

Configurazione degli archivi

La configurazione degli archivi consiste di due grandi aree.

- Configurazione delle Unità Informative, dei vocabolari e dei canali di ricerca
- Configurazione dei profili generali dell'archivio

In ambo i casi va tenuto in considerazione che quando una componente della configurazione può accettare valori *booleani* i valori ammessi sono:

Tipo Booleano	Elenco valori ammessi ¹⁾
Affermativo (true)	On, Y, Yes, True, Si, Ok, 1
Negativo (false)	Off, N, No, False, Nok, 0

Configurazione Unità Informative e Chiavi

Perché un archivio esista è fundamentalmente necessario che venga istruito per riconoscere quelli che saranno i suoi record, altrimenti detti *Unità Informative*. Questa terminologia spiega, ancor meglio del termine inglese record che pure è riconosciuto a livello internazionale, l'intenzione che il Server eXtraWay si prefigge, ovvero di trattare in maniera unitaria degli insiemi di dati omogenei elevandone il ruolo a quello di informazione.

Il progetto eXtraWay prende le mosse da un'intenzione ambiziosa, quella di saper considerare come *Unità Informativa* qualsiasi insieme di elementi ed attributi riconoscibili, appunto, per mezzo di un elemento che li racchiude e che viene di seguito identificato come *Primary Node Container* o, per brevità, *pne*.

In questo modo si sarebbe potuto creare un archivio sulla base di Xml esistenti con poche attività di configurazione.

Parimenti l'intenzione originaria era anche di concepire e gestire archivi in cui diverse tipologie di *Unità Informative* potessero essere nidificate le une nelle altre, come rientra nella natura del formato Xml prescelto per la conservazione delle informazioni. Questo comporta il fatto che un'*Unità Informativa* può essere considerata Padre di una o più altre *Unità Informative* e così via, rendendo di fatto più complessa ed articolata l'attività di manutenzione delle stesse.

La realtà dei fatti, però, dimostra che la stragrande maggioranza dei casi rientra in una casistica più circoscritta e precisa, quella in cui un archivio nasce per sopperire ad una specifica esigenza e viene quindi modellato disegnando specifiche *Unità Informative* senza che esse siano nidificate tra loro bensì creando legami logici, relazionali o applicativi tra di esse, legami che si svincolino, di fatto, da una struttura Xml nidificata.

Ogni archivio avrà quindi uno o più diversi tipi di *Unità Informativa* e per ciascuna di esse dovrà definire i dettagli per mezzo dei quali essa verrà riconosciuta, gestita sul *file system* e condurrà all'alimentazione di vocabolari per mezzo dei quali compiere le attività di ricerca.

Nell'esemplificazione che segue si terrà conto di questo aspetto, ovvero delle caratteristiche degli archivi standard. Le varianti, per così identificarle, riferite agli archivi realizzati con *Unità Informative* non disegnate appositamente²⁾ ovvero con *Unità Informative* nidificate verranno trattate di volta in volta come eccezioni.

Le Unità Informative

L'identificazione di un unità informativa si attua introducendo nel file di configurazione un elemento `primary_node` per ciascuna unità informativa desiderata. Esso deve avere i seguenti attributi³⁾.

Attributo	Significato
ud_name	Nome dell'elemento che rappresenta l'unità informativa. Tale elemento, i suoi attributi e tutto il suo contenuto rappresentano i dati di questo tipo di unità informativa
ud_container	Nome dell'elemento che funge da <code>root element</code> nei files Xml che ospiteranno le unità informative. Com'è noto un file Xml può e deve avere un solo elemento che funga da radice. Salvo casi rari ⁴⁾ in ogni file Xml sono presenti una o più unità informative e diviene quindi necessario avere un elemento che le racchiuda tutte. Questo è il nome di tale elemento
ud_validator	Se presente è il nome del file DTD ⁵⁾ per mezzo del quale compiere la validazione dei documenti di questa unità informativa
ud_base	URI che identifica l' <code>xml:base</code> per quest'unità informativa
ud_xslt	Nome del file Xslt da applicare ai record per estrarre da essi una sorta di semilavorato dal quale realizzare indici, titoli, ordinamenti e così via
metadata	Indica se le unità informative debbano essere corredate o meno con i metadati. Default "yes"
globalkey	Indica se si desidera generare una chiave globale per quest'unità informativa. Il tema verrà più accuratamente discusso in un separato paragrafo ma per mezzo di questa voce si può indicare se si vuole che tutti i canali di ricerca dichiarati affluiscano nella chiave globale (valore "yes"), se si vuole che vi affluiscano anche i canali sottintesi (valore "full") oppure che non venga realizzata la chiave globale (valore "no" che rappresenta il Default)
empty_key	Introdotta nella sola versione Enterprise per dichiarare l'intenzione di calcolare la chiave nulla per tutte le chiavi esplicitamente dichiarate. La chiave nulla non può essere calcolata per le chiavi implicite in quanto solo quando vengono rilevate la prima volta si ha percezione della necessità di calcolare tale chiave anche per tutti i record pregressi.



