

Installare o Aggiornare itaEngine LINUX/WINDOWS

Questo wiki raccoglie la descrizione delle diverse attività che concorrono all'installazione di itaEngine. Il wiki è in fase di costruzione quindi allo stato attuale prevalentemente una raccolta di appunti da consultare secondo le esigenze del sistemista. si è cercato comunque di rispettare la sequenza temporale migliore per ottenere una corretta installazione delle procedure.

Preparazione del server web

Linux

Si parte dal presupposto che si disponga di una installazione lamp su una distribuzione linux quale:

- Suse >= 11.4
- Debian
- Centos
- Redhat

Altre distribuzioni non sono state utilizzate ma fondamentalmente si necessita dei pacchetti elencati:

Apache Http server 2.2 >=

php 5.x =

Jre 1.6 >=

Verifica Autorizzazioni SELINUX

Nei casi in cui si attivo il pacchetto SELINUX Disabilitare come segue:

```
vi /etc/sysconfig/selinux
```

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
# enforcing - SELinux security policy is enforced.
# permissive - SELinux prints warnings instead of enforcing.
# disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
# targeted - Targeted processes are protected,
# mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

Riavviare il sistema oppure:

```
# setenforce 0
```

Controllare se SELinux è stato correttamente disattivato

```
# getenforce
```

Configurare Apache

- Editare un nuovo file di configurazione di apache es. **/etc/apache2/conf.d/itaEngine.conf** e predisporre una cartella web che ospiti l'applicativo itaEngine:

httpd 2.4

```
Alias /itaEngine "E:/Cityware.Online/xampp/htdocs/itaEngine"  
<Directory E:/Cityware.Online/xampp/htdocs/itaEngine>  
    AllowOverride None  
    Options -Indexes +FollowSymLinks +ExecCGI  
    Order deny,allow  
    Allow from all  
</Directory>
```

httpd 2.2

```
Alias /itaEngine/ "d:/Cityware.online/xampp/htdocs/itaEngine/"  
<Directory d:/Cityware.online/xampp/htdocs/itaEngine/>  
    AllowOverride None  
    Options -Indexes +FollowSymLinks +ExecCGI  
    Order allow,deny  
    Allow from all  
</Directory>
```

Moduli PHP

Verificare l'esistenza dei seguenti moduli:

CentOS > *php -m*

- apc
- Bz2
- Chilkat
- curl
- dom
- gd
- ftp
- json
- ldap (**Attenzione non sempre presente**)
- libxml
- mbstring
- mysql

- mysqli
- openssl
- simplexml
- soap
- pgsql
- PDO:pdo_mysql
- PDO:pdo_postgres
- xml
- xmlreader
- xmlrpc
- xmlwriter
- [Zend Modules]
- Xdebug

Verificare la presenza del pacchetto pear con il comando:\

```
pear list -a
```

Pacchetti:

- mail
- mail_mime
- mail_mimedecode
- soap

Se mancante installare con gli strumenti messi a disposizione dalla distribuzione linux:

* yast * yum * apt-get * Etc...

```
pear list -a
Mail_Mime
Mail_mimeDecode
Net_dime
SOAP
pear install Mail_Mime
pear install Mail_mimeDecode
pear install Net_Dime
pear install SOAP-beta
```

Parametrizzazione php

- Editare il file: **/etc/php5/apache2/php.ini** e gestire i seguenti parametri
 - session.gc_maxlifetime = 3600 per aumentare o diminuire il tempo di sessione
 - session.gc_divisor = 1000 per aumentare o diminuire il numero di chiamate a php prima di effettuare il controllo di validità sessione
 - max_execution_time = 7200 per aumentare il tempo massimo di esecuzione delle script
 - memory_limit = 1024M aumentare il picco di memoria ad almeno 1GB
 - upload_max_filesize = 20M
 - post_max_size = 50M
 - max_input_vars = 3000 per aumentare o diminuire il numero di dati in input da elaborare

- In modalità “debug” attivare il display_error (ricordarsi di disattivarlo in produzione!)
 - display_error on

Configurazione repository locale

Clonare il repository pubblico:

```
git clone http://itadeployer@srvgitlab.apra.it/itaengine/itaEngine.git
git checkout [branch] (es. sviluppo, pagoPa)
git pull
```

Utente pubblico per accesso a Git: itadeployer / deploy123

esempio del file .git/config

```
[core]
  repositoryformatversion = 0
  filemode = true
  bare = false
  logallrefupdates = true
[remote "origin"]
  fetch = +refs/heads/*:refs/remotes/origin/*
  url =
http://itadeployer:deploy123@srvgitlab.apra.it/itaengine/itaEngine.git
[branch "master"]
  remote = origin
  merge = refs/heads/master
[credential]
  helper = store
```

phpMyAdmin

Installare l'ultima versione di phpmyadmin ed eventualmente rendere sicura la cartella con le direttive di htaccess o editando il file **/etc/apache2/conf.d/phpMyAdmin.conf**: (esempio)

```
Alias /qaz321 /usr/share/phpMyAdmin
Alias /mko987 /usr/share/phpMyAdmin

<Directory /usr/share/phpMyAdmin/>
  Order Deny,Allow
  Allow from 188.219.234.118
  Allow from 127.0.0.1
</Directory>
```

