



PROGETTO GESTIONE DOCUMENTALE

INDEX

Installazione ECM Engine

Versione 1.0

STATO DELLE VARIAZIONI

VERSIONE	PARAGRAFO O PAGINA	DESCRIZIONE DELLA VARIAZIONE
1.0	Tutto il documento	Versione iniziale del documento

	<p style="text-align: center;">INDEX</p> <p style="text-align: center;">INSTALLAZIONE PIATTAFORMA</p>	<p>PGED-INDEX-ARC-v01</p> <p>Pag. 2 di 21</p>
---	---	---

INDICE

1.	Introduzione	3
1.1	Scopo del documento	3
1.2	Riferimenti	3
2.	Installazione ambiente	4
2.1	Java Development Kit	4
2.2	Jboss	4
2.3	Mysql	4
3.	Creazione del pacchetto di deploy	4
3.1	Scarico dei sorgenti (operazione necessaria solo per le realta' che creano i pacchetti dai sorgenti)	5
3.2	Parametri di compilazione (operazione necessaria solo per le realta' che creano i pacchetti dai sorgenti)	6
3.3	Parametri di configurazione della piattaforma (operazione necessaria solo per le realta' che creano i pacchetti dai sorgenti)	6
3.3.1	Web service basati su engine Axis 1.4 (opzionali)	11
3.4	Creazione ed installazione dei pacchetti da deployare (operazione necessaria sia per le realta' che creano i pacchetti sia per quelle che deployano il pacchetto scaricato)	12
3.5	Configurazione ambiente (operazione necessaria sia per le realta' che creano i pacchetti sia per quelle che deployano il pacchetto scaricato)	16
3.5.1	Creazione base/i dati	16
3.5.2	Configurazione logger applicativo (operazione necessaria sia per le realta' che creano i pacchetti sia per quelle che deployano il pacchetto scaricato)	17
3.6	Avvio di jboss (operazione necessaria sia per le realta' che creano i pacchetti sia per quelle che deployano il pacchetto scaricato)	19
3.7	Accesso client verso web service Axis 1.4 (opzionale)	19
4.	Test dei servizi pubblicati per i fruitori	19
4.1	Test unitari	19
4.1.1	Sviluppo dei test unitari	19
4.1.2	Creazione pacchetto dei test unitari	20

	<p style="text-align: center;">INDEX</p> <p style="text-align: center;">INSTALLAZIONE PIATTAFORMA</p>	<p>PGED-INDEX-ARC-v01</p> <p>Pag. 3 di 21</p>
---	---	---

1. Introduzione

1.1 Scopo del documento

Questo documento fornisce le regole per l'installazione della piattaforma ecmengine. L'installazione copre le due possibilità :

- realta' in cui si vogliano scaricare i sorgenti dal sito www.doqui.it, ricreare il pacchetto di deploy e deployarlo
- realta' in cui si voglia utilizzare il pacchetto scaricato dal sito www.doqui.it e deployarlo

1.2 Riferimenti

2. Installazione ambiente

2.1 Java Development Kit

Dev'essere stata presente il jdk1.5.0* o superiore; per controllare la presenza del jdk consigliato :

- lanciare dalla shell unix o dal command prompt di windows il comando `java -version`. l'output dovrà essere simile a quello della figura sottostante (sulla macchina nell'esempio e' presente un jdk 1.5.0_03)

```
java version "1.5.0_03"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_03-b07)
Java HotSpot(TM) Client VM (build 1.5.0_03-b07, mixed mode, sharing)
```

Figura 2.3 : Versione jdk presente nel classpath

2.2 Jboss

Dev'essere presente l'application server jboss (<http://www.jboss.org>).

La versione attuale di riferimento e' la 4.0.5.

2.3 Mysql

Dev'essere presente o visibile dalla macchina jboss almeno un utente su un'istanza mysql.

La versione di riferimento e' la 5.0.

E' necessario inserire nel classpath dell'istanza di jboss la libreria che permette la connessione alla base dati mysql dall'application server

Nella directory <jboss_home>/server/<partizione/istanza>/lib dev'essere presente la libreria riportata nella figura sottostante o versioni successiva di questa.

 mysql-connector-java-3.1.14-bin.jar	459,094	01/02/2008 11.16.03
---	---------	---------------------

Figura: 2.4 : libreria di connessione alla base dati

3. Creazione del pacchetto di deploy

La creazione del pacchetto di deploy prevede le seguenti azioni :

- scarico dei sorgenti dal repository centrale
- parametri di compilazione
- parametri di configurazione della piattaforma
- creazione del pacchetto di deploy