L'aspetto che un archivio potrebbe avere sarà il seguente:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE xway_cfg SYSTEM "http://www.3di.it/extraway/xway_20020215.dtd">
<xway_cfg>
  <primary_node ud_name="doc" ud_container="list_of_doc" globalkey="full"
ud_xslt="proprietario.xsl">
    ...
  </primary_node>
</xway_cfg>
```

L'elemento *primary_node* è destinato a contenere una serie di impostazioni specifiche dell'*Unità Informativa* indicata ed in particolare delle chiavi che in essa devono essere riconosciute. Di queste ultime parleremo in seguito, per ora soffermiamoci sugli elementi *unique_rule* e *file_location*.

L'elemento *unique_rule* indica la ricerca, in linguaggio eXtraWay, il cui esito indicherà se il documento in esame viola o meno un'univocità, ovvero contenga valori che possono essere presenti in uno ed un documento soltanto e tale documento sia già esistente. Ovviamente questa verifica viene compiuta dal server ad ogni salvataggio che viene compiuto e ne provoca il fallimento se l'univocità non viene rispettata.

L'elemento *unique_rule* contiene un singolo attributo, denominato *search_rule* che contiene, come detto, la sintassi di ricerca richiesta. Essa può prevedere operatori booleani (quali AND e OR) per poter esprimere una condizione complessa ed eventualmente correlata a più clausole alternative o complementari. Nella sintassi non si indicherà altro che il canale sul quale compiere la ricerca in quanto ad esso verrà automaticamente accoppiato il valore estratto dal documento con operatore di uguaglianza.

Il seguente può essere un esempio rappresentativo

