



eXtraWay Platform Server - eXtraWay Tree Library

Versioni eXtraWay Tree library



####Not yet implemented/released####

Con l'introduzione del **server** di classe **24** si è fatto un più ampio uso delle **librerie dinamiche**, specialmente in forma di **PlugIn**. Per accedere ad essi si fa uso di un **comando XML** generico col quale indicare **parametri** di varia natura ottenendo in risposta un corrispondente **XML**.

La libtree è una di questi **PlugIn** ed è finalizzata a navigare una **gerarchia** di **nodi** che si prestano per diversi scopi.

Di seguito l'elenco delle **versioni** e delle loro **caratteristiche**, [in coda](#) un breve tutorial.

N.B Per quanto concerne la **numerazione** dei **moduli** si rimanda alla [descrizione della numerazione adottata per il Server](#)

Versioni '1'

Versione 1.0.0

- Prima versione. Introduce l'uso dei **nodi standard** e dei **nodi template**;
- Adeguato il **comportamento** della **libreria** allo stesso risultato per **alberi** di **nodi template** e **standard**;
- Introdotta la possibilità di filtrare ulteriormente gli **alberi** di tipo **template** impostando una precedente selezione come **restrizione** da applicarsi in analisi spettrale;
- Introdotta il concetto di **frequenza**: Mentre negli **alberi template** di tipo voc essa è una conseguenza naturale, in quanto derivante dal **vocabolario**, negli **alberi standard** si applica solo là dove sia presente un'apposita action di tipo search. Indicando nella richiesta del **caricamento** di un nodo la specifica che si desidera la **frequenza**, essa verrà tornata nell'**attributo** supplementare frequency ;
- Corretto trattamento dei **caratteri accentati** o **non validi** e dei **vocabolari double** nei **nodi** di **alberi template** introducendo la **codifica Base 64** dove necessaria ed un più puntale trattamento dei vocabolari double.

Come si configurano ed utilizzano i Nodi

Tutto il **sistema** degli **alberi** di **navigazione** si basa sul concetto di **nodo**.

Ogni **nodo** può rappresentare un **punto intermedio**, appunto nodo, ovvero un'estremità del nostro albero, quindi una **foglia**.

Questi **nodi** altro non sono che **unità informative** che devono convivere nell'ambito dello stesso **archivio** con le unità informative che realmente lo caratterizzano.

Questo comporta che il **file** di **configurazione** dell'**archivio**, per poter operare correttamente con gli **alberi** di **nodi**, deve prevedere la configurazione di questa particolare **unità informativa**, come nell'**esempio** seguente:

```
<primary_node ud_name="xwNodeRecord" ud_container="xwNodeRecordList">
  <unique_rule search_rule="[XML,/xwNodeRecord/@recordId]"/>
  <key name="XML,/xwNodeRecord/" key_style="skip" path_style="container"/>
  <key name="XML,/xwNodeRecord/@recordId" key_style="single"/>
  <key name="XML,/xwNodeRecord/node/@nodeId" key_style="single"/>
  <key name="XML,/xwNodeRecord/node/@type" key_style="skip"/>
  <key name="XML,/xwNodeRecord/node/@label" key_style="skip"/>
  <file_location mode="rule" rule="$@XML,/xwNodeRecord/@recordId|tl:@$;trees"
move_always="yes"/>
</primary_node>
```

Esistono due **tipologie** di **nodi**, quelli **statici** e quelli **dinamici**, o **template**.

Se pure nulla escluda che si possano combinare in vario modo, di fatto essi sono sino ad ora stati realizzati solo distintamente, vale a dire **alberi** di soli **nodi statici** ed **alberi** di soli **nodi dinamici**.

Introduciamo innanzitutto il concetto di **nodo**. Esso è rappresentato da un'**unità informativa** di tipo xwNodeRecord caratterizzata da un recordId **univoco** e da una serie di **nodi** al suo interno. Ciascuno di essi indicherà se conduce ad altri nodi o se si tratta di un **elemento terminale**, una **foglia**.

Il recordId di un **albero** si compone sempre di una **label** che lo identifica e da un **nodo principale** riconosciuto come root. Da quel momento l'**identificazione** di tutti i successivi **nodi** dello stesso **albero** deve avere in comune solo la **radice** del recordId. Ogni recordId è suddiviso quindi in due **parti**, **radice** ed **estensione**, separate dal **carattere** :