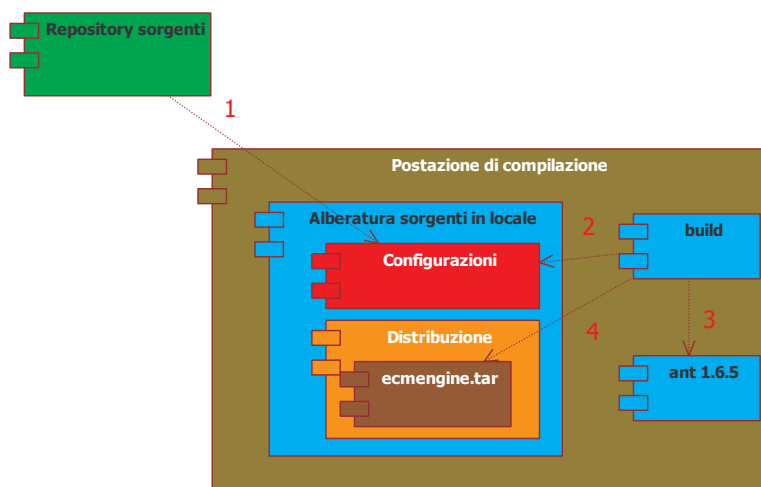


Figura 2.1 : Processo di scarico dei sorgenti e build

3.1 Scarico dei sorgenti (operazione necessaria solo per le realta' che creano i pacchetti dai sorgenti)

Il repository dei sorgenti rappresenta la posizione centrale in cui sono contenuti tutti i sorgenti, le configurazioni della piattaforma e la configurazioni di build.

Per la gestione dei sorgenti e' stato utilizzato subversion (<http://subversion.tigris.org/>), un prodotto open source che prevede lo scarico dei sorgenti su una postazione locale e la gestione di questi attraverso operazioni di lock, check-in e check-out.

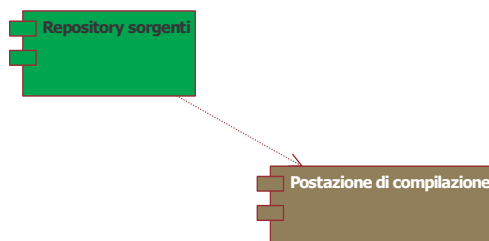


Figura 2.2 : Scarico dei sorgenti

Lo scarico in locale avviene attraverso un client installato sulla postazione di compilazione; si possono utilizzare come client di subversion

Tortoise (<http://tortoisesvn.tigris.org/>),
Rapidsvn (<http://rapidsvn.tigris.org/>)

il repository in locale della piattaforma ecmengine ha la seguente struttura

Name	Size	Type	Modified	Attributes	
.svn		Folder	01/02/2008 12.36.29	drwxr-xr-x	Directory di subversion
build		Folder	01/02/2008 14.35.47	drwxr-xr-x	Cartella per il build delle classi e la personalizzazione delle configurazioni
buildfiles		Folder	01/02/2008 12.35.46	drwxr-xr-x	Directory contenente le configurazioni modificabili in fase di creazione del pacchetto
conf		Folder	01/02/2008 12.36.09	drwxr-xr-x	Configurazioni della piattaforma
dist		Folder	01/02/2008 12.38.15	drwxr-xr-x	Pacchetti di distribuzione
docs		Folder	01/02/2008 12.36.28	drwxr-xr-x	Documentazione
lib		Folder	01/02/2008 12.37.35	drwxr-xr-x	Librerie di compilazione e/o presenti nel pacchetto
src		Folder	01/02/2008 12.36.28	drwxr-xr-x	Sorgenti
test		Folder	01/02/2008 12.35.45	drwxr-xr-x	Test junit
build.xml	95,393	XML Document	01/02/2008 12.36.29	-rw-r--r--	Build file

Figura 2.3 : Repository fisico locale

La medesima operazione di scarico dei sorgenti deve essere effettuata per la componente che gestisce la base dati (creazione delle tavole della base dati e installazione di una store procedure). Tale componente ha la medesima struttura fisica di quella presentata nella figura soprastante ma conterra' due script sql.

3.2 Parametri di compilazione (operazione necessaria solo per le realta' che creano i pacchetti dai sorgenti)

La compilazione avviene utilizzando il framework ant (<http://ant.apache.org/>) utilizzando i target contenuti nel file build.xml per la compilazione delle classi e/o la creazione del pacchetto di deploy.

Per la compilazione e la creazione del pacchetto di deploy e' necessario l'installazione del framework ant 1.6.5 o superiore a cui vanno aggiunte le librerie contrib nella directory

```
<installazione_ant>/lib
```

Libreria	Fonte
ant-contrib-1.0b3.jar	http://ant-contrib.sourceforge.net/
commons-collections-3.1.jar	http://commons.apache.org/collections/
commons-lang-2.0.jar	http://commons.apache.org/lang/
ivy-jdk1.3.1-1.0.0.jar	http://www.jaya.free.fr/ivy/doc/install.html
jakarta-oro-2.0.8.jar	http://jakarta.apache.org/oro/index.html
JeraAntTasks.jar	http://www.jera.com/tools/anttasks/
jsch-0.1.33.jar	http://sourceforge.net/projects/jsch/
xercesImpl.jar	http://xml.apache.org/xalan-j/downloads.html
xml-apis.jar	http://xml.apache.org/xalan-j/downloads.html

3.3 Parametri di configurazione della piattaforma (operazione necessaria solo per le realta' che creano i pacchetti dai sorgenti)

Nella directory buildfiles del repository dei sorgenti sono presenti i file di configurazione per i vari ambienti per cui puo' essere creato il pacchetto.

