release* aktiválja a 'parent-pom-release' maven profilt
- *clean* kitörli a *\*/target* mappákat
- *deploy* deploy parancs, ami röviden fő lépésekben a következőket hajtja végre:
  - kompilálja a kódot
  - jar, war, pom csomagokat készít
  - elkészült jar, war, pom csomagokat bemásolja a lokális *.m2* repositoryba
  - elkészült jar, war, pom csomagokat feltolja a definiált *snapshotRepository* helyre, ami a céges nexust jelenti
- *source:jar* ez egy kapcsoló hogy az elkészült jar, war, pom csomagokkal együtt a forráskódokat is másolja/játssza fel a nexusba

## 2.2. Release

Hogyha egy fix verziót akarunk kiadni SNAPSHOT jelölés nélkül. Maga a tiszta release folyamat 4 lépésből áll:

### 2.2.1. #1 Ellenőrzés

Ez a lépés szolgál arra, hogy technikai oldalról ellenőrizzük le, hogy a maven minden beállítása és release követelménye megvan. Röviden a következő történik:

- minden változás commitolva van
- fordítható a kód
- tesztek sikeresen le tudnak futni
- SCM beállítások megvannak

- sikeres a verzióemelés próbálkozás

```
mvn -B -Pparent-pom-release release:prepare -DdryRun=true
```

- *mvn* maven executable parancs
- *-B* batch futás - ne kérjen be konzolból megerősítéseket a maven
- *-Pparent-pom-release* aktiválja a 'parent-pom-release' maven profilt
- *release:prepare* maven-release-plugin része, indítja az SCM oldali verzió kiadását
- **-DdryRun=true** kapcsoló, hogy a verzió kiadás csak "virtuális" legyen, tehát hogy csak ellenőrzés céljából indítjuk

Ennek a lépésnek le kell futnia minden hibajelentés nélkül. Esetleges hibákat javítani kell, mert a valós verziókiadás sem fog lefutni.

### 2.2.2. #2 Tisztítás

A próba során létrejött temp fájlokat töröljük, ne zavarjanak a valódi verziókiadásnál.

```
mvn -B release:clean
```

- *mvn* maven executable parancs
- *-B* batch futás - ne kérjen be konzolból megerősítéseket a maven
- *release:clean* maven-release-plugin része, tisztítja a **target** könyvtárat és törli a temporális release fájlokat

### 2.2.3. #3 SCM verzió módosítás

Ha sikeres az #1 és #2 lépés, valós SCM verzió kiadását indítjuk

```
mvn -B -Pparent-pom-release release:prepare
```

- *mvn* maven executable parancs
- *-B* batch futás - ne kérjen be konzolból megerősítéseket a maven
- *-Pparent-pom-release* aktiválja a 'parent-pom-release' maven profilt
- *release:prepare* maven-release-plugin része, indítja az SCM oldali verzió kiadását

Ez a lépés gyakran hibára fut első futásnál (új környezetben), melynek a legfőbb okai ezek szoktak lenni:

- környezeti beállítások nincsenek felkészítve konzolos maven, git és java futásra
- lokális **settings.xml** fájlban nincsenek rögzítve a **pom.xml**-ben szereplő szerverek autentikációs adatai

- a java verzióban nincsenek benne a szerverek tanúsítványai (ez szok lenni a legnagyobb szívás)

A parancs röviden a következő lépéseket csinálja:

- #1 pontban leírt ellenőrzések
- "-SNAPSHOT" levágása a verziószámból
- compile
- lokális `.m2` könyvtár feltöltése
- commit
- tag létrehozása
- verziószám növelése + "-SNAPSHOT" kiegészítés
- compile
- commit

A parancs lefutása után a lokális fájlokban a verziószám növelve és commitolva lesz. A fájlok között megtalálhatók a **release temporális fájlok, melyeket nem szabad kitörölni!**

#### 2.2.4. #4 Nexus feltöltés

Ha elkészülünk a #3 ponttal, akkor már csak a csomagok elkészítése marad hátra. Fontos hogy a #3 pont temporális fájljai ott legyenek a projektben!

```
mvn -B -Pparent-pom-release release:perform
```