Prendiamo ad **esempio** un **albero** di **classificazione**:

```
<xwNodeRecord recordId="MainClassif:root">
  <node nodeId="MainClassif:0" type="list" label="0 - TEMI GENERALI E CIRCOLAZIONE STRADALE">
    <action type="search">[labelFull]="classif.0.*"</action>
```



```
</node>
<node nodeId="MainClassif:1" type="list" label="1 - VEICOLI">
  <action type="search">[labelFull]="classif.1.*"</action>
</node>
<node nodeId="MainClassif:2" type="list" label="2 - ENTIT&#xC0; TECNICHE E COMPONENTI DI
VEICOLI">
  <action type="search">[labelFull]="classif.2.*"</action>
</node>
<node nodeId="MainClassif:3" type="list" label="3 - VISITE E PROVE VEICOLI (ALLESTIMENTI,
TRASFORMAZIONI, REVISIONI)">
  <action type="search">[labelFull]="classif.3.*"</action>
</node>
<node nodeId="MainClassif:4" type="list" label="4 - CONDUCENTI">
  <action type="search">[labelFull]="classif.4.*"</action>
</node>
<node nodeId="MainClassif:5" type="list" label="5 - AUTOTRASPORTO">
  <action type="search">[labelFull]="classif.5.*"</action>
</node>
<node nodeId="MainClassif:6" type="list" label="6 - STRADE E SEGNALETICA">
  <action type="search">[labelFull]="classif.6.*"</action>
</node>
<node nodeId="MainClassif:7" type="list" label="7 - TRASPORTI CON IMPIANTI FISSI">
  <action type="search">[labelFull]="classif.7.*"</action>
</node>
<node nodeId="MainClassif:8" type="list" label="8 - NAVIGAZIONE INTERNA E DA DIPORTO">
  <action type="search">[labelFull]="classif.8.*"</action>
</node>
</xwNodeRecord>
</xwNodeRecordList>
```

Come si può notare il recordId di questo **nodo** è identificato da una **label principale**, MainClassif e dal **suffixo** : root ad identificare che si tratta del **nodo principale** dell'**albero**.

Essi contiene uno o più **elementi** node. Vediamone uno nel **dettaglio**:

```
<node nodeId="MainClassif:0" type="list" label="0 - TEMI GENERALI E CIRCOLAZIONE STRADALE">
  <action type="search">[labelFull]="classif.0.*"</action>
</node>
```

L'**elemento** node deve sempre avere un **attributo** type che indichi la sua **natura**. Il **valore** di questo **attributo** può essere:

list	Indica che il nodo così identificato punta ad un ulteriore xwNodeRecord il cui recordId corrisponde al nodeId
node	Indica che si tratta di una foglia, non destinata a condurre altrove
template	Indica che il nodo in esame è in forma di template e che quindi segue un comportamento del tutto differente che sarà spiegato in seguito

Il node così identificato ha normalmente un nodeId¹⁾. Il **valore** di tale nodeId segue i **criteri** del recordId già visto ed impone, di fatto, solo la presenza della stessa **radice** del recordId cui si appartiene.

Analogamente può avere, e normalmente ha, un **attributo** label finalizzato alla sua **rappresentazione esteriore** in sede applicativa.

Al node può essere associata un'**azione**. Attualmente la sola **azione** prevista (indicata dall'attributo type) è l'azione di **ricerca**, codificata con il **valore** search. L'**elemento** action a ciò preposto conterrà quindi l'**espressione** di **ricerca** che si intende utilizzare.

In **condizioni normali** tutti i **nodi foglia** hanno un'azione associata. Essa si associa sovente anche a **nodi intermedi** ma, mentre il primo caso rientra nella normalità, la presenza di un'**azione** in un nodo intermedio dipende totalmente dalle **esigenze** applicative e dalla natura dell'**albero** di **nodi** che si sta costituendo.

Un albero di nodi statici

Secondo l'**esempio** precedente, vediamo il **nodo puntato** dal primo dei nodi citati nella **radice** dell'**albero**:

```
<xwNodeRecord recordId="MainClassif:0">
  <node nodeId="MainClassif:0.0" type="list" label="0.0 ORDINAMENTO DELLO STATO - PUBBLICA
AMMINISTRAZIONE">
    <action type="search">[labelFull]="classif.0.0.*"</action>
  </node>
  <node nodeId="MainClassif:0.1" type="list" label="0.1 MINISTERO DEI TRASPORTI">
```



```

    <action type="search">[labelFull]="classif.0.1.*"</action>
  </node>
  <node nodeId="MainClassif:0.2" type="list" label="0.2 TRIBUTI E TARIFFE DEL SETTORE">
    <action type="search">[labelFull]="classif.0.2.*"</action>
  </node>
  <node nodeId="MainClassif:0.3" type="list" label="0.3 ENTI E ORGANIZZAZIONI DEL SETTORE">
    <action type="search">[labelFull]="classif.0.3.*"</action>
  </node>
  <node nodeId="MainClassif:0.4" type="list" label="0.4 REGOLAMENTAZIONE DELLA CIRCOLAZIONE">
    <action type="search">[labelFull]="classif.0.4.*"</action>
  </node>
  <node nodeId="MainClassif:0.5" type="list" label="0.5 CIRCOLAZIONE STRADALE - NORME DI
COMPORTEMENTO">
    <action type="search">[labelFull]="classif.0.5.*"</action>
  </node>
  <node nodeId="MainClassif:0.6" type="list" label="0.6 CIRCOLAZIONE DI PARTICOLARI VEICOLI O
UTENTI">
    <action type="search">[labelFull]="classif.0.6.*"</action>
  </node>
  <node nodeId="MainClassif:0.7" type="list" label="0.7 SICUREZZA STRADALE, INFORTUNISTICA E
ASSICURAZIONI">
    <action type="search">[labelFull]="classif.0.7.*"</action>
  </node>
  <node nodeId="MainClassif:0.8" type="list" label="0.8 REGIME SANZIONATORIO DELLA CIRCOLAZIONE
STRADALE">
    <action type="search">[labelFull]="classif.0.8.*"</action>
  </node>
  <node nodeId="MainClassif:0.9" type="list" label="0.9 PROCEDURE AMMINISTRATIVE, MODULISTICA E
DOCUMENTI">
    <action type="search">[labelFull]="classif.0.9.*"</action>
  </node>
</xwNodeRecord>

