



Applicativo SBNWeb

Installazione dell'applicativo SBNWEB in ambiente Linux

Compilato	: Alessandro Fusco;Simone
Data Compilazione	: 19 ottobre 2009
Versione	: 1.2.9
Aggiornato	: Alessandro Fusco;Simone
Data	: 22 giugno 2018
Distribuito a	: ICCU



Applicativo SBNWeb

Installazione dell'applicativo SBNWEB in ambiente Linux

Versione : 1.2.9
Data : 22 giugno 2018
Distribuito a : ICCU

INDICE

1. PREMESSA.....	1
2. INSTALLAZIONE E CONFIGURAZIONE DELL'APPLICATIVO SBNWEB.....	2
2.1. Installazione dei moduli EAR dell'applicativo SBNWEB	2
2.2. Installazione dei moduli per export UNIMARC.....	2
2.2.1. Struttura delle cartelle di lavoro	2
2.2.2. Installazione e configurazione del modulo	3
2.3. Creazione del Data Base.....	4
2.4. Configurazione dell'applicativo	6
2.4.1. Personalizzazione della configurazione di JBOSS.....	9
2.4.2. Configurazione delle categorie di esecuzione delle elaborazioni differite.....	11
2.5. Installazione dei moduli per import Utenti Lettori.....	13
2.5.1. Struttura delle cartelle di lavoro	14
2.5.2. Installazione e configurazione del modulo	14
3. DEFINIZIONE E IMPOSTAZIONE DELLE POLITICHE DI BACKUP.....	15
3.1. Storicizzazione	15
3.2. Oggetti del backup	15
3.2.1. Il backup della base dati	15
3.2.2. Il backup dell'ambiente	15
3.2.3. Il backup su nastro	16
3.3. Schedulazione	16
4. CONCLUSIONE ATTIVITÀ	18
4.1. Le shell di backup	18
4.2. La shell per l'aggiornamento del software applicativo.....	18
4.3. Gli script di startup	18
4.4. Ambiente grafico	18
4.5. Pulizia dei log di JBOSS.....	19
4.6. Autenticazione alla console di JBOSS.....	19
ALLEGATO A.....	21
ALLEGATO B.....	24

1. PREMESSA

Obiettivo del presente documento è descrivere le operazioni che occorre eseguire per l'installazione dell'applicazione SBNWEB e la sua configurazione.

La descrizione non comprende le attività relative alla configurazione del sistema operativo e dell'ambiente di base del server (definizione dei file-system, spazio disco ecc.), per le quali si rimanda all'apposito documento "Predisposizione di un server LINUX all'installazione dell'applicativo SBNWEB".

Si tenga altresì conto che tutti i riferimenti a file system e/o direttori in cui posizionare le componenti che costituiscono l'applicativo assumo che siano state rispettate le indicazioni fornite nel documento "Predisposizione di un server LINUX all'installazione dell'applicativo SBNWEB".

In attesa della realizzazione di un sistema di distribuzione delle componenti che costituiscono l'applicativo SBNWEB, i diversi moduli che lo compongono verranno resi disponibili sotto forma di directory e file compressi all'interno della directory **install_sbnweb** (fornito in formato *tar.gz*), la cui organizzazione è riportata nell'allegato A "Contenuti del file *install_sbnweb*".

Si consiglia di decomprimere il file *install_sbnweb.tar.gz* ed estrarne la directory **install_sbnweb** in una propria directory di lavoro, da cui successivamente prelevare le varie componenti dell'applicativo.

2. INSTALLAZIONE E CONFIGURAZIONE DELL'APPLICATIVO SBNWEB

L'applicativo è costituito da tre moduli: due distribuiti in formato Enterprise Archive file (ear) che implementano l'interfaccia web e la logica elaborativa dell'applicativo SBNWEB, un terzo per la produzione degli export UNIMARC costituito di più file di vario tipo.

2.1. Installazione dei moduli EAR dell'applicativo SBNWEB

I due moduli che realizzano l'interfaccia web e la gestione della BD, distribuiti come Enterprise Archive file (ear), sono:

- **sbn.ear** che contiene tutti i moduli web e la logica elaborativa dell'applicativo SBNWEB.
- **sbnMarc.ear** che implementa il protocollo SBNMARC per la gestione del catalogo bibliografico di Polo.

I due EAR vanno copiati nella cartella di deploy di jboss.

Assumendo che jboss sia stato installato al path `/usr/local/jboss-4.2.3.GA`, la cartella dove copiare i file .ear è:

`/usr/local/jboss-4.2.3.GA/server/default/deploy.`

2.2. Installazione dei moduli per export UNIMARC

Nel caso in cui si scelga di utilizzare un'architettura che utilizza server separati per l'application e il data server, l'installazione del modulo esportazione UNIMARC va fatta sul server dove sarà in esecuzione sbn.ear.

2.2.1. Struttura delle cartelle di lavoro

La tabella sottostante riporta l'organizzazione delle cartelle di lavoro necessaria al corretto funzionamento delle componenti che costituiscono il modulo di esportazione del catalogo in formato UNIMARC.

L'esportazione può essere attivata, in modalità differita, da client web, o anche direttamente da connessione ssh al server; in questa seconda ipotesi l'output del processo non sarà visibile da client web, ma va recuperato dalla 'cartella di deposito dei file unimarc' (vedi Tabella 1).

Cartella	Descrizione
/export/exportUnimarc/dp/export	Cartella dove risiedono gli script per la predisposizione: <ul style="list-style-type: none">• dei file sequenziali copia del DB• della lista documenti da estrarre nel caso di esecuzione da terminale
/export/exportUnimarc/db_export	Cartella dove si deposita la copia su files sequenziali delle tabelle del DB
/export/exportUnimarc/dp	Cartella dove risiedono gli script che eseguono: <ul style="list-style-type: none">• la creazione degli indici dei file sequenziali• la creazione del file unimarc
/export/exportUnimarc/dp/input	Cartella dove vengono generati gli indici
/export/exportUnimarc/dp/unimarc	Cartella dove viene generato il file animarci di output
/export/exportUnimarc/dp/it/finsiel/misc	Cartella dove risiedono le classi Java per funzioni di utilità
/export/exportUnimarc/dp/it/finsiel/offlineExport	Cartella dove risiedono le classi Java che realizzano la logica di estrazione, indicizzazione e conversione

Tabella 1 - struttura delle cartelle per modulo esportazione UNIMARC

2.2.2. Installazione e configurazione del modulo

Copiare nella cartella **/export** il file `exportUnimarc.template.codpolo.zip` contenuto all'interno del file `install_sbnweb`.