```
<unique_rule search_rule="[XML,/doc/@nrecord] or [XML,/doc/@num_prot] or
[XML,/doc/extra/raccomandata/@numero]" />
```

Analogo ragionamento va fatto per la regola di posizionamento delle *Unità Informative* nei files XML. Lo scopo, ovviamente, è quello di dare alle *Unità Informative* una collocazione fisica che ricalchi l'organizzazione logica attribuita all'archivio oppure ottenere una distribuzione che non comporti files troppo piccoli o troppo grandi.

La regola di distribuzione su *file system* si basa su regole fisse o su una sintassi piuttosto complessa⁶⁾ che utilizza i contenuti del documento stesso.

Gli attributi dell'elemento *file_location* sono:

Attributo	Valore
mode	Indica in che modo avviene la collocazione su disco. Può assumere i valori: single , day , month , year e rule . Come risulta intuitivo i primi metodi indicano di costituire un file per ogni documento ovvero per intervalli di tempo ⁷⁾ per giorno, mese o anno. In fine si prevede la modalità <i>rule</i> che introduce la modalità complessa, regolata da una batteria di regole di collocazione
dir	Obsoleto, inutilizzato
rule	Quest'attributo viene preso in esame esclusivamente se l'attributo <i>mode</i> vale <i>rule</i> . Esso contiene una o più regole di collocazione su disco. Ciascuna acquisisce dal documento una o più "parti" e compone un percorso di file. Se una regola comprende parti che non sono presenti/disponibili nel documento in esame, si passa alla successiva sino a raggiungere l'ultima regola indicata. Se neppure questa porta alla determinazione completa di un percorso, il salvataggio darà errore
move_always	Negli scenari in cui al variare dei contenuti di un documento varia anche il percorso del file che dovrebbe ospitarlo, il server non provvede a spostarlo nel file di destinazione calcolato se in quello in cui si trova il documento c'è spazio sufficiente per contenerlo dopo la modifica. In pratica, quindi, un documento viene spostato dal file in cui si trova verso il file frutto dell'applicazione della regola solo se la sua crescita ne impedisce il salvataggio sul posto ⁸⁾ . Impostando a <i>true</i> quest'attributo, invece, l' <i>Unità Informativa</i> viene spostata in file differente non appena il suo contenuto, applicato alle regole, produce un percorso di file diverso. In questo modo si garantisce sempre un'organizzazione fisica sul <i>file system</i> congrua con l'organizzazione logica dell'archivio.

Queste impostazioni possono essere applicate tanto alla singola *primary_node* quanto all'intero archivio. Ciò comporta che, se per una data *primary_node* non viene indicata la regola di univocità o la regola di collocazione sul disco, il server tenta di applicare la regola generale. In tal caso gli elementi descritti verranno cercati direttamente come figli dell'elemento radice *xway_cfg* anziché internamente alla *primary_node*.

La Configurazione delle Chiavi

In ciascuna *primary node* verranno indicate le chiavi che si intende riconoscere esplicitamente. Questo concetto può apparire curioso ma di fatto il server eXtraWay, una volta riconosciute tutte le *Unità Informative*, compirà indici su ciascuna delle parti, siano essi elementi o attributi, in essa presenti, seguendo un proprio standard: tutti i contenuti degli elementi verranno indicizzati come vocabolari *multi valore* quindi tokenizzati e ricercabili parola per parola, mentre tutti gli attributi verranno indicizzati come vocabolari *mono valore* quindi assunti per la loro interezza e ricercabili solo per il valore completo. Ovviamente, essi verranno



identificati per mezzo del loro *XPath*.

Per dare maggior ricchezza agli indici prodotti da simili contenuti è stata realizzata una batteria di impostazioni che può essere valorizzata secondo le istruzioni che seguono.

Ogni chiave si esprime per mezzo di un elemento *key* che deve avere per lo meno l'attributo *name* che ne consenta l'identificazione. Esso esprime il percorso XML dell'elemento o dell'attributo per il quale si intende dettare una regola specifica. Il valore indicato viene normalmente prefissato dall'etichetta 'XML,' che serve a distinguere le chiavi native del documento da quelle frutto di estensione *xs/t* alle quali viene dato prefisso 'XSL,'.

Dopo tale prefisso il percorso che viene indicato può essere assoluto o relativo⁹⁾

I "MacroCanali" XML e XSL

Il canale XSL è stato introdotto diverso tempo fa ma, di fatto, sempre utilizzato in modo marginale. Il suo scopo è molteplice.

Parametri per la configurazione

Vediamo ora gli attributi che possono essere impostati ed il loro valore e significato.

Attributo	Significato