```

ed infine il **nodo** di **livello** ulteriormente **inferiore**, sempre riferito al primo dei **nodi** qui citati:

```

<xwNodeRecord recordId="MainClassif:0.0">
  <node nodeId="MainClassif:0.0.1" type="list" label="0.0.1 organizzazione dell'Italia e
dell'Unione Europea">
    <action type="search">[labelFull]="classif.0.0.1.*"</action>
  </node>
  <node nodeId="MainClassif:0.0.2" type="node" label="0.0.2 trasporti in generale, anche
nell'Unione europea">
    <action type="search">[labelFull]="classif.0.0.2.*"</action>
  </node>
  <node nodeId="MainClassif:0.0.3" type="node" label="0.0.3 lavori pubblici - agricoltura">
    <action type="search">[labelFull]="classif.0.0.3.*"</action>
  </node>
  <node nodeId="MainClassif:0.0.4" type="list" label="0.0.4 ambiente - salute - protezione
civile">
    <action type="search">[labelFull]="classif.0.0.4.*"</action>
  </node>
  <node nodeId="MainClassif:0.0.5" type="node" label="0.0.5 istruzione pubblica - formazione
professionale - CNR">
    <action type="search">[labelFull]="classif.0.0.5.*"</action>
  </node>
  <node nodeId="MainClassif:0.0.6" type="list" label="0.0.6 interni e polizia">
    <action type="search">[labelFull]="classif.0.0.6.*"</action>
  </node>
  <node nodeId="MainClassif:0.0.7" type="node" label="0.0.7 diritto internazionale nei trasporti
e nella circolazione stradale">
    <action type="search">[labelFull]="classif.0.0.7.*"</action>
  </node>
  <node nodeId="MainClassif:0.0.8" type="node" label="0.0.8 economia - industria - commercio -
lavoro">
    <action type="search">[labelFull]="classif.0.0.8.*"</action>
  </node>
</xwNodeRecord>

```



```
</node>
<node nodeId="MainClassif:0.0.9" type="node" label="0.0.9 enti locali">
  <action type="search">[labelFull]="classif.0.0.9.*"</action>
</node>
</xwNodeRecord>
```

In questo ultimo caso risulterà evidente come alcuni dei **nodi citati** siano list ed altri siano invece node in quanto non prevedono ulteriori **livelli d'alberatura**.

Un albero di nodi template

Visto l'**esempio** di cui al caso precedente, vediamo ora come si componga e comporti un **albero** di **nodi** di tipo **template**.

Se è evidente che un **albero** composto di numerosi rami avrà un diverso **record** xwNodeRecord per ciascun **ramo** da percorrere, nel caso dell'albero di tipo **template** si avrà uno ed un solo **nodo auto-consistente**.

Vediamo come si dichiara un **simile nodo**:

```
<xwNodeRecord recordId="ChronoTree:root" template="voc">
  <node voc="/articolo/specifiche_rivista/@anno/"> <!-- Primo livello dell'albero -->
    <node voc="/articolo/specifiche_rivista/@numero/"> <!-- Secondo livello dell'albero -->
      <node voc="/articolo/rubriche/rubrica/"></node> <!-- Terzo livello dell'albero -->
    </node>
  </node>
</xwNodeRecord>
```

Innanzitutto l'**albero** di tipo **template** ha un **attributo omonimo** che dichiara la modalità secondo la quale il suo **contenuto** dev'essere interpretato. Nella fattispecie esiste attualmente solo la **modalità** voc che comporta un accesso ai **vocabolari** potenziato dall'analisi spettrale.