Eseguire l'unzip del file: l'operazione creerà automaticamente la struttura (directory) **exportUnimarc.template** con all'interno tutto l'ambiente già configurato per l'export Unimarc del polo.

Quindi procedere con i seguenti passi

- eseguire il comando **`mv exportUnimarc.template exportUnimarc`**
- eseguire il comando **`chown -R export:jboss exportUnimarc`**
- creare la directory `/export/exportUnimarc/dp/export/trasf` e assegnarle i proprietari `export:export` (con l'opzione `-R`) e i permessi `777`
- rimuovere il file `.zip`.
- nel caso si disponga anche del server OPAC, nella shell `/export/exportUnimarc/dp/export/TRASF.sh`, sostituire alla stringa `xxx.xxx.xxx.xxx` l'indirizzo IP del server OPAC, assicurandosi che abbia i permessi di esecuzione (755)

Si noti che il file `.zip`, riferendosi ad un ambiente elaborativo strettamente legato al polo, viene preliminarmente configurato in fase di assemblaggio del kit di installazione, assegnando alle seguenti variabili i valori di riferimento del polo in questione:

Variabile	Valore di template	Adeguamento
polo	PPP	Sostituire PPP con la sigla del codice polo
descPolo	"Polo PPP"	Sostituire con una descrizione del Polo mantenendo i doppi apici
bibliotecaRichiedenteScarico	IT-000000	Sostituire la parte numerica con il codice anagrafe della biblioteca di riferimento del Polo

Ad altre variabili, di seguito riportate, vengono assegnati i valori di default, a meno che non siano stati precedentemente comunicati, dai responsabili del polo, valori alternativi (e comunque modificabili successivamente nel file `offlineExportUnimarc.linux.cfg` sotto `/export/exportUnimarc/dp/`):

Digita l'indirizzo IP del server che ospita il DB [Default è localhost]:

Digita la porta dove il DB sta in ascolto [Default è 5432]:

Digita il nome del database [Default è sbnwebDb]: sbnwebDbPPPprv

Si vuole un record unimarc su una riga singola? (S/N) [Default è N]:

Vuoi l'etichetta 314? (Si/No) [Default è Si]:

Vuoi l'etichetta 326? (Si/No) [Default è Si]:

Vuoi l'etichetta 440? (Si/No) [Default è Si]:

Vuoi l'etichetta 801 o 850? (801/850) [Default è 801]:

Vuoi l'etichetta 95x o 96x? (95x/96x) [Default è 95x]:

Come si vede, viene fornito come nome di database, quello utilizzato per la migrazione di prova; in caso il polo non utilizzasse la suddetta migrazione, il file `.zip` sarà creato con il nome del database di esercizio: `sbnwebDb<codpolo>ese`.

In ogni caso, in `/export/exportUnimarc/dp/export` modificare all'occorrenza i files **download<polo>.txt**, **downloadFileAccessori<polo>.txt** e **getBidList.txt** adeguando opportunamente i valori assegnati alle variabili preposte all'accesso al DataBase:

- **connectionUrl**=`jdbc:postgresql://localhost:5432/sbnwebDbXXXese/prv`
usando il suffisso `ese` o `prv` a seconda del database che si intende usare
- **userName**=`sbnweb`
- **userPassword**=`sbnweb`

2.3. Creazione del Data Base

Come utente postgres eseguire la seguente procedura:

1. Creare le cartelle per i tablespace

```
mkdir /db/pgsql8.3.5/data/pg_tblspc/tbs_indici_ese
```

```
mkdir /db/pgsql8.3.5/data/pg_tblspc/tbs_dati_ese
```

dove **tbs_indici_ese** e **tbs_dati_ese** sono solo nomi suggeriti; se però si sceglie di cambiarli allora il nome adottato va sostituito negli sql di create del DB (file *Create_DB_XXXese.sql*) e in quello di creazione degli indici (*sbnwebDbXXX_Completo.sql*).

2. Creare il data base: copiare dal file *install_sbnweb* (sottodirectory *DB_creaDB*), il file *Create_DB_XXXese.sql* in una cartella di lavoro e adeguare i nomi dei tablespace e del database, quindi lanciare il seguente comando

```
psql -L Create_DB_XXXese.log -f Create_DB_XXXese.sql
```

e verificare nel file *Create_DB_XXXese.log* che i tablespace e il database siano stati correttamente istanziati.

N.B.: nel caso sia prevista la migrazione di prova, i passi 1 e 2 vanno eseguiti di nuovo sostituendo nei nomi dei tablespace e in quello del database (anche all'interno degli eseguibili sql) il suffisso ese con il suffisso prv.

3. Generare le strutture dati: sempre dalla sottodirectory *DB_creaDB* di *install_sbnweb*, copiare nella cartella di lavoro il file *sbnwebDbXXX_Completo.sql* e, se è stato modificato, adeguare il nome del tablespace predisposto per gli indici, quindi eseguire il seguente comando:

```
psql -d <nomedb> -L sbnwebDbXXX_Completo.log -f sbnwebDbXXX_Completo.sql
```

In alternativa, copiare nella cartella di lavoro il file che crea la struttura del database senza indici *sbnwebDbXXXeseSenzaIndici.sql* ed eseguire il seguente comando:

```
psql -d <nomedb> -L sbnwebDbXXXeseSenzaIndici.log -f sbnwebDbXXXeseSenzaIndici.sql
```

4. Caricare nel DB i dati di inizializzazione: l'attività è finalizzata al caricamento nel DB dei valori codificati, e dei codici di funzione che controllano il comportamento dell'applicativo, nonché alla creazione dell'utente 'root'; le occorrenze da inserire si distinguono in due categorie: a valori predefiniti, e a valori da personalizzare per ciascun polo.

