



IN ALLESTIMENTO

Maven ed Eclipse

Nelle ultime release di Eclipse, l'integrazione con Maven è stata integrata direttamente nella IDE Java/J2EE dopo essere stata disponibile per qualche tempo come plugin di terze parti.

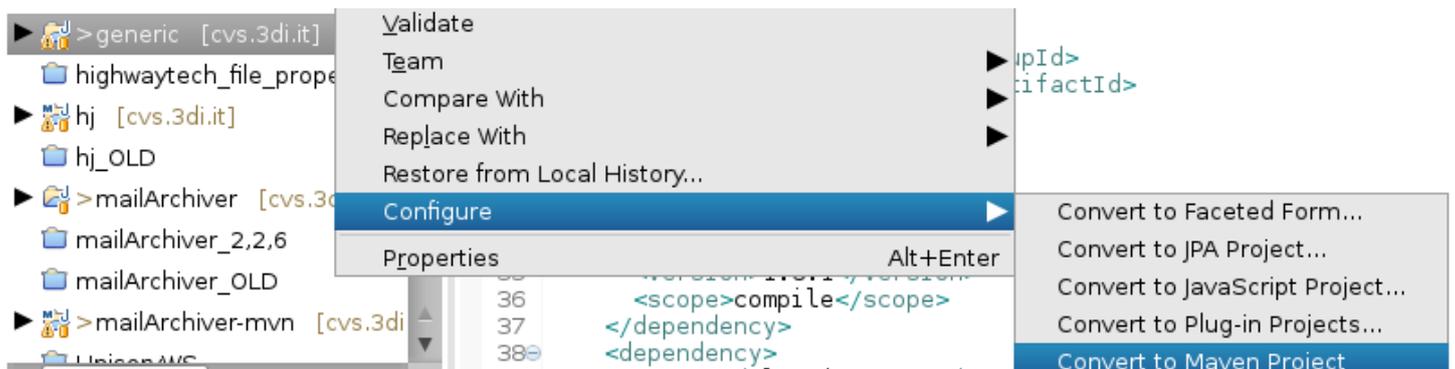
Il compito del plugin m2eclipse è quello di fare da ponte tra i due "mondi", poiché il layout dei progetti e dei sorgenti in maven ed Eclipse differiscono significativamente, come riepilogato nella seguente tabella:

	Maven	Eclipse
Directory sorgenti Java	src/main/java	src/
Directory risorse Java	src/main/resources	?
Directory sorgenti di test	src/test/java	?
Directory risorse per i test	src/test/resources	?
Directory di output dei binari	target/	build/
Layout gerarchia progetti	gerarchico	piatto

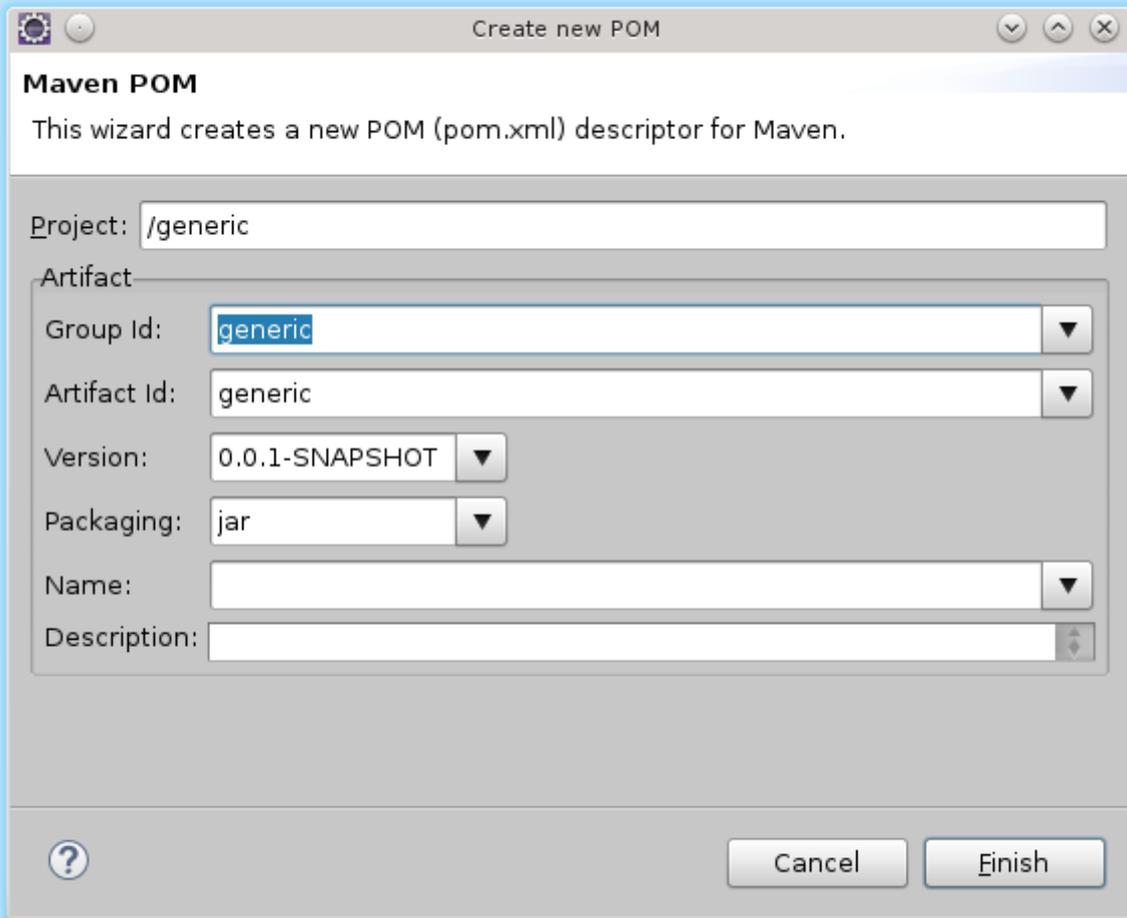
In fase di conversione di un progetto Java semplice in un progetto Maven all'interno di Eclipse, il plugin m2eclipse effettua gli opportuni override per indicare a maven dove trovare i vari tipi di sorgenti, le risorse ecc. Il pom.xml effettivo che ne risulta è quello (non modificabile) riportato nella tab "Effective POM", quando si apre un file pom.xml.