	<p style="text-align: center;">INDEX</p> <p style="text-align: center;">INSTALLAZIONE PIATTAFORMA</p>	<p>PGED-INDEX-ARC-v01</p> <p>Pag. 7 di 21</p>
---	---	---

Su ogni file di property vi sono i parametri che possono essere personalizzati, in modo da creare un pacchetto che possa rispondere alle necessita' di ogni iniziativa.

Parametro	Valore	Descrizione
target.dir.root	/ecmengine	directory in cui verranno inseriti i contenuti della piattaforma ecmengine
target.dir.index	/ecmengine	directory in cui verranno inseriti gli indici lucene della piattaforma ecmengine (il file system deve essere locale alla macchina su cui e' presente l'application server jboss)
target.conf.external.path	ecmengine/extension	posizione in cui verranno posizionate le estensioni della piattaforma (la directory e' relativa all'environment jboss, quindi fisicamente saranno posizionati nella directory <jboss_home>/server/<partizione o istanza>/conf/ecmengine/extension
target.db.url	jdbc:mysql://localhost:3306/ ecmdb	url della base dati per la configurazione con il single repository
target.db.usr	indexadmin	username della base dati per la configurazione con il single repository
target.db.pwd	indexadmin	password della base dati per la configurazione con il single repository
target.db.driver	org.gjt.mm.mysql.Driver	classe contenente il driver per la connessione alle base dati per la configurazione con il single repository
target.db.url.primary	jdbc:mysql://localhost:3306/ ecmdb	url della base dati primaria per la configurazione con il multi repository
target.db.usr.primary	indexadmin	username della base dati primaria per la configurazione con il multi repository



INDEX

INSTALLAZIONE PIATTAFORMA

PGED-INDEX-ARC-v01

Pag. 8 di 21

<code>target.db.pwd.primary</code>	<code>indexadmin</code>	password della base dati primaria per la configurazione con il multi repository
<code>target.db.driver.primary</code>	<code>org.gjt.mm.mysql.Driver</code>	classe contenente il driver per la connessione alle base dati primaria per la configurazione con il multi repository
<code>target.db.url.secondary</code>	<code>jdbc:mysql://localhost:3306/ ecmdbsecondary</code>	url della base dati secondaria per la configurazione con il multi repository
<code>target.db.usr.secondary</code>	<code>indexseco</code>	username della base dati secondaria per la configurazione con il multi repository
<code>target.db.pwd.secondary</code>	<code>indexseco</code>	password della base dati secondaria per la configurazione con il multi repository
<code>target.db.driver.secondary</code>	<code>org.gjt.mm.mysql.Driver</code>	classe contenente il driver per la connessione alle base dati primaria per la configurazione con il multi repository
<code>target.db.dialect</code>	<code>org.hibernate.dialect. MySQLInnoDBDialect</code>	Dialect utilizzato da Hibernate per la connessione alla/e basi dati mysql
<code>target.db.pool.initial</code>	<code>20</code>	Dimensione iniziale di connessioni del connection pool
<code>target.db.pool.max</code>	<code>40</code>	Dimensione massima di connessioni del connection pool
<code>target.db.pool.timeout</code>	<code>120</code>	Timeout per la connessione alla base dati
<code>target.application_server</code>	<code>jboss</code>	Application server su viene deployata la soluzione (al momento e' valida solo l'opzione jboss)
<code>target.cluster.clustered</code>	<code>false</code>	La piattaforma viene deployata in ambiente jboss su cui non e' gestito il cluster (possibili valori false o true)
<code>target.cluster.partition</code>	<code>defaultPartition</code>	Partizione dedicata al deploy del prodotto ecmengine (valido solo se valorizzata a true la property



INDEX

INSTALLAZIONE PIATTAFORMA

PGED-INDEX-ARC-v01

Pag. 9 di 21

		target.cluster.clustered)
target.cluster.nodes.master	node1	Nodo master nella configurazione jboss in cluster per la gestione dei job (al momento non presente)
target.cluster.nodes	node1,node2	Nodi presenti nella partizione del cluster
target.cluster.ehcache. multicast_group_address.ip	228.1.1.3	Indirizzo multicast (valido solo per la configurazione in cluster in single repository)
target.cluster.ehcache. multicast_group_address.port	11140	Porta di gruppo multicast (valido solo per la configurazione in cluster in single repository)
target.cluster.node1.ehcache. rmi_cache_manager_peer_ listener_factory.ip	0.0.0.0	Ip di binding per i cache listener per il nodo1 del cluster (valido solo per la configurazione in cluster)
target.cluster.node2.ehcache. rmi_cache_manager_peer_ listener_factory.ip	0.0.0.0	Ip di binding per i cache listener per il nodo2 del cluster (valido solo per la configurazione in cluster)
target.cluster.node1.ehcache. rmi_cache_manager_peer_ listener_factory.port	11130	Port di binding per i cache listener per il nodo1 del cluster (valido solo per la configurazione in cluster in single repository)
target.cluster.node2.ehcache. rmi_cache_manager_peer_ listener_factory.port	11130	Port di binding per i cache listener per il nodo2 del cluster (valido solo per la configurazione in cluster in single repository)
target.cluster.ehcache. multicast_group_address. port.primary	11140	Porta di gruppo multicast (valido solo per la configurazione in cluster in multi repository per il primary repository)
target.cluster.ehcache. multicast_group_address. port.secondary	11141	Porta di gruppo multicast (valido solo per la configurazione in cluster in multi repository per il secondary repository)