- 4.1. Per le occorrenze che non necessitano di personalizzazione copiare nella cartella di lavoro i file di seguito elencati:

```
00_sbnwebDBinsert_ts_stop_list.sql
```

```
01_sbnwebDBinsert_tbf_attivita_sbnmarc-tbf_attivita.sql
```

```
02_sbnwebDBinsert_tb_codici.sql
```

```
03_sbnwebDBinsert_X_Batch_e_Default.sql
```

```
04_sbnwebDBinsert_tbf_parametro.sql
```

E per ciascuno di essi, avendo cura di elaborare i file nell'ordine della numerazione riportata nel *nomefile*, eseguire il comando:

```
psql -d <nomedb> -L nomefile.log -f nomefile.sql
```


- 4.2. Per le occorrenze che necessitano di personalizzazione copiare nella cartella di lavoro i file di seguito elencati:

05_Definisci_Polo.sql

06_sbnwebDBinsert_AbilitaPolo_UtenzaRoot.sql

Adeguare opportunamente il file *05_Definisci_Polo.sql* nel quale sono indicati i dati di identificazione del polo e le credenziali per la connessione al server INDICE; a tale scopo sostituire i valori:

'PPP'	con	'Codice del Polo'	3 chr
'XXXAMMnnnnnn'	con	'username fornita da Indicè	12 chr
'pwdindicè	con	'password fornita da Indicè	varchr
'mailgestorepolo'	con	'indirizzo e-mail del gestore'	varchr

quindi per ciascuno file della lista al p.to 4.2, avendo cura di elaborare i file nell'ordine della numerazione riportata nel loro nome, eseguire il comando:

psql -d <nomedb> -L nomefile.log -f nomefile.sql

Si precisa che il template viene fornito con il puntamento all'Indice di collaudo

"url_indice"='http://193.206.221.21/indice/servlet/serversbnmarc'

Per quanto riguarda campo "url_polo", nel template è valorizzato ipotizzando che l'esecuzione di ambedue i moduli .ear sia controllata dalla stessa istanza di application server e che questo sia in ascolto sulla porta 8080, se le scelte del polo fossero diverse il valore di tale campo va opportunamente adeguato all'effettivo url dell'application server che esegue il modulo sbnmarc.ear.

"url_polo"='http://localhost:8080/SbnMarcWeb/SbnMarcTest'

Si precisa inoltre che per rendere operative variazioni apportate al record inserito nella tabella tbf_polo è richiesto il restart dell'applicativo sbn.ear.

2.4. Configurazione dell'applicativo

Di seguito sono riportate le istruzioni per verificare/adequare i parametri che realizzano la configurazione dell'applicativo SBNWEB.

Il file in cui tali parametri sono definiti è **sbnweb.conf** (in install_sbnweb/jboss-configuration).

Prima di tutto editare il file, sostituendo nelle ultime due istruzioni il codice polo (in maiuscolo) alle XXX presenti nei nomi dei file di integrazione.

Per il corretto funzionamento dell'applicativo, **sbnweb.conf** va posizionato in una cartella accessibile all'utente jboss (si consiglia il posizionamento nella cartella puntata dalla variabile \$HOME di jboss, /home/jboss), quindi il path scelto va referenziato nel file di configurazione di jboss **run.conf** (per ulteriori informazioni vedi par. 2.4.1)

19 ottobre 2009

Installazione dell'applicativo SBNWEB in ambiente Linux

Parametri predefiniti nel file **sbnweb.conf**:

Nome Parametro	Default	Note
DATA_SOURCE	java:jdbc/sbnHibernatePostgresJDS	Nome JNDI del data source usato per le query in SQL (non Hibernate). Deve essere lo stesso al quale punta il file SbnWebPostgres-ds.xml.
QUEUE_NAME	queue/sbnWebBlocchi	Nome della coda JMS usata per memorizzare le sintetiche paginate. La coda è configurata nel file sbnWeb-queue-service.xml (vedi paragrafo 2.4.1).
MESSAGE_TTL	21600000	Durata massima (in millisecondi) del mantenimento di un blocco non richiesto dall'utente.
SBNWEB_BATCH_FILES_PATH	/home/jboss/sbn/download	Path assoluto in cui verranno salvati gli output e i log delle elaborazioni differite.
SBNWEB_EXPORT_UNIMARC_HOME	/export/exportUnimarc/dp	Path assoluto alla cartella di installazione della procedura di Export Unimarc offline.
EXPORT_UNIMARC_FILE_ACCESSORI		Elenco di file (separati da punto e virgola) che, se trovati nella cartella unimarc del path SBNWEB_EXPORT_UNIMARC_HOME , verranno inseriti nel file .zip output della funzione export Unimarc
CLONER_CLASS	it.iccu.sbn.util.cloning.impl.JDKActualCloner	Classe standard che implementa il sistema di clonazione degli oggetti di scambio usati dall'applicativo.
CLONER_POOL_SIZE	5	Numero di istanze concorrenti delle classi indicate dal parametro CLONER_CLASS.
JBAC_BUFFER_SIZE	65536	Dimensione (in bytes) del buffer per la serializzazione degli oggetti clonati.
BATCH_CLEANING_AGE_THRESHOLD	30	Numero di giorni dopo il quale il sistema elimina i riferimenti ad una elaborazione differita. Se posto uguale a 0 il sistema non cancella mai alcun riferimento.
BATCH_CLEANING_DELETE_OUTPUTS	true	Indica se il sistema cancella automaticamente i file di output salvati nella cartella indicata dal parametro SBNWEB_BATCH_FILES_PATH .
BATCH_USER_DELETE_OUTPUTS	true	Indica se una richiesta puntuale di cancellazione elimina anche i file di output salvati nella cartella indicata dal parametro SBNWEB_BATCH_FILES_PATH .
LOG_LEVEL_SBNWEB LOG_LEVEL_SBNMARC LOG_LEVEL_HIBERNATE	DEBUG	Livello di dettaglio dei log prodotti dall'applicativo.
SBNWEB_HTTP_SESSION_TIMEOUT SERVIZI_HTTP_SESSION_TIMEOUT	30 5	Timeout (in minuti) della sessione HTTP dell'applicativo gestionale e del modulo Servizi.
SIP2_BIND_PORT	7892	Porta di ascolto del servizio di auto-prestito SIP2.
SIP2_CURR_POLE	Non impostato	Codice Polo assegnato al totem SIP2.