- *mvn* maven executable parancs
- *-B* batch futás - ne kérjen be konzolból megerősítéseket a maven
- *-Pparent-pom-release* aktiválja a 'parent-pom-release' maven profilt
- *release:perform* maven-release-plugin része, indítja a nexus oldali verzió kiadását:
  - leszedi a temporális fájlokban jelölt SCM TAG-el jelölt forráskódot
  - compile, jar, war, pom, source csomagok létrehozása
  - feltölti a nexusba

A leggyakoribb hiba az szokott lenni, hogy a nexus szerveren SSL tanúsítvány nincs benne a java verzióinkban, ezért a futás hibára fut.

## 2.3. Release problémák

Sajnos nem fog mindig sikerrrel végződni a fenti 4 lépés, ilyenkor sokszor javításokat kell elvégezni. A leggyakoribb esetek és azok javítási lépései a következő pontokban vannak kifejtve.



# Chapter 3. Migrációs leírások

## 3.1. v1.1.0 → v1.2.0

parent-pom v1.1.0 → v1.2.0 migrációs leírás, újdonságok, változások leírása

### 3.1.1. Újdonságok

#### parent-icellmobilsoft-pom

Fel lett cserélve a repository sorrendje, hogy a `icell-public-repository` legyen az első helyen, ezzel optimalizálva a CI/CD belső folyamatainkat.

#### Átállás

A változtatások nem eredményeznek átállási munkálatokat, visszafelé kompatibilis.

#### parent-pom

*Plugin frissítések:*

- maven-gpg-plugin 1.6 → [3.0.1](#)
- maven-antrun-plugin 1.8 → [3.0.0](#)
- maven-assembly-plugin 3.1.1 → [3.3.0](#)
- maven-dependency-plugin 3.1.1 → [3.2.0](#)
- maven-release-plugin 2.5.3 → [3.0.0-M4](#)
- maven-resources-plugin 3.1.0 → [3.2.0](#)
- maven-surefire-plugin 3.0.0-M3 → [3.0.0-M5](#)
- maven-failsafe-plugin 3.0.0-M3 → [3.0.0-M5](#)
- maven-jar-plugin 3.1.1 → [3.2.0](#)
- maven-war-plugin 3.2.2 → [3.3.2](#)
- maven-site-plugin 3.7.1 → [3.9.1](#)
- maven-javadoc-plugin 3.1.0 → [3.3.1](#)
- maven-source-plugin 3.0.1 → [3.2.1](#)
- build-helper-maven-plugin 3.0.0 → [3.2.0](#)
- maven-enforcer-plugin 3.0.0-M2 → [3.0.0](#)
- asciidoctor-maven-plugin 1.6.0 → [2.2.1](#) Itt található 1.6.0 → v2 váltás migrációs leírás
- asciidoctorj-diagram 2.0.1 → [2.2.1](#)

#### Átállás

A pluginek többsége java 7 és 8 minimum requirement-esek, tehát minimum Java8 szükséges a projekten.

Általánosan nem várható probléma, esetleg az asciidoc plugin használatánál, de valószínű az sem critical szint.

## 3.2. v1.2.0 → v1.3.0

parent-pom v1.2.0 → v1.3.0 migrációs leírás, újdonságok, változások leírása

### 3.2.1. Újdonságok

#### parent-icellmobilsoft-pom

Fel lett cserélve a plugin repository sorrendje, hogy a **icell-public-repository** legyen az első helyen, ezzel optimalizálva a CI/CD belső folyamatainkat.

#### Átállás

A változtatások nem eredményeznek átállási munkákat, visszafelé kompatibilis.

#### github workflow

Complie job matrix konfigurációt kapott, amiben az operációs rendszer és a 11 valamint a 17-es java verzió van minden lehetséges variációban megadva a buildelési folyamatnak.

Dokumentum generálásnál a java verzió 17-re lett emelve így az asciidoctor-maven-plugin verziója 2.2.2 verzióra lett emelve.

Manual release indításához czenczl felhasználó is bekerült.

#### Átállás

A változtatások nem eredményeznek átállási munkákat, visszafelé kompatibilis.