Conversione di un progetto semplice in un progetto Maven

Per convertire un normale progetto Java in un progetto Maven all'interno di Eclipse, è sufficiente cliccare col tasto destro sul progetto, selezionare **Configure** ⇒ **Convert to Maven project**:



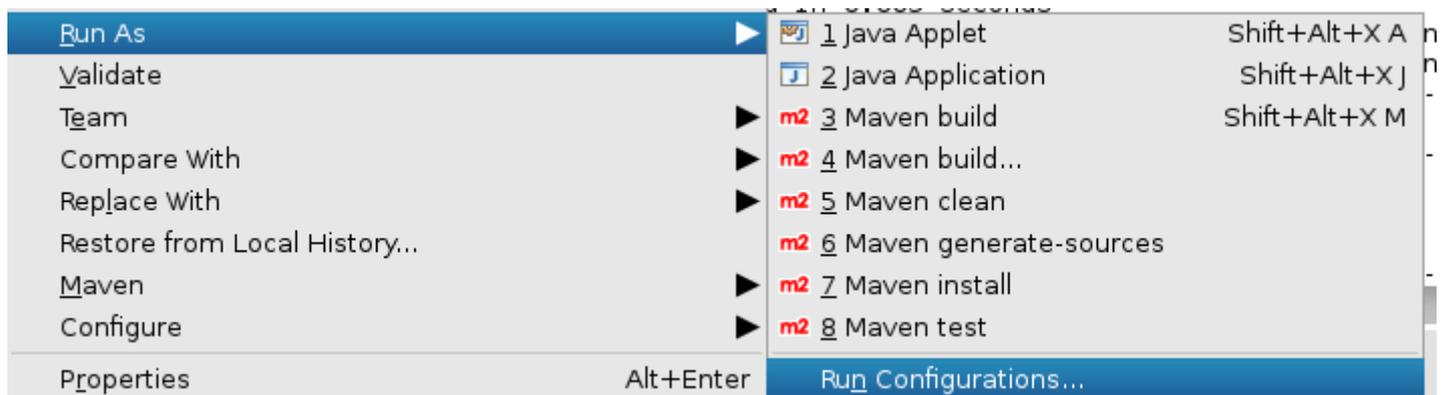
Verrà mostrata una finestra nella quale bisogna inserire i metadati indispensabili a Maven per identificare il progetto, ovvero il GroupID e l'Artifact Id, oltre al numero di versione, il tipo di packaging, un'eventuale nome e descrizione:



In seguito, verrà creato il pom.xml, nel quale sarà poi possibile specificare altre informazioni, quali dipendenze ecc. Si rimanda alla pagina [Introduzione a maven](#) per informazioni più specifiche al riguardo.