Java Virtual Machine

installare una jdk 1.8.045 o superiore, verificare con il comando:

```
java -version
```

Attenzione utilizzare prodotto oracle e non openjdk per maggiore compatibilità

(Documentare i repository)

Php Database driver

Configurazione Sql Server su Linux centOS

Per poter usare il driver pdo per Microsoft SQL Server è necessario installare i pacchetti php-pdo e php-mssql presenti nei repository della distribuzione, il driver pdo_dblib verrà automaticamente abilitato. Nel caso in cui, dopo il riavvio del servizio apache, pdo_dblib non fosse presente nella lista dei driver abilitati si può scaricare manualmente il modulo *pdo_dblib.so*, inserirlo nella directory contenente le estensioni di php (php -info | grep extension_dir) e attivarlo tramite il file di configurazione php.ini (extension=pdo_dblib.so)

I parametri di configurazione del driver pdo_dblib vengono letti dal file */etc/freetds.conf*. Esempio:

```
[global]
    timeout = 10
    connect timeout = 10
    text size = 6451200
```

Installare pacchetto: php-mssql-5.3.3-5.el6.x86_64

Moduli php: mssql pdo_dblib

Riferimenti esterni altri driver

Per l'installazione dei driver fare riferimento al seguente link:

<http://wikidev.apra.it/itawiki/doku.php?id=sviluppo:multidb>

Attivazione moduli php

mbstring Il modulo mbstring è necessario per far funzionare l'istruzione mb_detect_encoding(). Per CentOS è possibile con il seguente comando:

```
yum install php-mbstring
```

Documentazione di riferimento: [mbstring.installation](#)

Chilkat 9.5.0

Chilkat è utilizzato per convertire un certificato ppk in certificato openssh. Viene utilizzato ad esempio su pagopa per convertire il certificato dell'sftp di efil. Per ora è richiesto solo se presente il modulo pagopa.

Installazione windows: aggiungere extension=chilkat_9_5_0.dll su php.ini e mettere sotto php/ext la

dll presa da <https://www.chilkatsoft.com/php.asp>

Installazione linux: seguire le istruzioni in base al tipo di macchina:
<https://www.chilkatsoft.com/php.asp>

APC

Linux:

```
pecl install apc
```

Utilizzare le opzioni di default

In alternativa le istruzioni per l'installazione: <https://www.electrictoolbox.com/install-apc-php-linux/>

Windows:

Scaricare il pacchetto precompilato da <http://windows.php.net/downloads/pecl/releases/apcu>

Xdebug

Linux: scaricare e compilare il pacchetto wget <https://xdebug.org/files/xdebug-2.4.1.tgz>

```
phpize  
./configure  
make  
make install
```

```
; Enable xdebug extension module  
zend_extension=/usr/lib64/php/modules/xdebug.so  
xdebug.profiler_append = 0  
xdebug.profiler_enable = 1  
xdebug.profiler_enable_trigger = 0  
xdebug.profiler_output_dir = "/tmp"  
xdebug.profiler_output_name = "cachegrind.out.%t-%s"  
xdebug.profiler_enable = 0  
xdebug.profiler_append=0  
xdebug.extended_info=1  
xdebug.remote_enable=1  
xdebug.remote_handler=dbgp  
xdebug.remote_mode=req  
xdebug.remote_host=127.0.0.1  
xdebug.remote_port=9000  
xdebug.idekey=netbeans-xdebug  
xdebug.trace_output_dir = "/tmp"  
xdebug.show_exception_trace=0  
xdebug.show_local_vars=9  
xdebug.show_mem_delta=0  
xdebug.trace_format=0
```

Documentazione Composer

<http://wikidev.apra.it/itawiki/doku.php?id=itaengine:composer>

Soap

Per eseguire l'installazione del modulo soap usare:

```
pear install soap
```

in caso di errore specificare il channel con:

```
pear install channel://pear.php.net/SOAP-0.13.0
```

Configurazione Omnis Server

Da interfaccia web itaEngine aprire il pannello di controllo > Configurazione parametri > Parametri Omnis Studio

Impostare i seguenti parametri

```
Protocollo: http
Url Http Server: [indirizzoHttpServer:porta]
Url Omnis Web Server: [indirizzoOmnisServer:porta]
Omnis CGI Name: /cgi-bin/nph-omniscgi.exe
Libreria di default: CITYWARE
Nome Remote Task: RT_HTTP_REQUEST
Nome Remote Task (interattivo): RT_HTTP_REQUEST_INT
Parametri in formato esadecimale: 0
Gestione abilitazione Omnis Server: 1
```

Esempio



Configurazione Moduli itaEngine

Database Cityware : Configurazione di config/connections.ini

<http://wikidev.apra.it/itawiki/doku.php?id=itaengine:citywarePdo>

City People : <http://wikidev.apra.it/itawiki/doku.php?id=itaengine:cityPeople>

City PagoPA : <http://wikidev.apra.it/itawiki/doku.php?id=itaengine:cityPagoPA>

Albo Online : <http://wikidev.apra.it/itawiki/doku.php?id=itaengine:cityAlboOnline>

From:
<https://wiki.nuvolaitalsoft.it/> - **wiki**

Permanent link:
<https://wiki.nuvolaitalsoft.it/doku.php?id=installazione:itaengine&rev=1518611065>

Last update: **2018/03/19 10:45**