19 ottobre 2009

Installazione dell'applicativo SBNWEB in ambiente Linux

Nome Parametro	Default	Note
SIP2_CURR_BIBLIO	Non impostato	Codice Biblioteca assegnato al totem SIP2.
MAX_RESULT_ROWS	4000	Limite al numero di righe restituite dalla componente SBNMARC di Polo.
SMS_PROVIDER_CLASS	it.iccu.sbn.util.sms.impl.DummySMSProvider	Classe che implementa il servizio personalizzato per l'invio di messaggi SMS.
SBNMARC_INDICE_USE_PROXY	False	Attiva la gestione della connessione via proxy HTTP al sistema Indice SBNMARC.
SBNMARC_INDICE_PROXY_URL SBNMARC_INDICE_PROXY_PORT SBNMARC_INDICE_PROXY_USERNAME SBNMARC_INDICE_PROXY_PASSWORD	Non impostati	Parametri per la connessione al proxy HTTP server.
SBNMARC_INDICE_ALLINEA_REQUEST_TIMEOUT	3600000	Durata massima (in millisecondi) dell'attesa per una risposta dall'Indice SBNMARC durante un'elaborazione di Allineamento.
RFID_ENABLE	false	Se impostato a true abilita la visualizzazione del campo RFID per la lettura automatica della chiave inventario. Per dettagli sul formato ammesso per i dati immessi nel campo si veda l'ALLEGATO B
CSV_FIELD_SEPARATOR	virgola (\u002C)	Separatore di default per i file prodotto nel formato csv.
SBNWEB_EXPORT_IGNORE_FILE	Non impostato	Percorso del file opzionale contenente la lista di BID da ignorare durante l'esportazione Unimarc.
ESAME_COLLOCAZIONI_MAX_RESULT_ROWS	1000	Numero massimo di righe da precaricare durante le interrogazioni della linea "Esame Collocazioni".
WS_MAX_CONCURRENT_CLIENTS	10	Numero massimo di client ammessi sul modulo web-service.
SBNWEB_IMPORTA_UTENTI_HOME	/home/jboss/sbn/lettori	Path assoluto alla cartella di installazione della procedura di importazione utenti lettori.
LOC_MASSIVA_INDICE_WAIT_TIMEOUT	30000	Relativo al processo differito 'Localizzazione massiva' (IA006). Durata (in millisecondi) dell'intervallo tra una richiesta di localizzazione in Indice e la successiva.
SBNMARC_SCHEMA_VERSION	1.16	Versione dello schema SBNMARC supportata dall'applicativo.
INDICE_CONNECTION_CHECKER_TIMEOUT	300000	Durata (in millisecondi) dell'intervallo tra una richiesta di verifica della disponibilità dell'Indice e la successiva (solo nel caso in cui il flag di connessione non disponibile sia attivo).
SBNMARC_ALLINEAMENTO_SU_FILE_INTERVAL	3600000	Intervallo (in millisecondi) impostato per l'inserimento automatico di una nuova richiesta di allineamento a fronte di una risposta di Indice per allineamento da file. Se assente il sistema assume il valore di default (attivazione dopo un'ora dalla risposta dell'Indice) se presente, ma valorizzato a zero, la prenotazione automatica non viene effettuata.
SBNWEB_BATCH_ALLINEAMENTO_DA_LOCALE		Path assoluto alla cartella dove saranno salvati i file XML prodotti dal processo differito "Salvataggio file XML Indice per allineamento" (IA007).

Nome Parametro	Default	Note
SRV_PRENOTAZIONE_POSTO_UTENTI_MULTIPLI	false	Se impostato a true permette di aggiungere più utenti a una singola prenotazione posto (a sistema il posto occupato sarà il medesimo).
ILL_SBN_SERVER_URL		URL che specifica il punto di ascolto del Server SBN ILL Nazionale.
ILL_SBN_DOC_DELIVERY_URL		URL che specifica il punto d'ascolto per il servizio di document-delivery del Server SBN ILL.
OPAC_Z3950_SEARCH_ENABLE OPAC_Z3950_URL OPAC_Z3950_DB	false opac.sbn.it:3950 nopac	Parametri di configurazione per l'accesso all'OPAC di Indice SBN o altro OPAC che colloquia secondo il protocollo Z39.50.
SRV_EVENTO_ACCESSO_MERGE_THRESHOLD	60000	intervallo minimo (in millisecondi) tra due eventi di accesso per lo stesso utente. Gli eventi che ricadono nell'intervallo saranno fusi sul primo evento registrato.

2.4.1. Personalizzazione della configurazione di JBOSS

- Configurazione del profilo di jboss

Nel file **.bash_profile** nella directory `/home/jboss`, aggiungere le seguenti righe:

```
PATH=/usr/local/jdk/bin:$PATH:$HOME/bin (prima della riga "export PATH")
```

```
export DISPLAY=localhost:1
```

- Path della configurazione di sbnweb

Nel file `/usr/local/jboss-4.2.3.GA/bin/run.conf` aggiungere l'opzione `sbnweb.conf.path` per la JVM:

```
JAVA_OPTS="$JAVA_OPTS -Dsbnweb.conf.path={path}"
(JAVA_OPTS="$JAVA_OPTS -d64 -Dsbnweb.conf.path={path}" se l'installazione avviene in modalità 64 bit)
```

dove `{path}` indica il percorso dove è situato il file **sbnweb.conf** (`/home/jboss`).

Qualora il server non disponga di un terminale video, aggiungere in fondo alla precedente istruzione, prima del doppio apice di chiusura, il parametro

```
-Djava.awt.headless=true
```

Quindi settare la variabile `JAVA_HOME` come segue:

```
JAVA_HOME=/usr/local/jdk
```

Infine, nella sezione "Specify option to pass to the Java VM", sostituire l'istruzione presente all'interno del controllo "if" con la seguente (tutta su un'unica riga):

```
JAVA_OPTS="-Xms512m -Xmx1280m -XX:MaxPermSize=512m -Dsun.rmi.dgc.client.gcInterval=3600000  
-Dsun.rmi.dgc.server.gcInterval=3600000"
```

- Attribuzione del localhost

Nel file `/usr/local/jboss-4.2.3.GA/bin/run.sh` aggiungere l'opzione `-b 0.0.0.0` nelle due sezioni *"Execute the JVM in the foreground/background"*, come mostrato di seguito:

```
org.jboss.Main -b 0.0.0.0 "$@"      per la sezione foreground  
org.jboss.Main -b 0.0.0.0 "$@" &   per la sezione background
```

- Durata massima delle transazioni per elaborazioni differite

Nel file `/usr/local/jboss-4.2.3.GA/server/default/conf/jboss-service.xml` portare il timeout della singola transazione a 86400 secondi (24 ore) modificando opportunamente il valore dell'attributo `"TransactionTimeout"`:

```
<!-- JBoss Transactions JTA -->  
<mbean code="com.arjuna.ats.jbossatx.jta.TransactionManagerService"  
  name="jboss:service=TransactionManager">  
  <attribute name="TransactionTimeout">86400</attribute>  
  <attribute name="ObjectStoreDir">${jboss.server.data.dir}/tx-object-store</attribute>  
</mbean>
```

- Trattamento URL con codifica UTF-8

L'adeguamento va operato per consentire il corretto trattamento di eventuali caratteri speciali presenti nelle attivazioni via URL (es. attivazione del modulo servizi da OPAC).