name	Come detto questo è l'unico attributo obbligatorio e deve descrivere un percorso XML assoluto o relativo. In caso di percorso relativo, qualora esistano altre chiavi con percorsi assoluti o relativi che potrebbero intercettare chiavi simili, quello con la maggiore estensione dopo l'indicazione di percorso relativo(//) verrà preso in esame
valute_type	Tutti i valori sono considerati alfanumerici, salvo diversa impostazione. I valori che possono essere indicati sono - alfa: Alfanumerico - num: Numerico - date: Data - file: Percorso di file con separatori da normalizzare - alnum: Prevede che si assumano solo i valori numerici ed alfabetici escludendo ogni altro carattere. Il valore di default è "alfa".
key_style	Indica la modalità con la quale si devono fare le <i>tokenizzazioni</i> e quindi calcolare le chiavi sul campo. I valori ammessi sono: - auto: Mono valore sugli attributi e multi valore sugli elementi - one: Mono valore - one-md5: Trattamento del campo come mono valore. La chiave prodotta è l'MD5 del contenuto senza rispetto del Case ¹⁰⁾ - one-md5cs: Trattamento del campo come mono valore. La chiave prodotta è l'MD5 del contenuto con rispetto del Case - single: Mono valore come la chiave one ma senza posizione. Si presta particolarmente per le chiavi univoche e per le chiavi primarie Si ricorda che queste sono le sole chiavi a non avere una posizione e quindi i vocabolari così costituiti non possono e non devono essere combinati (Vds. <i>key_also</i>) con altri che non abbiano pari caratteristiche - multi: Multi valore - double: Doppia indicizzazione. Sul contenuto del campo vengono compiute due diverse operazioni producendo le chiavi multi valore ed una chiave mono valore che affluiscono nello stesso vocabolario ¹¹⁾ - double-md5 ovvero double-md5cs : Analogamente a quanto detto nel caso one-md5 questa modalità influenza la chiave non tokenizzata di questo campo a doppia indicizzazione - skip: Indica che per questo canale non si intendono calcolare chiavi di alcun genere - spatial: Qualifica il campo come "spaziale" ovvero destinato a contenere coordinate atte ad identificare punti, linee e poligoni secondo una sintassi ben precisa. L'esito di tale indicizzazione consente ricerche di un documento secondo la sua collocazione geografica.
primary_key	Indica che la chiave è da considerarsi primaria. In archivi in cui l'indicizzazione avviene in due fasi queste chiavi vengono indicizzate subito mentre tutte le altre in un secondo momento. Si applica solitamente a valori seriali ed univoci
empty_key	Booleano che indica se per questo canale si debba compiere anche l'indicizzazione del valor vuoto ¹²⁾ . La chiave ottenuta, diversamente dalle altre, non è posizionale e quindi non può essere utilizzata nelle adiacenze. Nota: Dalla versione 24.7.* si introduce una correzione richiesta dalla combinazione di quest'impostazione con l'impostazione <i>key_alias</i> . Essa causava un errato comportamento in quanto a popolazione del vocabolario effettivamente desiderato, vale a dire quello della chiave di cui si fa <i>alias</i> . A seguito dell'intervento è comunque necessario indicare come <i>empty_key</i> tanto la chiave che dichiara altra chiave come suo <i>alias</i> quanto la stessa chiave che ospita questi contenuti.
path_style	Quest'attributo qualifica solo percorsi XML di elementi. Esso indica come debba essere trattato il contenuto di tali elementi e degli elementi che a loro volta esso contiene. I valori ammessi sono - auto: Comportamento di default, le regole applicate all'elemento valgono solo per esso e per nessuno degli elementi in esso contenuti - container: Ogni elemento contenuto in quello indicato ¹³⁾ non apparirà nell'elenco delle chiavi disponibili e tutto il suo contenuto verrà fatto convergere in questo stesso vocabolario. - transparent: E' un po' l'opposto del caso precedente. In pratica un elemento <i>transparent</i> fa sì che l'elemento stesso non appaia nel percorso XML di chiave facendo equivalere questa ad altre diversamente configurate



Attributo	Significato
stop_list	Indica quale file di <i>Stop List</i> utilizzare in selezione ¹⁴⁾ . Valore di default: italian.stp, collocata nella directory conf del server
attr_collate	Indica una regola di concatenamento degli attributi di un elemento per produrre automaticamente delle chiavi composite. <i>Obsoleto, sostituito da collate_rule</i>
collate_rule	Definisce con una sintassi complessa ¹⁵⁾ in che modo compiere un concatenamento di attributi di un elemento per produrre una o più chiavi speciali
key_alias	Indica che la chiave in esame non diverrà mai una chiave vera e che il suo contenuto verrà dirottato integralmente nella chiave il cui percorso è indicato in questo attributo
