

# **Piattaforma SISO**

## **Manuale di Installazione e Configurazione**

## Indice dei Contenuti

### Sommario

1.	Introduzione .....	3
1.1	Premessa .....	3
1.1	Storia delle modifiche .....	3
1.2	Riferimenti al repository dei rilasci .....	3
2.	Istallazione .....	4
2.1	Database .....	4
2.2	JVM – Server JBoss .....	5
2.3	Cartella “dati diogene” .....	5
2.4	Configurazione Server JBoss .....	5
2.4.1	Impostazione opzioni JVM .....	5
2.4.2	Creazione Servizio Windows .....	6
2.4.3	Istanze multiple di JBoss 7 su stesso server .....	6
2.4.4	Configurazione datasource .....	6
2.4.5	Configurazione logging .....	7
2.4.6	Ulteriori opzioni di configurazione .....	7
3.	Deploy Applicazioni .....	8
3.1	Procedura di deploy .....	8
3.2	Aggiornamento Applicazioni .....	9

# 1. Introduzione

## 1.1 PREMESSA

La piattaforma SISO è una piattaforma distribuita di servizi che prevede una fase di installazione descritta in questo documento e una fase successiva di configurazione strettamente dipendente dalla realtà nella quale il prodotto viene applicato.

La piattaforma **SIGESS** deriva dalla piattaforma a riuso **SISO** e ne condivide repository sorgente e repository delle release. Il presente documento viene applicato ad entrambe le piattaforme in quanto i due prodotti fanno riferimento ad una stessa baseline di sorgente applicativo.

## 1.1 STORIA DELLE MODIFICHE

### Versione 4

Versione con eliminazione delle istruzioni per la piattaforma GIT/C&T ed elencazione delle sole istruzioni per SISO/SIGESS.

## 1.2 RIFERIMENTI AL REPOSITORY DEI RILASCI

La piattaforma SISO mette a disposizione un repository SVN dal quale possono essere scaricate le release rilasciate dal laboratorio sviluppo unico interregionale mantenuto da Umbria Digitale per i soggetti cedenti della piattaforma e per i soggetti riusanti con accordi di collaborazione in corso per il mantenimento e la gestione del laboratorio.

L'accesso al repository prevede delle credenziali rilasciate da Umbria Digitale.

### Repository release:

Item	Riferimento
Release X.Y.Z-SISO-w.y (*)	<a href="http://172.29.0.251:3690/svn/ConduzioneOperativa-CeT/dist/tags/SISO/">http://172.29.0.251:3690/svn/ConduzioneOperativa-CeT/dist/tags/SISO/</a>
Release di GIT X.Y.Z. (necessaria per SISO)	<a href="http://172.29.0.251:3690/svn/ConduzioneOperativa-CeT/dist/tags/">http://172.29.0.251:3690/svn/ConduzioneOperativa-CeT/dist/tags/</a>
Database di riferimento	Release X.Y.Z-SISO-w.y / full / database (**) Release X.Y.Z-SISO-w.y / full / database
Applicazioni	Release X.Y.Z-SISO-w.y / full / app (**) Release X.Y.Z-SISO-w.y / update / app
Schede Intervento Software di release	Release X.Y.Z-SISO-w.y / update / Schede Intervento Software

(\*) X.Y.Z. si riferisce alla release GIT necessaria a SISO w.y

(\*\*) nuovi impianti

La Piattaforma viene anche rilasciata agli enti riusanti che possono predisporre un proprio repository generando un fork applicativo.

## 2. Installazione

### 2.1 DATABASE

Il database di riferimento di ogni versione di SISO è contenuto all'interno del pacchetto di rilascio della stessa release sotto forma di file dump oracle.

Il suddetto file, contenente gli schemi 'vuoti' denominato SISO.DMP v. 11.02 di Oracle, va importato utilizzando il comando IMPDP; le tablespace di origine sono denominate 'GIT' per i dati, mentre per la temp è denominata 'TEMP'. Gli schemi da importare contenuti nel file sono i seguenti: AM, ARGO, ARGOBI, CS, DIOGENE\_H501, RE

Successivamente al caricamento degli schemi di base va eseguito lo script .SQL presente nel pacchetto rilasciato denominato '\_nuovoente.sql' passandogli alcuni parametri; di seguito un esempio pratico:

```
sqlplus system/manager@istanza @_nuovoente.sql 'A769' 'AM_01' 'CS_01_09'  
'DIOGENE_A769' 'BELVEDERE OSTRENSE' 'belvedereostrenseM' 'E' 'A769' '042005' >  
Script_A769.log
```