Modificare connettore HTTP del sottosistema Tomcat usato da JBOSS. Nel file `/usr/local/jboss-4.2.3.GA/server/default/deploy/jboss-web.deployer/server.xml` aggiungere parametro `URIEncoding` al connettore HTTP (tag `<Connector>`):

```
<Connector port="8080" address="${jboss.bind.address}"  
  maxThreads="250" maxHttpHeaderSize="8192"  
  emptySessionPath="true" protocol="HTTP/1.1"  
  enableLookups="false" redirectPort="8443" acceptCount="100"  
  connectionTimeout="20000" disableUploadTimeout="true" URIEncoding="UTF-8" />
```

- Configurazione dell'accesso al DataBase

L'ultimo passo riguarda i due file che contengono le informazioni per il corretto puntamento al DB da parte delle due componenti applicative. I file template sono forniti nel kit di installazione (`install_sbnweb/jboss-configuration/server/default/deploy`) con i nomi:

`SbnMarcPostgres-ds.xml` (data source per il modulo `sbnmarc.ear`)
`SbnWebPostgres-ds.xml` (data source per il modulo `sbn.ear`)

I due files vanno posizionati nella cartella

/usr/local/jboss-4.2.3.GA/server/default/deploy

Vanno quindi personalizzati riportando in <connection-url> l'URL del DataBase di polo.

```
<connection-url>jdbc:postgresql://localhost:5432/nomedb</connection-url>
```

Eventualmente va adeguata anche le password delle credenziali di accesso al DB.

```
<password>sbnweb</password>
```

Nell'esempio riportato l'url del DB è valorizzato ipotizzando che application server e data server rispondano allo stesso ip address e che l'RDBMS sia in ascolto sulla porta 5432, se le scelte adottate del polo fossero diverse il valore di tale campo va opportunamente adeguato sostituendo localhost con l'ip address del data server.

- Configurare le code JMS per l'utilizzo con l'applicativo SBNWEB:

Nella cartella */usr/local/jboss-4.2.3.GA/server/default/deploy/jms:*

- sostituire **hsqldb-jdbc2-service.xml** (serializzazione code in memoria) con il file **postgres-jms-sbnweb-service.xml** (serializzazione code su DB) presente nel pacchetto di installazione (install_sbnweb/jboss-configuration/server/default/deploy/jms);
- copiare il file di configurazione **sbnWeb-queue-service.xml** (configurazione personalizzata delle code). Il file contiene la configurazione minima per il corretto funzionamento dell'applicativo SBNWEB.

A seguire un elenco delle code definite dall'installazione standard:

Nome Coda	Valore	Note
sbnMarcBlocchi	NON MODIFICARE	Coda che memorizza i blocchi che compongono le sintetiche prodotte dal protocollo SbnMarc di Polo.
sbnWebBlocchi	NON MODIFICARE	Coda che memorizza i blocchi che compongono le sintetiche prodotte dall'applicativo SBNWEB.
sbnBatch_Input	NON MODIFICARE	Coda usata per inserire a sistema una nuova prenotazione di elaborazione differita.
sbnBatch_Output1 sbnBatch_Output2		Code definite per contenere i riferimenti agli output delle elaborazioni differite. Definire più code di output può essere utile per rendere più veloce l'interrogazione delle stesse da applicativo, nel caso si sia scelto di non effettuare la cancellazione periodica degli output delle elaborazioni differite (parametro BATCH_CLEANNING_AGE_THRESHOLD = 0).

2.4.2. Configurazione delle categorie di esecuzione delle elaborazioni differite

L'installazione preliminare di un DB SBNWEB prevede una configurazione funzionante delle code destinate alle elaborazioni differite, raggruppate in un'unica categoria d'esecuzione sincrona di tipo IMMEDIATE (attivazioni di tipo seriale, 24 ore su 24).

Quanto segue permette una suddivisione del carico elaborativo su più categorie di esecuzione, rendendo possibile la discriminazione dei processi per evitare il sovraccarico del server e permettere l'elaborazione di più richieste (di tipologie diverse) contemporaneamente.

NB: Un processo differito, identificato dal suo codice attività, può appartenere a una sola categoria d'esecuzione. È possibile definire un numero illimitato di categorie, ma quelle che non hanno nessuna procedura assegnata non appariranno nella maschera di ricerca dell'applicativo SBNWEB.

Per rendere visibili all'applicativo SBNWEB le variazioni apportate alla configurazione delle code è richiesto il **riavvio** del server JBOSS.

- Definizione di una nuova categoria di esecuzione

La tabella **tbf_coda_jms** contiene la definizione delle singole categorie. È possibile definire una nuova categoria semplicemente inserendo una nuova riga nella tabella. A seguire una descrizione dei campi che compongono la tabella in oggetto:

Nome Campo	Default	Note
id_coda	Progressivo calcolato dal sistema	ID univoco della categoria di esecuzione
nome_jms	queue/sbnBatch_Input	Nome della coda JMS usata per inserire a sistema una nuova prenotazione di elaborazione differita. Attenzione: La coda JMS usata come default è attualmente l'unica supportata a questo scopo dall'applicativo SBNWEB.
sincrona	S	Indica che solo un processo di elaborazione differita, tra quelli assegnati a questa categoria, potrà essere elaborato dal sistema in un dato istante (stato EXEC).
id_descrizione		Nome assegnato alla categoria e che apparirà nelle maschere dell'applicativo SBNWEB.
id_descr_orario_attivazione		NON UTILIZZATO
Id_orario_di_attivazione		NON UTILIZZATO
cron_expression	0/30 * * * * ?	Espressione in sintassi unix cron che permette la personalizzazione degli orari di attivazione e disattivazione della categoria d'esecuzione. Utilizza lo scheduler open-source Quartz. Esempio: 0 0/5 14 10-15 * 2010 Controlla la presenza di una prenotazione al primo secondo ('0'), ogni 5 minuti ('0/5'), solo dalle 14 alle 15 ('14'), dal 10 al 15 del mese ('10-15'), ogni mese ('*') del 2010 ('2010') Per ulteriori informazioni consultare il sito del produttore: http://www.quartz-scheduler.org/docs/tutorials/crontrigger.html

NB: L'espressione contenuta nel campo **cron_expression** indica la modalità con cui l'applicativo interrogherà la coda di input per verificare sia la presenza di una prenotazione in attesa che di un processo in esecuzione (nel caso si sia configurata la categoria come sincrona). Nel caso si sia scelto di attivare numerose categorie è opportuno configurare le rispettive espressioni cron per evitare troppi accessi concorrenti alle code JMS che potrebbero rallentare il server.