INDEX

INSTALLAZIONE PIATTAFORMA

PGED-INDEX-ARC-v01

Pag. 10 di 21

<code>target.cluster.node1.ehcache. rmi_cache_manager_peer_ listener_factory.port.primary</code>	<code>11130</code>	Port di binding per i cache listener per il nodo1 del cluster (valido solo per la configurazione in cluster in multi repository per il repository primario)
<code>target.cluster.node2.ehcache. rmi_cache_manager_peer_ listener_factory.port.primary</code>	<code>11130</code>	Port di binding per i cache listener per il nodo2 del cluster (valido solo per la configurazione in cluster in multi repository per il repository primario)
<code>target.cluster.node1.ehcache. rmi_cache_manager_peer_ listener_factory.port.secondary</code>	<code>11131</code>	Port di binding per i cache listener per il nodo1 del cluster (valido solo per la configurazione in cluster in multi repository per il repository secondario)
<code>target.cluster.node2.ehcache. rmi_cache_manager_peer_ listener_factory.port.secondary</code>	<code>11131</code>	Port di binding per i cache listener per il nodo2 del cluster (valido solo per la configurazione in cluster in multi repository per il repository secondario)
<code>target.use_csi</code>	<code>false</code>	Utilizzo del framework CSI (per iniziative sviluppate all'interno del CSI) (possibili valori true o false)
<code>target.use_splitting</code>	<code>false</code>	Utilizzo della funzionalita' di splitting dei contenuti
<code>target.use_multirepository</code>	<code>true</code>	Utilizzo del multirepository (possibili valori true o false)
<code>target.use_encryption</code>	<code>true</code>	Utilizzo del meccanismo di crypting dei contenuti (possibili valori true o false)
<code>target.splitting.parts</code>	<code>20</code>	Numero di parti da configurare per la distribuzione dinamica dei
<code>target.repository.definition</code>	<code>single</code>	Configurazione in multirepository Possibili valori (single multi single-clustered multi-clustered)
<code>target.avm.remote.port</code>	<code>23462</code>	Porta dei servizi rmi
<code>ds.jndi-name</code>	<code>ecmengine/jdbc/ecmengineDS</code>	Nome del data source per i servizi aggiunti da ecmengine
<code>ds.connection-url</code>	<code>jdbc:mysql://localhost:3306/</code>	Url di connessione alla base dati

	<p style="text-align: center;">INDEX</p> <p style="text-align: center;">INSTALLAZIONE PIATTAFORMA</p>	<p>PGED-INDEX-ARC-v01</p> <p>Pag. 11 di 21</p>
---	---	--

	ecmdb	
ds.driver-class	org.gjt.mm.mysql.Driver	Driver da utilizzare per la connessione alla base dati
ds.user-name	indexadmin	Utente della base dati
ds.password	indexadmin	Password dell'utente della base dati
ds.min-pool-size	0	Dimensione iniziale del connection pool
ds.max-pool-size	11	Dimensione massima del connection pool
ds.idle-timeout-minutes	2	Timeout di connessione espresso in minuti
ds.exception-sorter-class-name	org.jboss.resource.adapter. jdbc.vendor. SQLExceptionSorter	Traduttore dei codici di errore di MySQL
target.job.node.master	node1	Nodo master del cluster (valido solo per la configurazione in cluster)
target.include_pabr	false	Inclusione del bridge SOAP del framework CSI
URL_TO_CONNECT	jnp://localhost:1099	Url di connessione alla piattaforma nel formato <protocollo>://<url>:<porta> (utilizzato dalla soluzione csi web services)
CONTEXT_FACTORY	org.jnp.interfaces. NamingContextFactory	Classe per effettuare il binding su JNDI di Jboss (utilizzato dalla soluzione csi web services)

3.3.1 Web service basati su engine Axis 1.4 (opzionali)

E' possibile generare web service basati su Axis 1.4 mediante il file `external_ws.properties` presente nella directory buildfiles del repository dei sorgenti contenente i seguenti parametri di configurazione:

Parametro	Valore	Descrizione
DELEGATE_CLASS	it.doqui.index.ecmengine.client.	Classe che implementa il pattern

	<p style="text-align: center;">INDEX</p> <p style="text-align: center;">INSTALLAZIONE PIATTAFORMA</p>	<p>PGED-INDEX-ARC-v01</p> <p>Pag. 12 di 21</p>
---	---	--

	<code>engine.EcmEngineDelegateImpl</code>	Delegate per la chiamata ai servizi della piattaforma per i servizi di management e ricerca
BACKOFFICE_DELEGATE_CLASS	<code>it.doqui.index.ecmengine.client.backoffice.EcmEngineBackofficeDelegateImp</code>	Classe che implementa il pattern Delegate per la chiamata ai servizi della piattaforma
URL_TO_CONNECT	<code>jnp://localhost:1099</code>	Url di connessione alla piattaforma nel formato <code><protocollo>://<url>:<porta></code>
CONTEXT_FACTORY	<code>org.jnp.interfaces.NamingContextFactory</code>	Classe per effettuare il binding su JNDI di Jboss
DISABLE_DISCOVERY	<code>true</code>	Disabilitazione del discovery sulle istanze jboss

3.4 Creazione ed installazione dei pacchetti da deployare (operazione necessaria sia per le realta' che creano i pacchetti sia per quelle che deployano il pacchetto scaricato)

A questo punto puo' essere lanciata la compilazione delle classi e la creazione del pacchetto di deploy.