key_also	Si affianca e completa lo scenario descritto da <i>key_alias</i> alimentando il vocabolario del campo in oggetto e, contemporaneamente, anche il vocabolario degli altri canali indicati in questo attributo. Se sono presenti più canali essi sono separati dal simbolo (' '). <i>Si prega di prestare attenzione a non combinare vocabolari non compatibili tra loro.</i> <i>Nella fattispecie si ricorda che le chiavi di tipo “single” sono le sole ad essere prive di posizione quindi una chiave di questo tipo non può essere posta in <i>key_also</i> su un vocabolario che non sia anch'esso “single” e vice versa.</i>
instance	Valore booleano che consente di identificare gli elementi che sono ripetibili o che sono destinati a fungere da contenitori di elementi più di dettagli. Gli elementi indicati come <i>instance</i> possono essere utilizzati in ricerca esprimendo adiacenze di campo che, in questo caso, si intendono come <i>compresenza dei valori richiesti in elementi/attributi tutti facenti parte della stessa ripetizione dell'elemento instance indicato</i> . Quando tale modalità di ricerca ha luogo sugli attributi di un elemento, l'identificazione dell'elemento <i>instance</i> di riferimento avviene automaticamente
expert_system	Utilizzato nell'ambito di speciali applicazioni aventi estensioni di indicizzazione e ricerca di tipo linguistico. <i>Obsoleto, caduto in disuso.</i>
extra_cat	Si applica esclusivamente per archivi ai quali sia stata accoppiata una libreria dinamica appositamente studiata e realizzata. Comporta che il valore di questo campo venga inviato alla libreria dinamica che, al termine dell'indicizzazione di ciascun documento, ha la facoltà di produrre chiavi speciali ¹⁶⁾
serial_id	Indica quale delle regole seriali debba applicarsi al campo indicato. Il tema dei seriali sarà approfondito in seguito.
global	Valore booleano che indica se le chiavi estratte da questo campo debbano far parte o meno del vocabolario globale il cui senso verrà chiarito in seguito
attach	Indica se il contenuto del campo dev'essere trattato come un allegato. Può assumere 3 valori: - 'no' (defulat): ad indicare che il campo non contiene ID di allegati; - 'yes' ad indicare che il campo contiene un ID di allegato; - 'index' ad indicare che il campo contiene un ID di allegato e che lo stesso va sottoposto ad incicizzazione. In caso di valore valido ('yes' 'index') il contenuto del campo viene trattato alla stregua dell'attributo @name di un elemento xw: file, naturale collocazione degli ID di allegato.
sql_single	Nella simulazione SQL delle <i>Unità Informative</i> solo gli attributi dell'elemento radice e gli elementi presenti come suoi figli una sola volta vengono considerati <i>singoli</i> e quindi non portano alla produzione di una combinazione di righe quando il documento viene <i>tradotto</i> in una vista SQL. Per tutti gli altri attributi ed elementi che non rispettino questa collocazione, l'indicazione di <i>sql_single</i> li identifica in egual maniera
presence_key	Se attivato indica al server di creare una chiave specifica atta ad indicare che in questo documento esiste l'elemento/attributo indicato. Si presta principalmente per compiere controlli di consistenza sugli archivi ed identificare, ad esempio, documenti che non siano doverosamente conformati o che contengano dati in un formato diverso da quello richiesto
iwgap	Rappresenta un indice di <i>scalatura</i> . Fino alla versione 25.8.3 non ha mai trovato reale applicazione salvo alcune attività sperimentali. Dalla versione 25.9.0 (Candidate) assume il seguente comportamento: Consente di stabilire “aree” di numerazione atte a separare i numeri d'indice dei termini tra loro. Le aree sono da considerarsi in sequenza quindi il beneficio si percepisce sin quando le stesse vengono sempre espresse nello stesso ordine quando il documento subisce una modifica. Si deve inoltre aver cura di non spezzare con aree separate porzioni di documento nelle quali si intenda compiere selezioni con 'adiacenza di gruppo'. Il valore da assegnare è la potenza di 2 da usare per spostare avanti la soglia di numerazione. Il valore dev'essere compreso tra 5 e 16 ($2^5 - 2^{16}$). Valori inferiori a 5 vengono considerati comunque 5. Nella versione Enterprise il valore espresso assume un significato diverso. Il valore impostato indica semplicemente il <i>punto di partenza</i> per la numerazione delle chiavi calcolate su quella determinata porzione di record. Le posizioni non sono più in sequenza, ciascuna porzione del record può quindi avere una numerazione propria, con netti benefici in termini di indicizzazione specialmente in caso differenziale.
defaultkey	Documentazione in fase di allestimento

Esempi

Documentazione in allestimento

Presence

La chiave che si produce in caso di applicazione di “presence_key” è la seguente.

In un'unità informativa doc avente un attributo @type, un attributo @num ed un elemento text, tutti definiti *presence_key*=“yes”,



avremo un vocabolario automaticamente prodotta avente il seguente XPath...