Esempio di configurazione:

categoria 1	0/30 *****?	la coda viene interrogata ogni 30 secondi, al secondo 0 e al secondo 30.
categoria 2	15/45 *****?	la coda viene interrogata ogni 30 secondi, al secondo 15 e al secondo 45.
categoria 3	25/55 *****?	la coda viene interrogata ogni 30 secondi, al secondo 25 e al secondo 55.
categoria 4	0 0/5 ***?	La coda viene interrogata ogni 5 minuti, al secondo 0.

- Assegnazione di un processo differito alla categoria d'esecuzione

I processi disponibili come procedure differite sono descritti e configurati nella tabella **tbf_batch_servizi**.

NB: nella sua installazione base la tabella contiene già tutti i processi previsti dall'applicativo SBNWEB. Eliminare un record dalla tabella renderà il processo cancellato non più utilizzabile.

A seguire una descrizione dei campi che compongono la tabella in oggetto:

Nome Campo	Default	Note
id_batch_servizi	Progressivo calcolato dal sistema	ID univoco del processo di elaborazione differita
cd_attivita	NON MODIFICABILE	Codice Attività univoco che identifica a livello di applicativo il processo di elaborazione differita. Questo valore è riportato nei profili di abilitazione dei bibliotecari e ne influisce la prenotabilità.
id_coda_input	Riferimento al campo id_coda della tabella tbf_coda_jms	Categoria d'esecuzione a cui è assegnato questo processo di elaborazione differita.
nome_coda_output	Definito nel file sbnWeb-queue-service.xml	Nome della coda JMS dove saranno salvati i dettagli riguardanti le prenotazioni e l'esecuzione del processo di elaborazione differita. La coda deve essere definita a livello di server e configurata nel file sbnWeb-queue-service.xml .
class_name	NON MODIFICABILE	Percorso completo della classe Java che realizza il servizio indicato da questo processo di elaborazione differita.
visibilita	P	Parametro che indica se, in fase di interrogazione da applicativo SBNWEB, i dettagli riguardanti questo processo di elaborazione differita saranno visibili a tutti gli utenti del polo (visibilita=P) oppure solo a quelli della biblioteca che effettua la prenotazione del processo (visibilita=B)
fl_canc	NON MODIFICABILE	Flag di cancellazione logica.

2.5. Installazione dei moduli per import Utenti Lettori

Il paragrafo contiene la descrizione delle operazioni necessarie per installare la procedura di importazione o aggiornamento dei dati relativi agli utenti lettori e alle corrispondenti autorizzazioni in SBNWEB.

2.5.1. Struttura delle cartelle di lavoro

La procedura viene attivata, in modalità differita, da client web. Può anche essere attivata direttamente da connessione SSH al server. In questa seconda ipotesi, è necessaria l'esecuzione dello script "**importaUtenti.sh**" a cui devono essere forniti i seguenti parametri:

- Il file di configurazione "**ImportaUtentiLettori.cfg**" da personalizzare per ciascun polo con il codice del polo stesso e i dati relativi al database a cui accedere.
- Il nome del file di input.
- La data di inizio validità delle autorizzazioni da assegnare nel formato dd/mm/yyyy.

2.5.2. Installazione e configurazione del modulo

Copiare nella cartella **/home/jboss/sbn** il file **lettori.zip** contenuto all'interno del file **install_sbnweb**.

Eseguire l'unzip del file: l'operazione creerà automaticamente la struttura (directory) **lettori** con all'interno tutto l'ambiente da configurare con i parametri del polo.

Quindi procedere con i seguenti passi:

- eseguire il comando **chown -R jboss:jboss lettori**.
- rimuovere il file .zip.
- portarsi nella cartella **lettori** e assegnare i permessi di esecuzione alla shell **importaUtenti.sh** eseguendo il comando:
chmod 755 importaUtenti.sh
- modificare il file di configurazione **ImportaUtentiLettori.cfg**, facendo riferimento alla seguente tabella:

Variabile	Valore di template	Adeguamento
codPolo	PPP	Sostituire PPP con la sigla del codice polo.
connectionUrl	jdbc:postgresql://localhost:5432/sbn webDbPPPe	Sostituire con l'indirizzo e il nome del Db associato all'applicativo SBNWEB.
username	sbnweb	Nome utente per l'accesso al Db.
userPassword	sbnweb	Password per l'accesso al Db.

Agli altri parametri vengono assegnati valori di default e non vanno modificati.

3. DEFINIZIONE E IMPOSTAZIONE DELLE POLITICHE DI BACKUP

Le politiche di backup del server sono state definite in relazione ai tre aspetti più importanti da tenere in considerazione: l'oggetto del backup, per quanto tempo mantenere le informazioni (ossia la sua storicizzazione), la schedulazione.

In generale i backup vengono effettuati su disco; in coda all'intero processo, gli output prodotti sono successivamente trasferiti su nastro.

Vediamo in dettaglio i tre aspetti sopra citati.

3.1. Storicizzazione

Partiamo da questo aspetto della questione dal momento che le scelte effettuate si riflettono nella organizzazione delle operazioni di backup.

In generale, in mancanza di specifiche esigenze al riguardo, lo standard è quella di effettuare un backup giornaliero con ciclo settimanale; in altre parole si sceglie di mantenere le informazioni salvate per il tempo massimo di una settimana.

Solitamente i file di output prodotti dalle procedure riportano nel nome il giorno in cui vengono scritti e pertanto, al termine del ciclo, il file più vecchio viene sovrascritto. Nella maggioranza dei casi si può omettere il backup nel giorno di domenica (vedi più avanti il paragrafo 3.3 sulla schedulazione).

3.2. Oggetti del backup

Possiamo distinguere tali oggetti in due gruppi principali: la base dati e l'ambiente operativo (file-system e singole directory). Le shell si trovano in **/home/SCRIPTS** e l'output prodotto si trova in **/backup/DUMP_DB** (base dati) e **/backup/DUMPSYS** (ambiente).

3.2.1. Il backup della base dati

La shell preposta al backup della base dati è la **pg_dump.sh** che sfrutta una utility di *Postgres*, chiamata appunto *pgdump*, per effettuare un backup logico del database indicato nella relativa istruzione. L'output dell'operazione viene posto in **/backup/DUMP_DB** con la seguente nomenclatura:

<giorno>.<nomedb>.gz

La procedura produce un file di log sotto la directory **/backup/logs**

bckdb-<giorno>.log

che riporta l'esito del backup.