I parametri da modificare indicano:

'A769' il codice Belfiore del comune

'AM\_01' lo schema AM precedentemente importato

'CS\_01\_09' lo schema CS precedentemente importato

'DIOGENE\_A769' lo schema DIOGENE precedentemente importato

'BELVEDERE OSTRENSE' la descrizione del Comune che si intende importare

'belvedereostrenseM' utente manager del Comune

'E' lettera che indica l'unità disco del filesystem dove risiedono alcune directory di personalizzazione

'042005' Codice ISTAT Provincia-Comune

Il secondo script da eseguire è il seguente con alcuni parametri da personalizzare:

```
sqlplus system/manager@istanza @r_connection.sql 'A769' 'A769' 'RE_01' '10.101.11.62'  
'SISO' > R_connection_A769.log
```

'A769' il codice Belfiore del comune

'RE\_01' lo schema RE precedentemente importato

'10.101.11.62' indirizzo IP del Database

'SISO' istanza del database

Sono presenti dei sinonimi nei vari schemi per cui una verifica da fare e se questi sinonimi siano referenziati correttamente.

## 2.2 JVM – SERVER JBOSS

Per la procedura di installazione dell'ambiente server viene fornito un JBoss 7.1.1 personalizzato che sarà successivamente necessario configurare.

Per installare l'ambiente di base è necessario:

- Copiare JBoss contenuto nella directory **ambiente\JBoss** in una directory del server. Solitamente tale procedura viene eseguita nella directory root del sistema operativo quindi `c:\` per windows o `/` per unix
- Installare la JVM più aggiornata contenuta nella cartella **ambiente\jvm**

Una volta scompattato l'archivio .rar di Jboss, il server è installato, tuttavia è necessaria una fase successiva di configurazione per definire alcune impostazioni fondamentali.

## 2.3 CARTELLA “DATI DIOGENE”

Copiare la cartella config/Dati\_Diogene In un path del server application.

Tale path verrà poi configurato nella variabile `dir.files.datiDiogene` di AMProfiler.

## 2.4 CONFIGURAZIONE SERVER JBOSS

NOTA:

Con la dicitura `[$JBoss_HOME]` si intende il percorso `/ jboss-as-7.1.1.Final /`

Con la dicitura `[$_server]` si intende il percorso `/ jboss-as-7.1.1.Final /standalone/`

### 2.4.1 IMPOSTAZIONE OPZIONI JVM

NOTA:

Le istruzioni sono per sistemi windows, per sistemi linux/unix il file da configurare è `standalone.conf` e `standalone`

Editare il file `standalone.conf.bat` ed individuare le righe nelle quali viene impostata la variabile `JAVA_HOME` ed indicare la path nella quale è stata installata al punto 2.2

Se si volessero variare i parametri di memoria di partenza utilizzati da JBOSS si può modificare l'impostazione della variabile `JAVA_OPTS` presente in una delle ultime righe del suddetto file.

Editare il file `standalone.bat` e modificare la path di installazione di JBOSS, variabile `'TMP_FOLDER'`; quest' ultima tende ad occupare molto spazio disco per cui al riavvio di JBOSS viene cancellata.

## 2.4.2 CREAZIONE SERVIZIO WINDOWS

Se si vuole che JBOSS parta come servizio si può utilizzare il comando '**service install**' presente nella cartella [%JBOSS\_HOME] \bin

## 2.4.3 ISTANZE MULTIPLE DI JBOSS 7 SU STESSO SERVER

Si possono avere più istanza di JBOSS nello stesso server previa modifica di alcuni parametri nel file 'standalone.xml' presente nella cartella [%JBOSS\_HOME] \standalone\configuration. Nello stesso file si possono modificare le porte utilizzate da JBOSS nel caso in cui vadano in conflitto con altre applicazioni presenti nello stesso server.

## 2.4.4 CONFIGURAZIONE DATASOURCE

deployati all'interno del server jboss ([%JBOSS\_HOME] /standalone/deployments): di seguito vengono riportati uno per uno i file di configurazione dei datasources delle varie base dati necessarie.