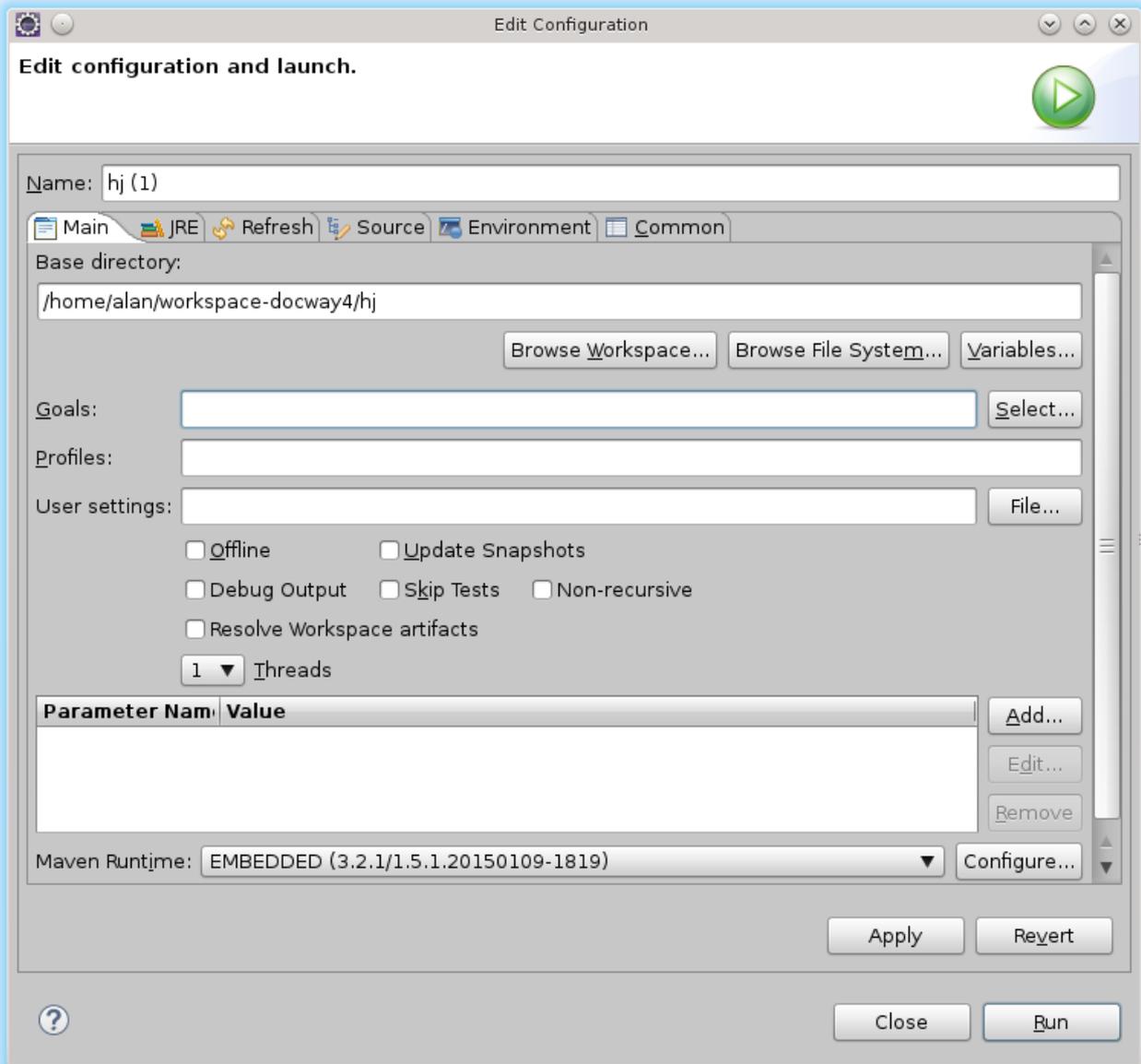
Esecuzione di target specifici della build

Una volta che il progetto è stato convertito in un progetto Maven, è possibile eseguire passi del build lifecycle semplicemente cliccando col tasto destro sul progetto, selezionando Run as ⇒ Maven *



Esaminiamo le opzioni:

- Maven build : esegue l'ultima Maven build definita
- Maven build... : permette di specificare quali goals eseguire, quali profili utilizzare ecc.