```
XML, /doc/#presence
```

...il contenuto di tale vocabolario riporterà, come chiavi i seguenti valori:

- /doc/@num
- /doc/@type
- /doc/text

Selezionando una di queste chiavi si identificheranno i record in cui quel percorso XPath esiste, indipendentemente dal fatto che esso sia in qualche modo valorizzato e/o che sia stato realizzato un vocabolario per il suo contenuto.

Al fine di identificare tutti i documenti nei quali non figura l'attributo @num, la corrispondente frase di ricerca sarebbe la seguente:

```
NOT ( [/doc/#presence]="/doc/@num" )
```

Il trattamento della chiave nulla e vuota

Il server eXtraWay consente la valutazione della chiave *nulla*, storicamente la chiave che indica l'assenza dell'elemento o attributo in un record, ovvero la sua presenza ma in assenza di un effettivo contenuto.

A partire dalla versione Enterprise questa funzionalità subisce un'importante modifica, vediamo quale e come farne uso.

In primo luogo per poter utilizzare ad ampio spettro le chiavi nulle è stata aggiunta la possibilità di indicare la richiesta di calcolare le chiavi nulle su tutti i canali dichiarati esplicitamente. Per farlo è sufficiente aggiungere l'attributo `empty_key` con valore positivo ('yes') nell'elemento `primary_node`.

In seguito il funzionamento del concetto di chiave nulla è stato posto in discussione compiendo un necessario distinguo tra chiave *nulla* e chiave *vuota*.

Iniziamo col dire cosa si intenda per chiave *nulla* e *vuota*. Un "campo" è nullo quando non gli viene assegnato alcun valore. È vuoto quando gli viene assegnato un valore vuoto. Ci si riferisce ad un "campo" di tipo alfanumerico (nel caso dei numerici il distinguo non può essere fatto).

Esistono due modalità di implementazione diffuse tra i diversi produttori: una prevede che ci sia differenza tra campo *nullo* e campo *vuoto*¹⁷⁾. Un'altra prevede invece che non ci sia alcuna distinzione ed un campo non valorizzato o valorizzato con un valor vuoto si considera comunque un campo *nullo*.

Questo secondo approccio corrisponde a quello adottato da eXtraWay nel corso degli anni e per questa ragione rappresenta il default. Non di meno è stata introdotta la possibilità di procedere con la modalità opposta.

Per ottenere questo risultato si deve intervenire su una voce di profilo, `arc.empty_is_null`, il cui valore di default è `true`.

Impostando a `false` questo profilo si ottiene una distinzione tra le due modalità.

<code>arc.empty_is_null</code>	<code>true</code>	<code>false</code>
Chiavi Create	Viene creata solo la chiave <code>&null</code> ; sia per campo assente che per campo vuoto.	Viene creata la chiave <code>&null</code> ; in caso di assenza del campo e <code>&empty</code> ; in caso di campo esistente ma vuoto

Configurazione Seriali

Nella stragrande maggioranza degli archivi sono presenti valori che vengono definiti *Seriali*. Essi sono valori solitamente numerici (ma non per forza e non soltanto) che il server assegna ad un documento per fornire, ad esempio, un codice univoco, una chiave primaria o un valore *Protocollare*.

Nel paragrafo precedente si è parlato di accoppiare una chiave ad un valore seriale tramite un ID. In sostanza si indica che tale chiave deve rispettare le regole indicate in questa sezione mentre ciascuna regola può trovare applicazioni su chiavi appartenenti a *Primary Nodes* differenti¹⁸⁾.

Questa sezione consiste di un elemento *serials* che contiene uno o più elementi *serial* ciascuno dei quali descrive una precisa regola.

Vediamo il seguente esempio cui seguirà la descrizione degli attributi.