E' necessario aver configurato

le variabili

JAVA_HOME=<path del jdk>

JAVA_HOME_SERVER=<path del jdk>

ANT_HOME=<path dell'istallazione di ant>

sotto un esempio di configurazione di file di lancio (windows ed unix) con le configurazioni applicate per l'installazione effettuata per le regioni liguria e valle d'aosta.

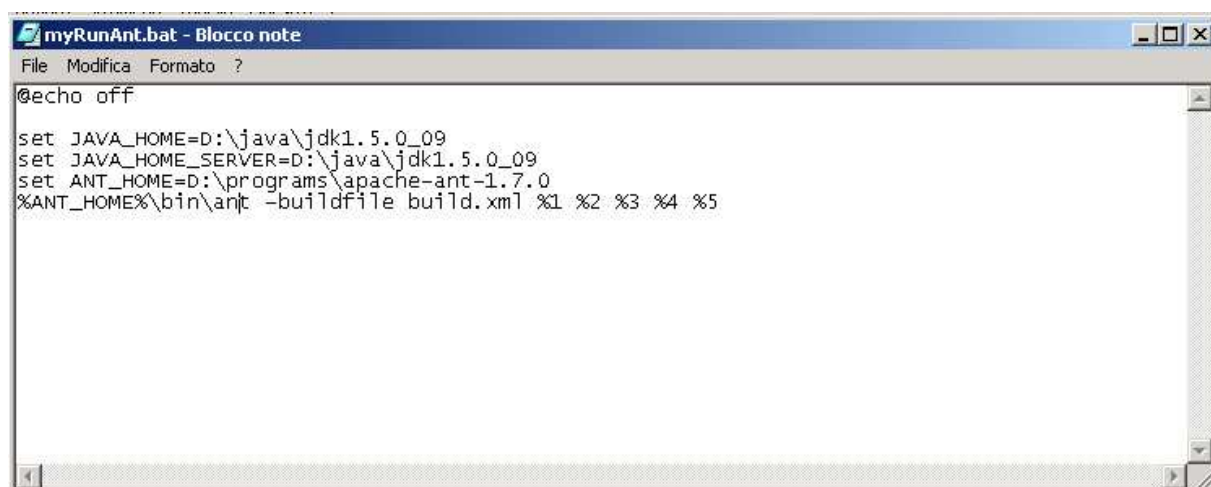


Figura 2.4 : file di lancio della compilazione per l'ambiente windows (myRunAnt.bat)

```
#!/bin/sh
export JAVA_HOME=/usr/lib/jvm/java-1.5.0-sun
export JAVA_HOME_SERVER=/usr/lib/jvm/java-1.5.0-sun
export ANT_HOME=/usr/share/ant
$ANT_HOME/bin/ant -buildfile build.xml $1 $2 $3 $4 $5
```

Figura 2.5 : file di lancio della compilazione per l'ambiente unix (myRunAnt.sh)

Principali target

Target	Valore
load-dependencies	All'interno del CSI le librerie vengono scaricate da un repository comune questo task prevede lo scarico in locale impostate per il progetto ecmengine
package-ear	Creazione dell'ear distribuzione
all-client-jar	Creazione dei client da distribuire ai fruitori della piattaforma (diversi se csi environment o no)
javadoc-servizi-applicativi	Creazione dei javadoc dei servizi
package-unit-test	Creazione del pacchetto per i test junit e' necessario inserire un file local.properties all'interno del repository dei sorgenti di cui daremo spiegazione nel paragrafo ??

	<p style="text-align: center;">INDEX</p> <p style="text-align: center;">INSTALLAZIONE PIATTAFORMA</p>	<p>PGED-INDEX-ARC-v01</p> <p>Pag. 14 di 21</p>
---	---	--

compile-all	Compilazione di tutte le classi
package-all	Creazione di tutti i pacchetti per l'ambiente impostato
distribution	Compilazione delle classi e creazioni dei pacchetti di installazione inseriti in un file in formato tar nella directory dist
generate-axis-webservices	Creazione del pacchetto war della web application dei web service basati su engine Axis 1.4
generate-axis-webservice-client	Creazione del client da distribuire ai fruitori del web service basato su Axis 1.4
show-vars	Visualizzazione dei target di ant impostati (piu' di quelli espressi in questa tabella)
clean-classes	Pulizia delle classi compilate
clean-all	Pulizia delle classi compilate e della directory in cui avvengono le sostituzioni dei file di configurazione

`./myRunAnt.sh -Dtarget=<ambiente> distribution`

Il target distribution prevede la creazione dei seguenti pacchetti :

<code>-rw-r--r-- 1 lab05 lab05 49848320 2008-02-01 09:47 ecmengine-2.0.0.tar</code>	—	Pacchetti e configurazione piattaforma
<code>-rw-r--r-- 1 lab05 lab05 123443 2008-02-01 09:47 all-client.zip</code>	—	Client da distribuire ai fruitori della piattaforma contenenti Eccezioni, DTO ed interfacce di comunicazione
<code>-rw-r--r-- 1 lab05 lab05 6474 2008-02-01 09:40 ecmengine_pd_v-2.0.0.zip</code>	—	Porte delegate del framework CSI (utilizzato per i verticali sviluppati per l'infrastruttura interna)

La piattaforma deplorable e' pero' composta dal pacchetto ecmengine-< versione nel formato major.medium.minor >.tar mentre lo zip all-client.zip contiene gli oggetti da distribuire ai verticali che utilizzano la piattaforma.