### Definizione ds multiente

**oracle-AmProfiler-ds.xml**

**oracle-REngine-ds.xml**

Entrambe gli script di definizione ds vanno configurati indicando i parametri di connessione adeguati (un solo datasource per ogni file).

### Definizione *ds custom* per singolo ente:

**oracle-Diogene-ds.xml**

datasource si deve riferire ad una connessione diversa appartenente ad uno specifico ente

### Definizione ds condivisi:

**oracle-CS-ds.xml**

Ogni datasource si deve riferire alla medesima connessione. Questo è il caso di più enti che condividono uno stesso schema.

Lo schema CS è infatti unico per più enti.

Definizione *file properties* che collega i singoli enti ed i corrispettivi numeri come *datasource*:  
[%JBOSS\_HOME] /standalone/configuration /datarouter.properties

Il file properties deve essere impostato per gestire le relazioni BELFIORE/n. datasource.

## 2.4.5 CONFIGURAZIONE LOGGING

Per la gestione del log bisogna intervenire sulle impostazioni definite nel file [\${JBoss\_HOME}]/standalone/configuration/standalone.xml

NOTA:

Il JBoss viene già fornito preconfigurato per quanto riguarda i log.

Le impostazioni definite consentono di suddividere i log per applicazione, potendo scegliere anche il livello da INFO a DEBUG.

Will Output Messages Of Level		DEBUG	INFO	WARN	ERROR	FATAL
Logger Level	DEBUG					
	INFO					
	WARN					
	ERROR					
	FATAL					
	ALL					
OFF	OFF					
	OFF					

Il livello DEBUG consente di vedere tutti i tipi di messaggio, il livello INFO esclude dall'appendere i messaggi di tipo DEBUG e così via. Se si vuole che la piattaforma generi file di log leggeri e con le sole informazioni su eventuali errori, settare gli appender con il livello WARN o ERROR.

## 2.4.6 ULTERIORI OPZIONI DI CONFIGURAZIONE

### 2.4.6.1 CONFIGURAZIONE JBossWS

Configurazione utile a controllare l'indirizzo che il server espone ai client per raggiungere l'endpoint del servizio.

Valorizzare il tag wsdl-host contenuto nel file standalone.xml inserendo l'ip della macchina application come segue:

```
<wsdl-host>172.16.2.117</wsdl-host>
```

### 2.4.6.2 BINDING PORTE SERVER

Il server viene già fornito con binding 0.0.0.0 su interface pubbliche e con 127.0.0.1 sull'interfaccia di management.

E' possibile modificare tali impostazioni editando il file  
[JBOSS\_HOME]/standalone/configuration/standalone.xml

## 3. Deploy Applicazioni

### 3.1 PROCEDURA DI DEPLOY

Per la procedura di deploy si assume che l'archivio da installare sia fornito nel pacchetto setup.

Per deploy si intende l'installazione delle varie applicazioni e/o servizi all'interno del server JBOSS nella cartella [JBOSS\_HOME] /standalone/deployments

La procedura di deploy di un archivio ear, war, etc., viene eseguita semplicemente copiando l'archivio stesso, dentro la cartella deploy del JBOSS: se il server JBOSS è avviato, automaticamente l'archivio copiato viene processato, validato e se non ci sono errori, l'applicazione viene startata. Se il server JBOSS non è attivo, è sufficiente avviarlo; si consiglia comunque sempre il riavvio.

Il server JBOSS durante la procedura di deploy verifica tutte le dipendenze che l'applicazione richiede, ad esempio la presenza di un datasource a cui si connette o la presenza di un determinato ejb o servizio attivo nel server stesso; l'eventuale assenza di una sola dipendenza richiesta causa l'immediata segnalazione ed errore nel processo di deploy.

Se ciò si verifica è necessario verificare l'entità e la causa del problema, intervenendo dove necessario solo dopo aver arrestato il server; solo dopo è possibile eseguire un nuovo deploy dell'applicazione.

Avvertenza:

**Tutte le impostazioni e configurazioni interne all'ear o war sono fatte in fase di creazione e non in fase di deploy; l'archivio da deployare è consegnato già configurato in base alle esigenze e all'ambito di installazione.**



## 3.2 AGGIORNAMENTO APPLICAZIONI

Per eseguire il processo di aggiornamento di un'applicazione è necessario arrestare il server JBOSS, assicurandosi che nessun processo sia in esecuzione, aggiornare l'archivio nella cartella deploy e riavviare.