- Maven `clean` : esegue il task `clean`, che rimuove gli artefatti generati
- Maven `generate-sources` : qualora il progetto preveda un passaggio di generazione sorgenti (per esempio, generazione mediante `javacc`, `jaxb` ecc.), questo target genera i sorgenti utilizzati poi durante la build vera e propria
- **Maven `install`**: effettua la compilazione del progetto, l'esecuzione degli unit-test, la generazione dell'artefatto e l'installazione dello stesso nel proprio repository locale. **Generalmente è il target da utilizzare per effettuare le build per intero.**
- Maven `test` : esegue gli unit-test impostati

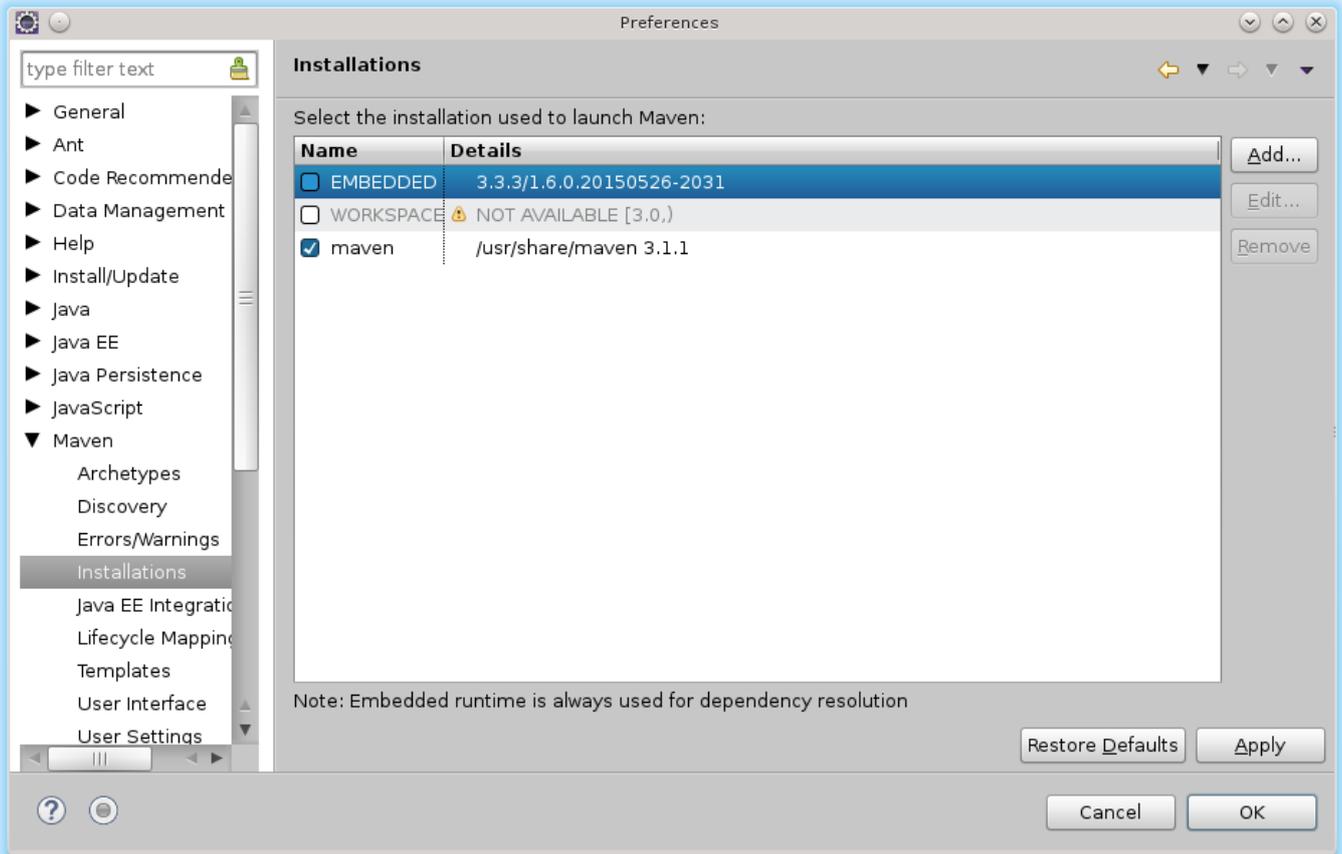
Note

Java < 1.7 ed uso di Eclipse Mars (4.5.0) o superiore



IMPORTANTE

Il plugin `m2eclipse` integrato in Eclipse Mars integra Maven 3.3.3 che risulta essere problematico per i progetti che devono compilare con una JDK < 1.7, in quanto esso stesso è stato compilato con Java 7 e non è eseguibile su un runtime inferiore come, ad esempio, Java 6. È sempre possibile specificare diversi runtime maven da utilizzare alla bisogna andando nelle preferenze di Eclipse e puntando ad una directory che contenga la root di installazione maven:



Per problemi di questo tipo, è possibile utilizzare Maven 3.1.1, che risulta ancora essere compatibile con Java 1.6 (disponibile anche su [\\STORAGE\RESOURCE\apache-maven](#)).