## 3.3. v1.3.0 → v1.4.0

parent-pom v1.3.0 → v1.4.0 migrációs leírás, újdonságok, változások leírása

### 3.3.1. Újdonságok

Bekerült a Java 21 támogatása a profilokhoz.

#### parent-pom

Új plugin:

- jacoco-maven-plugin 0.8.11 - általános beállításokkal a surefire és failsafe tesztekhez

Plugin frissítések:

- Bump maven-antrun-plugin 3.0.0 → 3.1.0
- Bump maven-assembly-plugin 3.3.0 → 3.6.0
- Bump maven-dependency-plugin 3.2.0 → 3.6.0

- Bump maven-release-plugin 3.0.0-M4 → 3.0.1
- Bump maven-compiler-plugin 3.8.1 → 3.11.0
- Bump maven-clean-plugin 3.1.0 → 3.3.1
- Bump maven-install-plugin 3.0.0-M1 → 3.1.1
- Bump maven-resources-plugin 3.2.0 → 3.3.1
- Bump maven-surefire-plugin 3.0.0-M5 → 3.1.2
- Bump maven-failsafe-plugin 3.0.0-M5 → 3.1.2
- Bump maven-deploy-plugin 3.0.0-M1 → 3.1.1
- Bump buildnumber-maven-plugin 1.4 → 3.2.0
- Bump maven-jar-plugin 3.2.0 → 3.3.0
- Bump maven-war-plugin 3.3.2 → 3.4.0
- Bump maven-site-plugin 3.9.1 → 3.12.1
- Bump maven-javadoc-plugin 3.3.1 → 3.6.0
- Bump maven-source-plugin 3.2.1 → 3.3.0
- Bump build-helper-maven-plugin 3.2.0 → 3.4.0
- Bump maven-enforcer-plugin 3.0.0 → 3.4.1
  - configuration **requireMavenVersion 3.3.0 → 3.8.1**
- Bump asciidoctor-maven-plugin 2.2.1 → 2.2.4
- Bump asciidoctorj-diagram 2.2.1 → 2.2.13
- Bump maven-gpg-plugin 3.0.1 → 3.1.0
- Bump nexus-staging-maven-plugin 1.6.8 → 1.6.13

## Átállás

A pluginek java 8 minimum requirement-esek. Minimum Maven függőségük 3.5.4 ami fel lett emelve 3.8.1 verzióra. Ez fel van készítve a 4+ könnyű átállásra és nagyban optimalizáva lett a paralel futás.

Tehát minimum Java8 és Maven 3.8.1+ szükséges a projekten.

## Átállás

A változtatások nem eredményeznek átállási munkálatokat, visszafelé kompatibilis.

## 3.4. v1.4.0 → v1.5.0

parent-pom v1.4.0 → v1.5.0 migrációs leírás, újdonságok, változások leírása

### 3.4.1. Újdonságok

## parent-pom

*Plugin csere:*

- org.sonatype.plugins:nexus-staging-maven-plugin 1.6.13 → org.sonatype.central:central-publishing-maven-plugin 0.8.0

### 3.4.2. Bugfix-ek

#### maven-deploy-plugin konfiguráció

- Jelenleg a maven-deploy-plugin 3.1.1-es verzióját használjuk. A 3.0.0-ás verzió óta a **uniqueVersion** konfigurációs paraméter teljesen megszüntetésre került, így mi is eltávolítottuk a konfigurációnkból. (<https://maven.apache.org/plugins-archives/maven-deploy-plugin-3.0.0/#major-version-upgrade-to-version-3-0-0>)

#### Átállítás

- Open Source Maven projektek a <https://github.com/i-Cell-Mobilsoft-Open-Source> alatt, akik a **parent-oss-pom**-ot használják frissíteniük kell Sonatype bejelentkezési adataikat és szerver azonosítóikat a GitHub workflow-kban és a settings.xml fájlban a következőképpen:
  - server-id: ossrh → central
  - secrets.OSS\_SONATYPE\_ORG\_USER → secrets.CENTRAL\_SONATYPE\_ORG\_USER
  - secrets.OSS\_SONATYPE\_ORG\_PASSWORD → secrets.CENTRAL\_SONATYPE\_ORG\_PASSWORD