```
<serials>
  <serial id="protocollo" mask="#####" prefix="yes" empty="yes"/>
  <serial id="numero_record" mask="%#####" />
</serials>
```

Gli attributi che si possono impostare sono:

Attributo	Singificato
id	Indica in modo univoco una regola seriale. Deve trattarsi dello stesso valore applicato all'attributo <i>serial_id</i> negli elementi <i>key</i> che dichiarano le chiavi
mask	Descrive la maschera seriale. Essa può essere composta da numeri (rappresentati dal carattere '#'), lettere (rappresentate dal carattere '%') e costanti espresse direttamente. La maschera seriale 'A%###' produrrà valori seriali da 'AA01' sino a 'AZ99'



Attributo	Significato
prefix	Indica che la componente seriale è solo la <i>Coda</i> del valore associato. In altre parole, nel campo potrà essere presente una stringa preposta al valore seriale ed il server attribuirà un valore progressivo per ciascun prefisso impostato. L'adozione del prefisso compete all'applicazione.
empty	Può valere 'yes' o 'no'. Se è ammesso il valore 'yes' il server consente che un documento venga salvato senza alcun valore in tale campo. L'assegnazione ¹⁹⁾ potrà avvenire in un momento successivo, modificando il documento
skip_test	Anche questo attributo può valere 'yes' o 'no'. Il server compie sempre e comunque il controllo sul contenuto dei campi seriali, anche se già valorizzati. Se impostato a 'yes' il server accetta valori di questo campo anche se non corrispondenti alla maschera. Si presta per tollerare valori che provengano da un import di dati da fonti diverse o qualora sia necessario modificare la maschera seriale in corso d'opera
slack²⁰⁾	Questo attributo può assumere valori positivi o negativi. In caso di valore positivo il seriale si considera <i>lasco</i> . Questo comporta che per tale seriale è lecito richiedere valori che poi non verranno utilizzati in quanto si tratta di un seriale che non ha necessità di garantire la progressività. Ovviamente, al fine di garantire compatibilità col passato, il valore di default è negativo così che solo per i seriali appositamente configurati sia consentito questo approccio. Altrettanto ovvio è che quest'impostazione non va mai messa sui seriali detti <i>protocolli</i>

Configurazione Thesaurus

Gli archivi eXtraWay si avvalgono sovente di Thesaurus di termini relazionati tra loro. Ciò viene particolarmente utile per il trattamento di vocabolari di termini chiusi tra i quali un operatore deve scegliere così come, in presenza di un ricco thesaurus di termini della lingua italiana opportunamente relazionati, per compiere estensioni lessicali alle selezioni effettuate. Per ragioni storiche, negli archivi eXtraWay vengono previsti solitamente due Thesaurus, uno per i termini in relazione ed uno per quelli che venivano definiti vocabolari vincolati²¹⁾. Essi sono identificati rispettivamente dalle etichette ths e thv. In questa sezione si indicano quali files di thesaurus usare (se quello già presente nell'archivio ovvero uno specifico) e come compiere gli accoppiamenti. Vediamo un esempio pratico.