Ecmengine_pd_v-< versione nel formato major.medium.minor >.zip contiene i file di configurazione per il framework CSI utilizzato per le iniziative sviluppate e deployate all'interno dell'infrastruttura CSI.

Ecmengine-<versione nel formato major.medium.minor>.tar contiene i seguenti pacchetti :



INDEX

INSTALLAZIONE PIATTAFORMA

PGED-INDEX-ARC-v01

Pag. 15 di 21

-rw-r--r--	1	root	root	8553	2008-02-01	10:08	ecmengine.zip	Estensioni di alfresco versione 2.1 e dell'iniziativa DOUI
-rw-r--r--	1	root	root	6549928	2008-02-01	10:08	ecmengine.war	Web archive per la comunicazione attraverso web services con l'engine Axis1.4
-rw-r--r--	1	root	root	835	2008-02-01	10:08	ecmengine-ds.xml	Pacchetto creato solo se si sceglie la configurazione web services e non csi
-rw-r--r--	1	root	root	43280274	2008-02-01	10:08	ecmengine.ear	Data source per la comunicazione verso la base dati per le funzionalita' aggiunte dall'iniziativa DOQUI (audit, audit trail) implementate attraverso il pattern DAO

Piattaforma dei servizi dell'iniziativa DOQUI

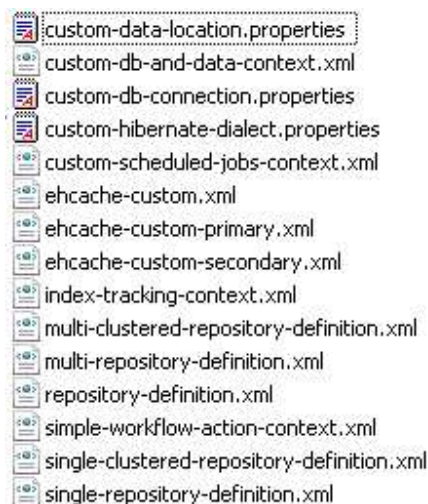
Ps: le dimensioni dei pacchetti potrebbero variare minimamente in base all'ambiente di compilazione e al jdk utilizzato.

Tali pacchetti vanno posizionati nelle seguenti posizioni :

Pacchetto	Azione e posizione
<target/nodi>.zip	File di configurazione esterni. Creare la directory <jboss_home>/server/<partizione/istanza>/conf/ecmengine/ e scompattarvi il contenuto dello zip file (Il pacchetto contiene all'interno la cartella extension in modo che i file .properties ed xml siano posizionati nella directory <jboss_home>/server/<partizione/istanza>/conf/ecmengine/extension).
ecmengine.war	Pacchetto opzionale per comunicazione via web service (Axis 1.4) da copiare nella directory <jboss_home>/server/<partizione/istanza>/deploy
ecmengine-ds.xml	da copiare nella directory <jboss_home>/server/<partizione/istanza>/deploy
ecmengine.ear	da copiare nella directory <jboss_home>/server/<partizione/istanza>/deploy

	<p style="text-align: center;">INDEX</p> <p style="text-align: center;">INSTALLAZIONE PIATTAFORMA</p>	<p>PGED-INDEX-ARC-v01</p> <p>Pag. 16 di 21</p>
---	---	--

Il pacchetto target/nod1.zip (il nome dipende dalla configurazione in cluster o meno) e' cosi' composto :



Controllare nel file di custom-db-connection.properties il corretto puntamento alla/e base/i dati.

Controllare nel file di custom-data-location.properties il corretto puntamento alle directory contenenti gli indici di lucene e i contenuti e che tali cartelle siano create e scrivibili.

Nel caso di configurazione in cluster controllare la configurazione del file ehcache-custom-<repository> che <indirizzo della porta multicast> e <indirizzo della porta multicast primario> non siano occupati da altri processi.

```
<cacheManagerPeerProviderFactory
  class="net.sf.ehcache.distribution.RMICacheManagerPeerProviderFactory"
  properties="peerDiscovery=automatic,
    multicastGroupAddress=<indirizzo della porta multicast>,
    multicastGroupPort=<indirizzo della porta multicast primario>"/>
```

3.5 Configurazione ambiente (operazione necessaria sia per le realta' che creano i pacchetti sia per quelle che deployano il pacchetto scaricato)

3.5.1 Creazione base/i dati

Recuperare il codice della parte di creazione della/e basi dati precentemente scaricato .