3.2.2. Il backup dell'ambiente

La shell preposta al backup dell'ambiente è la **backup.sh**. Essa, tramite il comando *tar*, effettua il salvataggio delle seguenti strutture:

file-sytsem /usr/local

file-system /home

file-system /etc

file-sytsem /export

L'output prodotto, un file *tar* per ciascuna struttura, è posto in */backup/DUMPSYS* con la seguente nomenclatura:

<nomedir>.<giorno>.tar.gz

mentre i log dell'operazione si trovano sotto */backup/logs* con analoga nomenclatura:

<nomedir>.<giorno>.log

3.2.3. Il backup su nastro

Come detto in precedenza, i backup sopra descritti vengono effettuati su disco. Per ulteriore sicurezza è stato previsto anche un salvataggio ridondante su nastro.

La shell preposta a questa operazione è la **tar.sh** che, tramite il comando *tar*, effettua il salvataggio su nastro (tipicamente delle cartucce DAT) delle seguenti strutture:

file-system /usr/local

file-system /home

file-system /backup

directory /db/pgsql8.3.5

L'output dell'operazione si trova sotto */backup/logs* con nomenclatura:

tar.<giorno>.out

mentre il log che riporta l'esito dell'operazione si chiama:

tar.<giorno>.log

Ovviamente per mantenere la storicizzazione di una settimana anche per i backup su nastro, sarà necessario disporre di un numero adeguato di nastri, opportunamente etichettati con la data del giorno a cui si riferisce il contenuto, che poi dovranno giornalmente essere sostituiti.

3.3. Schedulazione

La schedulazione dell'esecuzione dei backup viene effettuata attraverso la *crontab* di sistema, nella quale vengono stabiliti i giorni e l'ora di esecuzione di ogni singola shell. Generalmente

l'esecuzione delle tre shell sopra descritte viene programmata dal lunedì al sabato, con orari sfalsati per non produrre accavallamenti. Uno schema di massima potrebbe essere il seguente:

```
30 21 * * 1-6 /home/SCRIPTS/pg_dump.sh >/dev/null 2>&1
20 22 * * 1-6 /home/SCRIPTS/backup.sh >/dev/null 2>&1
55 23 * * 1-6 /home/SCRIPTS/tar.sh /dev/null 2>&1
```

L'esempio è puramente indicativo, in quanto gli orari vanno stabiliti anche tenendo conto di fattori quali le dimensioni della base dati, la presenza di altri batch ecc.

Nel caso non si volesse eseguire il backup su nastro, la relativa istruzione va commentata (carattere #).

4. CONCLUSIONE ATTIVITÀ

La predisposizione dell'ambiente operativo si conclude con la messa in linea delle shell di backup e l'attivazione degli script di startup automatico dei prodotti.

4.1. Le shell di backup

Le operazioni da effettuare sono nell'ordine:

- copiare dalla directory *script_ambiente* del kit di installazione le shell *pg_dump.sh*, *backup.sh*, *tar.sh* nella directory **/home/SCRIPTS**;
- assicurarsi che i proprietari delle shell siano *root:root* e i permessi settati su 755;
- configurare adeguatamente la crontab di sistema (cioè di root).

4.2. La shell per l'aggiornamento del software applicativo

Questa shell provvede ad effettuare in maniera automatica l'aggiornamento dei moduli *ear* nella apposita directory di jboss. La shell, inserita in crontab, verifica se ci sono moduli aggiornati nella directory */Trasf/dep*: in caso affermativo copia i moduli nella directory di destinazione e si incarica di chiudere e riavviare jboss, altrimenti termina senza fare nulla.

- copiare dalla directory *script_ambiente* del kit di installazione gli script *upd_sw.sh* e *upddb_sw.sh* nella directory **/export/Trasf**;
- assicurarsi che i proprietari delle shell siano rispettivamente *root:jboss* e *jboss:jboss* e i permessi di entrambe settati su 755;
- nella crontab di root aggiungere la seguente riga (togliendo il carattere di commento quando lo si ritiene opportuno):
#00 7 * * 2-6 /export/Trasf/upd_sw.sh /dev/null 2>&1

4.3. Gli script di startup

- copiare dalla directory *script_ambiente* del kit di installazione gli script di startup *jboss* e *postgresql* nella directory **/etc/init.d**;
- assicurarsi che i proprietari delle shell siano *root:root* e i permessi settati su 755;
- nella directory **/etc/rc5.d** creare i seguenti link

```
ln -s ../init.d/postgresql S64postgresql  
ln -s ../init.d/jboss S94jboss
```

N.B.: I numeri riportati sono puramente indicativi anche se in generale possono considerarsi attendibili; è comunque preferibile che, prima di creare il link, ci si assicuri che nella directory in questione i numeri scelti non siano già stati assegnati ad altri script. Si consideri infine che i due prodotti devono partire preferibilmente verso la fine del processo di boot.

- nella directory **/etc/rc3.d** creare gli stessi link sopra riportati.
- infine eseguire il comando **chkconfig --add postgresql**

4.4. Ambiente grafico

Nel file /etc/rc.local, come utente root, aggiungere in fondo le seguenti righe:

```
#-----#  
#Permette alle applicazioni java di generare immagini  
#-----#  
/usr/bin/Xvfb :1 -screen 0 800x600x8 &
```

4.5. Pulizia dei log di JBOSS

Nella crontab di root aggiungere in testa la seguente istruzione:

```
0 1 * * * find /usr/local/jboss-4.2.3.GA/server/default/log -name "*log*" -mtime +4 -exec rm -f {} \;
```

In tal modo vengono cancellati i file di log di jboss piu vecchi di due giorni.

4.6. Autenticazione alla console di JBOSS

Questo paragrafo riguarda l'attivazione della procedura di autenticazione per l'accesso alla console di Jboss.

Occorre modificare alcuni file di configurazione, per i quali si consiglia di farsi una copia prima di effettuare le modifiche.

- Nella directory /usr/local/jboss-4.2.3.GA/server/default/deploy/jmx-console.war/WEB-INF modificare il file **web.xml** nella sezione *<A security constraint that restricts access to...>* come di seguito riportato:

```
<!-- A security constraint that restricts access to the HTML JMX console  
to users with the role JBossAdmin. Edit the roles to what you want and  
uncomment the WEB-INF/jboss-web.xml/security-domain element to enable  
secured access to the HTML JMX console. -->  
<security-constraint>  
  <web-resource-collection>  
    <web-resource-name>HtmlAdaptor</web-resource-name>  
    <description>An example security config that only allows users with the  
    role JBossAdmin to access the HTML JMX console web application  
  </description>  
    <url-pattern>/*</url-pattern>  
    <http-method>GET</http-method>  
    <http-method>POST</http-method>  
  </web-resource-collection>  
  <auth-constraint>  
    <role-name>JBossAdmin</role-name>  
  </auth-constraint>  
</security-constraint>  
  
<login-config>  
  <auth-method>BASIC</auth-method>  
  <realm-name>JBoss JMX Console</realm-name>  
</login-config>
```

Quindi **trovare e cancellare** nel file suddetto le due seguenti righe consecutive:

```
<http-method>GET</http-method>
```

```
<http-method>POST</http-method>
```

Sempre nella stessa directory, modificare il file **jboss-web.xml** come di seguito riportato:

```
<jboss-web>
<!--  Uncomment the security-domain to enable security.  You will
need to edit the htmladaptor login configuration to setup the
login modules used to authentication users.  -->
<security-domain>java:/jaas/jmx-console</security-domain>

</jboss-web>
```

In pratica le modifiche consistono nello spostare i caratteri di fine-commento (-->) dalla loro posizione originaria nel file a quella mostrata nei due esempi sopra riportati.