Ogni ulteriore **nodo** dichiara, con un **attributo** voc, quale **vocabolario** dell'**archivio** debba essere interessato.

N.B: Il tipo di **nodo** ritornato dalle **funzioni** della libtree quando si accede ad un nodo di questa fatta è del tutto analogo a quelli appena visti negli **alberi statici**, con la differenza che il ciascun **nodo** viene prodotto automaticamente e conduce al **ramo** sottostante che rappresenta, appunto, l'**analisi spettrale** desiderata.

Il nostro **esempio** fa accesso al **vocabolario** dell'anno e mostrerà quindi un **nodo principale** con tanti **nodi** di tipo list quante sono le diverse **chiavi** del **vocabolario** dell'anno, nel nostro caso, quanti articoli saranno stati pubblicati in un determinato anno con relativa **cardinalità**.

Per ciascuno di questi **nodi**, percorrendo il legame al **nodo** successivo tramite il nodeId, avremo un'**analisi spettrale** sul **vocabolario** del numero della rivista.

Il **nodo** prodotto avrà tanti nodi di tipo list quanti saranno i **numeri emessi** nell'**anno** prescelto. Per ciascuno di essi verrà indicato il **numero** di **articoli pubblicati** quell'anno sulla **rivista** avente quel numero.

Compiendo un ulteriore passo avanti, ovvero richiedendo l'**analisi** della **rubrica**, avremo un nuovo **record** che conterrà **nodi** questa volta di tipo node, in quanto **nodi foglia**. Per ciascuno di essi verrà riportata la **cardinalità** di quanti **articoli** (pubblicati quell'anno nella rivista con quel numero) appartengano ad una data **rubrica**.

Quale che sia il **livello** del **nodo**, a ciascuno di essi viene associato d'ufficio un'**azione** atta a selezionare tutti i **record** che il **nodo** stesso descrive ed enumera.

Composizione del comando

I **comandi** per questo **Plug-In** si rifanno alle [regole standard per questo tipo di comandi XML](#)

Unifichiamo quindi nel concetto di **parametro** un **valore** che può essere espresso sia come **attributo** che come **elemento**.

Vediamo comunque che **comandi** sono **disponibili** e quali sono i loro **modificatori**.

Comando	Dettagli
XTree.treeList	Richiede che vengano elencati gli alberi disponibili, indipendentemente dal fatto che essi siano statici o dinamici ²⁾ . Prevede un parametro arc che indichi il nome logico dell'archivio per il quale si richiede tale elencazione. <?xml version="1.0"?> <cmd c="0" func="XTree.treeList" arc="rubricaIBC"/>



Comando	Dettagli
XTree.loadNodeRecord	Richiede che venga caricato un particolare nodo dell'albero indicato. L'indicazione è insita nel parametro <code>id</code> e comporta il caricamento del nodo avente corrispondente <code>recordId</code>. Nel momento in cui si accede al primo nodo di un albero, del quale si conosce la denominazione tramite il precedente comando <code>XTree.treeList</code>, si deve accodare all'etichetta che identifica l'albero il suffisso <code>:root</code>. In ogni altro caso, vale a dire caricando nodi successivi dell'albero dal loro nodo padre, il <code>nodeId</code> ricevuto sarà sufficiente per il caricamento. Richiede anche in questo caso l'indicazione del parametro <code>arc</code>. <pre><?xml version="1.0"?> <cmd c="0" func="XTree.loadNodeRecord" arc="rubricaIBC" id="ChronoTree:root"/></pre> Esplicitando anche il parametro <code>frequency</code> con valore positivo³⁾ in presenza di nodi aventi un'appropriata action associata si avrà in risposta anche un attributo <code>frequency</code> opportunamente valorizzato, in grado di farci scegliere se mostrare o meno il nodo in esame. <pre><?xml version="1.0"?> <cmd c="0" func="XTree.loadNodeRecord" arc="rubricaIBC" id="ChronoTree:root"> <frequency>true</frequency> </cmd></pre>
XTree.executeNode	Richiede gli stessi parametri di cui al comando precedente, quindi l'indicazione dell'archivio nel parametro <code>arc</code> e l'indicazione del nodo di cui compiere l'azione nel parametro <code>id</code>. Riferendosi alla action associata ad un nodo, l' <code>id</code> che si deve indicare si riferisce al <code>nodeId</code> dell'elemento <code>node</code> e non al <code>recordId</code> del record. <pre><?xml version="1.0"?> <cmd c="0" func="XTree.executeNode" arc="rubricaIBC" id="ChronoTree:2011:1"/></pre>

¹⁾
per quanto esso sia realmente necessario solo se il nodo è di tipo `list`

²⁾
template

³⁾
Si intendono valori positivi i valori quali `yes`, `on`, `true`, `si` e `1`.