Creare lo schema della base dati attraverso il comando riportato nella figura sottostante da MySQLQueryBrowser o da riga di comando come utente amministratore della base dati.

L'operazione deve essere ripetuta per tutte le basi dati utilizzate.



Figura 2.3 : Creazione dell'utente della base dati

Associati alla componente ecmengine vengono rilasciate due componenti :

indexadmin-<versione>.zip	per la creazione delle tavole della base dati mysql
checkpermission-<versione>.zip	per la creazione della stored procedure di check delle permission

All'interno di tali zip sono contenuti gli script sopra elencati.

Lanciare con l'utente precedentemente definito i due script (l'operazione deve essere effettuata per tutte le basi dati, se sono in numero superiore a uno).

3.5.2 Configurazione logger applicativo (operazione necessaria sia per le realta' che creano i pacchetti sia per quelle che deployano il pacchetto scaricato)

Per il log applicativo e' stato utilizzato il framework log4j (<http://logging.apache.org/>) versione 1.2.8 a cui si appoggia anche jboss.

La configurazione del log deve essere effettuata nella configurazione di log4j su jboss nella directory :

<jboss_home>/server/<partizione/istanza>/conf

il file log4j.xml

aggiungere le seguenti category :

Configurazione Category	Effetto
<pre><category name="org.hibernate"> <priority value="ERROR" /> <appender-ref ref="CONSOLE"/> </category> <category name="net.sf"> <priority value="ERROR" /> <appender-ref ref="CONSOLE"/> </category> <category name="org.alfresco"> <priority value="ERROR" /> <appender-ref ref="CONSOLE"/> </category> <category name="org.jboss"> <priority value="ERROR" /></pre>	<p>Stampa sulla console solo i messaggi di errore dei framework hibernate, lucene, spring, alfresco</p>

<pre> <appender-ref ref="CONSOLE"/> </category> <category name="org.springframework"> <priority value="ERROR" /> <appender-ref ref="CONSOLE"/> </category> </pre>	
<pre> <category name="index.ecmengine"> <priority value="INFO" /> <appender-ref ref="INDEX"/> </category> <category name="index.ecmengine.util.stopwatch"> <priority value="DEBUG" /> <appender-ref ref="INDEX_PERF"/> </category> </pre>	Ridirige l'output della piattaforma ecmengine (log applicativo e tracing dei tempi di esecuzione su un log separato)

Aggiungere i seguenti appender per log4j

Configurazione Appender	Effetto
<pre> <appender name="INDEX" class="org.jboss.logging.appender.RollingFileAppender"> <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/> <param name="File" value="<directory log>/index.log" /> <param name="Append" value="false"/> <param name="MaxFileSize" value="1000KB"/> <param name="MaxBackupIndex" value="5"/> <layout class="org.apache.log4j.PatternLayout"> <param name="ConversionPattern" value="%d %-5p [%c] %m%n"/> </layout> </appender> </pre>	Nella directory <directory log> creata viene scritto il log applicativo index.log Il file di log segue una politica di Rolling con 5 backup di 1000Kb l'uno
<pre> <appender name="INDEX_PERF" class="org.jboss.logging.appender.RollingFileAppender"> <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/> <param name="File" value="/home/lab05/projects/logs/index_perf.log" /> <param name="Append" value="false"/> <param name="MaxFileSize" value="10000KB"/> <param name="MaxBackupIndex" value="5"/> <layout class="org.apache.log4j.PatternLayout"> <param name="ConversionPattern" value="%d,[%p],[%c],[%t], %X{ index.ecmengine.stopwatch.elapsed}, %X{ index.ecmengine.stopwatch.context},%m,%n"/> </layout> </appender> </pre>	Nella directory <directory log> creata viene scritto il log applicativo index_pef.log contenente i tempi di esecuzione delle operazioni sulla piattaforma. Il file di log segue una politica di Rolling con 5 backup di 10000Kb l'uno

	<p style="text-align: center;">INDEX</p> <p style="text-align: center;">INSTALLAZIONE PIATTAFORMA</p>	<p>PGED-INDEX-ARC-v01</p> <p>Pag. 19 di 21</p>
---	---	--

3.6 Avvio di jboss (operazione necessaria sia per le realta' che creano i pacchetti sia per quelle che deployano il pacchetto scaricato)

Avviare jboss sul/i nodo/i, in fase di deploy del pacchetto potrebbero visualizzarsi dei warning, non importanti per il corretto funzionamento della piattaforma.

3.7 Accesso client verso web service Axis 1.4 (opzionale)

L'accesso client ad ecmengine tramite web service Axis viene eseguito tramite un'apposita libreria client Axis che viene generata dal build o può essere generata esternamente con altri strumenti sfruttando i file WSDL dei servizi presenti nel repository dei sorgenti:

- src/web/ecmengine/ecmengine-backoffice.wsdl
- src/web/ecmengine/ecmengine-management.wsdl

Le generazione della libreria client da build viene eseguita lanciando il comando:

```
./myRunAnt.sh generate-axis-webservice-client
```

L'esecuzione genererà il file ecmengine-ws-client-< versione nel formato major.medium.minor >.jar, che potrà essere utilizzato da subito per eseguire l'accesso al web service.