- Le stesse modifiche sopra descritte vanno riportate anche negli stessi file **web.xml** e **jboss-web.xml** che si trovano sotto la directory:

```
/usr/local/jboss-4.2.3.GA/server/default/deploy/management/console-mgr.sar/web-
console.war/WEB-INF
```

- Per quanto riguarda l'utente e la password di accesso alla console, si consiglia di lasciare l'utente di default (admin) e settare la password editando i seguenti file:

```
/usr/local/jboss-4.2.3.GA/server/default/conf/props/jmx-console-users.properties
/usr/local/jboss-4.2.3.GA/server/default/deploy/management/console-mgr.sar/web-
console.war/WEB-INF/classes/web-console-users.properties
```

ALLEGATO A

Install_sbnweb:

- DB_creaDB
- DB_iniDB
- DB_creaDB
- exportUnimarc.template.<data>-<polo>.zip
- jboss-configuration
- moduliEAR
- script_ambiente

DB_creaDB:

- Create_DB_XXXese.sql
- sbnwebDbXXX_Completo.sql
- sbnwebDbXXXSenzaIndici.sql

DB_iniDB:

- 00_sbnwebDBinsert_ts_stop_list.sql
- 01_sbnwebDBinsert_tbf_attivita_sbnmarc-tbf_attivita.sql
- 02_sbnwebDBinsert_tb_codici.sql
- 03_sbnwebDBinsert_X_Batch_e_Default.sql
- 04_sbnwebDBinsert_tbf_parametro.sql
- 05_Definisci_Polo.sql
- 06_sbnwebDBinsert_AbilitaPolo_UtenzaRoot.sql

DB_completaDB

- 07_aggiorna_codana_biblioteca_XXX.sql
- 08_pm1_ABILITA_BIBLIOTECHE_DEL_POLO.sql
- 09a_elimina_duplicazioni_userid_bibliotecari_XXX.sql
- 09b_pm2_ABILITA_BIBLIOTECARI DELLE BIBLIOTECHE.sql
- 10_Crea_Localizzazioni.sql
- 11_aggiorna_utenti.sql
- 12_INI_sequence.sql
- 13_ConfrontaCodiciDiValidazione.sql
- 14_catene_rinnovi.sql
- 15_lista_bibliotecari_abilitati.sql
- 16_Aggiusta_default.sql
- 17_Verifica_INI_sequence.sql

exportUnimarc.template.<data>-<polo>.zip:

- exportUnimarc/db_export
- exportUnimarc/dp

exportUnimarc/db_export:**exportUnimarc/dp:**

- export

input
it
makeIndici.sh
makeUnimarc.sh
offlineExportUnimarc
offlineExportUnimarc.linux.cfg
unimarc

exportUnimarc/dp/export:

download<codpolo>.txt
downloadFileAccessori<codpolo>.txt
exportFileAccessori.sh
export.sh
getBidList.txt
getBidList.sh
postgresql-8.2-509.jdbc2.jar

exportUnimarc/dp/input:**exportUnimarc/dp/it:**

finsiel

exportUnimarc/dp/it/finsiel:

misc
offlineExport

exportUnimarc/dp/it/finsiel/misc:

DateUtil.class
Misc.class
MiscString.class
MiscStringTokenizer.class

exportUnimarc/dp/it/finsiel/offlineExport:

CreateOffsetFile.class
CreateRelationFile.class
DbDownload.class
RearrangeFields.class
Select950.class

exportUnimarc/dp/unimarc:**jboss-configuration:**

bin
sbnweb.conf
server

jboss-configuration/bin:

run.conf

jboss-configuration/server:

default

jboss-configuration/server/default:

conf
deploy

jboss-configuration/server/default/conf:

jboss-service.xml

jboss-configuration/server/default/deploy:

jboss-web.deployer
jms
SbnMarcPostgres-ds.xml
SbnWebPostgres-ds.xml

jboss-configuration/server/default/deploy/jboss-web.deployer:

server.xml

jboss-configuration/server/default/deploy/jms:

postgres-jms-sbnweb-service.xml
sbnWeb-queue-service.xml

moduliEAR:

sbn.ear
sbnMarc.ear

script_ambiente:

backup.sh
jboss
pg_dump.sh
postgresql
tar.sh
upd_sw.sh
upddb_sw.sh

ALLEGATO B

Formato per la lettura automatica delle chiavi inventario (campo RFID)

Attualmente l'applicativo SbnWeb gestisce i seguenti formati per la valorizzazione del campo RFID:

Lunghezza del campo	Scomposizione del campo
< 4 caratteri	ERRORE
da 4 a 12 caratteri	<ul style="list-style-type: none">• codice serie (3 chr)• Inventario (max 9 chr numerici)
compreso tra 5 e 13	<ul style="list-style-type: none">• codice serie (3 chr)• separatore (1 chr a spazio)• Inventario (max 9 chr numerici)
14 caratteri	<ul style="list-style-type: none">• codice biblioteca (2 chr)• codice serie (3 chr)• Inventario (9 chr numerici con gli zeri non significativi a sx (0 padded a sx))
15 caratteri	<ul style="list-style-type: none">• codice biblioteca (3 chr) (spazio in testa)• codice serie (3 chr)• Inventario (9 chr numerici con gli zeri non significativi a sx (0 padded a sx))
18 caratteri	<ul style="list-style-type: none">• codice polo(3 chr)• codice biblioteca (3 chr (1 spazio + 2 chr significativi))• codice serie(3 chr)• Inventario (9 chr numerici con gli zeri non significativi a sx (0 padded a sx))
19 caratteri	<ul style="list-style-type: none">• codice serie (3 chr)• codice serie (3 chr)• Inventario (10 chr numerici con gli zeri non significativi a sx (0 padded a sx))• codice polo(3 chr)• codice biblioteca (3 chr (1 spazio + 2 chr significativi))