4. Test dei servizi pubblicati per i fruitori

4.1 Test unitari

Per i test unitari e' stato utilizzato il framework junit3.8.1.

4.1.1 Sviluppo dei test unitari

I test unitari sono stati implementati in una directory parallela (test) a quella contenente il codice sorgente del componente ecmengine.

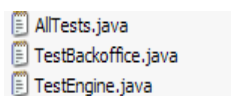
I test unitari sono contenuti nei package

it.doqui.index.ecmengine.test contenente la suite dei test unitari

it.doqui.index.ecmengine.test.util contenente le utility per i test unitari.

La suite dei test e' composta da una classe AllTests che estende il TestCase di junit.

Su questa classe viene definita una suite che carica le due classi contenenti i test per le funzionalità di backoffice (gestione utenti, gruppi etc. contenute in TestBackoffice.java) e i test per le funzionalità di gestione e ricerca (TestEngine.java).



4.1.2 Creazione pacchetto dei test unitari

E' stato deciso di creare un pacchetto in formato zip attraverso il target ant

ant -Dtarget=<ambiente.properties> package-unit-test

contenente tutte le librerie e un file di lancio (.bat e .sh) dipendente dall'ambiente di esecuzione.

Questa decisione è stata intrapresa per dare la possibilità di lanciare i test unitari anche dalla postazione locale degli sviluppatori, delle persone che testeranno i servizi, delle persone che scaricheranno la componente ecmengine e vorranno testare la propria installazione con la disponibilità dei servizi.

Il pacchetto dei test creato attraverso un target di build di ant, che descriveremo successivamente, conterrà

- Libreria contenente le classi dei test unitari (ecmengine-test-<versione>.jar)
- Librerie contenenti i file di comunicazione verso i servizi del ecmengine (ecmengine-backoffice-client-<versione>.jar, ecmengine-engine-client-<versione>.jar, ecmengine-common-<versione>.jar)
- Librerie di utilità e nel caso dell'utilizzo dello CSI Framework le librerie di tale framework come documentato sotto dalla presenza delle librerie csi*.jar
- File log4j.properties contenente le configurazioni di log4j per la parte dei test unitari.
- File di lancio (runJunit.bat o runJunit.sh a seconda dell'ambiente, nella figura sottostante un pacchetto generato per l'ambiente Windows)

Per creare il pacchetto dei test unitari e' necessario

- Aggiungere un file di proprietà contenente le informazioni riguardanti l'ambiente in cui verranno eseguiti i test
- Controllare il file di properties riguardante le proprietà dell'ambiente in cui sono disponibili i servizi della componente ecmengine.
- Lanciare il comando ant package-unit-test -Dtarget=<ambiente di deploy dei servizi ecmengine>

4.1.2.1 File di property local.properties

Tale file deve essere aggiunto sulle macchine su cui si sta creando il pacchetto dei test unitari.

Tale file non e' presente sul repository svn e non dovrà mai essere presente in quanto verrebbero salvate informazioni dipendenti soltanto dalla configurazione di un'unica macchina.

Il file local.properties contiene le seguenti properties

```
url_to_connect=<url di connessione ai servizi ecmengine>
cluster_partition=<partizione su cui sono stati deployati i servizi ecmengine >
disable_discovery=<parametro disable discovery del server su cui sono deployati i servizi ecmengine >
operating_system=<sistema operativo (windows o linux)>
path_jdk_executor_test=<path del jdk sulla macchina su cui verranno eseguiti i test>
path_log=<path dei log dei test unitari>
backofficeDelegateClass=<classe delegate da utilizzare per le chiamate ai servizi backoffice del ecmengine > (*)
delegateClass=<classe delegate da utilizzare per le chiamate ai servizi di management e ricerca del ecmengine >
(**)
```

4.1.2.2 File di property <ambiente di deploy servizi ecmengine >.properties

Su tale file e' necessario controllare il settaggio della proprietà target.use_csi.

target.use_csi=true

Tale proprietà imposta se si sta utilizzando il CSI Infrastructure (true si sta utilizzando, false non si sta utilizzando)

Per quanto riguarda la creazione del pacchetto dei test unitari la proprietà crea :

- Differenti librerie client dei servizi (se si utilizza la chiamata diretta all'InitialContext servono le interfacce degli EJB di pubblicazione servizi ecmengine, altrimenti con il framework CSI le interfacce sono mappate sulla Porta Applicativa di CSI Infrastructure)
- La property target.use_csi deve essere utilizzato in coppia con le property backofficeDelegateClass e delegateClass sul file local.properties secondo la tabella sottostante

	target_use_csi=true	target_use_csi=false
delegate (*)	OK	KO
direct delegate (**)	KO	OK

(*)delegate (it.doqui.index.ecmengine.client.backoffice_EcmEngineBackofficeDelegateImpl per i servizi del backoffice, it.doqui.index.ecmengine.client.engine_EcmEngineDelegateImpl per i servizi di gestione e ricerca)
(**)direct delegate (it.doqui.index.ecmengine.client.backoffice_EcmEngineDirectDelegateImpl per i servizi del backoffice,
it.doqui.index.ecmengine.client.backoffice_EcmEngineBackofficeDirectDelegateImpl per i servizi di gestione e ricerca)