```
<thesaurus>
  <th_file th_filename="../common.ths" th_type="ths">
    <th_link xw_key="XML,/doc/note" th_key="arrivo"/>
    <th_link xw_key="XML,/doc/classif/@cod" th_key="classif"/>
    <th_link xw_key="XML,/doc/tipologia/@cod" th_key="interno"/>
    <th_link xw_key="XML,/doc/mezzo_trasmissione/@cod" th_key="partenza"/>
    <th_link xw_key="XML,/doc/@tipo" th_key="varie"/>
    <th_link xw_key="XML,/doc/ordine/tipologia_ordine" th_key="tipologia_ordine"/>
    <th_link xw_key="XML,/doc/ordine/servizio" th_key="servizi_ordine"/>
    <th_link xw_key="XML,/doc/ordine/contratto" th_key="contratti_ordine"/>
    <th_link xw_key="XML,/doc/ordine/licenza" th_key="licenze_ordine"/>
    <th_link xw_key="XML,/doc/ordine/altro" th_key="altro_ordine"/>
  </th_file>
</thesaurus>
```

L'esempio chiarisce come la configurazione abbia luogo entro un elemento *thesaurus*. In esso possono essere presenti più elementi *th_file* ciascuno dei quali identifica un file²²⁾ ed un tipo (*th_type*), secondo le etichette viste precedentemente. In fine, in tale elemento si vengono a trovare gli accoppiamenti tra i canali dell'archivio e quelli del thesaurus. I canali dell'archivio, espressi nell'attributo *xw_key* dell'elemento *th_link*, devono essere quelli descritti nelle singole *primary_node* negli elementi *key²³⁾* e l'accoppiamento avviene indicando nell'attributo *th_key* il nome logico della chiave del thesaurus²⁴⁾.

Configurazione Titoli

La configurazione dei titoli si attua per mezzo di una complessa sintassi che deriva da quella esistente in */HighWay*. Prevede l'esistenza di una voce di profilo²⁵⁾ di tipo "title" il cui valore è appunto la sintassi di composizione. Sinteticamente si può indicare tale sintassi come una sequenza di "campi" e "diciture" racchiuse tra doppi apici²⁶⁾. Ogni campo si identifica allo stesso modo in cui si dichiarano gli *XPath* delle chiavi nella dichiarazione delle *primary_node* e possono essere seguiti da alcuni modificatori²⁷⁾. Le diciture, invece, sono semplici stringhe racchiuse tra doppi apici. Anch'esse possono contenere dei modificatori. Vale la pena soffermarci su di essi:

Modificatore	Significato
~	Rappresenta il condizionamento alla componente che segue o precede. Se collocato in testa ad una dicitura essa sarà presente <u>se e solo se</u> l'elemento o l'attributo indicato precedentemente è presente con un valore non nullo. Analogamente, se collocato in coda ad una dicitura essa sarà presente <u>se e solo se</u> l'elemento o l'attributo indicato successivamente è presente con un valore non nullo.
^	Rappresenta l'intenzione di eliminare gli spazi che il server pone normalmente tra le singole componenti per far sì che campi e diciture vengano uniti in una stringa unica. Se presente contemporaneamente al modificatore '~' quest'ultimo dev'essere più "esterno"

Vediamo alcuni esempi.



La sintassi “(~) <campo> ” ~ ” fa sì che il valore del campo sia racchiuso tra parentesi ma le stesse non appariranno se il campo non è valorizzato.

Analogamente, la sintassi può essere estesa come segue: “ ^ (~ ” <campo> ” ~ ^) ”. In questo modo gli spazi compresi tra le parentesi verranno rimossi mentre verranno mantenuti al di fuori di esse in quanto espressamente dichiarati nella dicitura stessa.

Ruolo dei Titoli

Ciò di cui abbiamo sino ad ora parlato viene sovente definito “Titolo” per ragioni più che altro storiche. La tecnologia *HighWay* introduceva questo concetto come forma sintetica di rappresentazione del documento selezionato quando l'esito di una selezione viene mostrato, ad esempio, in forma paginata.

Se pure questa finalità è del tutto presente anche in *eXtraWay*, di fatto il titolo del documento ha assunto col tempo un ruolo ben diverso. Esso, infatti, viene usato per almeno altri due scopi, ovvero la costituzione di titoli aventi una regola differente ma che comprenda solo parti presenti nel titolo di default) e l'ordinamento dei documenti per valori anch'essi dichiarati nella regola di composizione del titolo.

In pratica, quindi, quello che era il “Titolo” dei documenti è divenuta una sorta di “Cache” delle sue componenti salienti.

Va da sé che concepire un titolo che sia sufficientemente completo ed esaustivo consente alle applicazioni di cambiare la regola di composizione a *run time* ovvero richiedere ordinamenti di diversa natura avvalendosi di prestazioni sicuramente ottimizzate.

Perché questo avvenga, però, bisogna tener conto di alcuni accorgimenti. Nel capitolo precedente si è posta l'attenzione sui modificatori delle diciture. Esse non hanno ragione d'essere se si pensa al “Titolo” come “Cache” ma non di meno sono spesso presenti e corredate dai modificatori indicati.

Bisogna prestare attenzione al fatto che il server non può utilizzare come “Cache” del documento alcuna parte che si trovi dopo l'espressione di uno dei modificatori suddetti. Ne consegue che, per sfruttare al meglio le parti che si intende riutilizzare per ordinamenti e titoli diversi, si deve avere l'accortezza di elencarli in testa alla regola di composizione del titolo.

Configurazione Search Alias

La premessa di questo paragrafo è che i cosiddetti *Search Alias* non hanno alcuna funzionalità specifica per il server. Essi vengono resi disponibili a beneficio delle applicazioni che possono in tal modo non doversi preoccupare di quale sia l'effettivo *XPath* di una chiave.

In sostanza si tratta di compiere un accoppiamento tra il percorso *XPath* della chiave dichiarata negli elementi *key* delle *primary_node* con un'etichetta logica che abbia senso per l'applicazione finale.

Ciò avviene tramite un elemento *hw_fields* che contiene tanti elementi *search_alias* con 2 attributi, *search_name* e *search_key*.

L'attributo *search_name* rappresenta l'alias, l'etichetta logica nota ed interessante per l'applicazione finale mentre l'attributo *search_key* riporta il valore di *XPath* citato nell'attributo *name* dell'elemento *key* delle varie *primary_node*.

E' importante notare che possono esistere più alias per lo stesso percorso di chiave mentre non è vero l'opposto tranne in un caso che fa eccezione, ovvero il caso del “campo chiocciola”.

Per ragioni “storiche”, la ricerca globale viene effettuata compiendo una selezione ove il campo indicato è '@'. Questo porta il server ad avere comportamenti particolari che possono essere descritti come segue:

Stato	Comportamento
Non esiste alcuna <i>key</i> o <i>primary_node</i> che sia stata configurata per l'uso della chiave globale e non esiste alcun <i>search_alias</i> per l'etichetta '@'	Il server estende la ricerca a tutte le chiavi identificate per tutte le <i>primary_node</i> dichiarate
Indipendentemente dalla costituzione delle chiavi globali, esiste almeno un <i>search_alias</i> per l'etichetta '@'	Il server applica a tutti i canali di ricerca il cui alias è stato dichiarato come '@' la ricerca in esame
Non è stato dichiarato alcun <i>search_alias</i> con etichetta '@' ma esiste almeno un campo che affluisce nella chiavi globali	Il server estende la selezione solo ai campi globali espressamente dichiarati

Configurazione Profili Generali

I profili generali sono suddivisi a loro volta in diverse aree che si riferiscono ad insiemi di funzionalità o comportamenti del server. La sintassi con la quale viene indicata una voce di profilo è la seguente:

```
<profile type="..." value="..." />
```

Dove:

type	Indica la voce di configurazione che si vuole impostare
value	Indica il valore da assegnare a tale voce di configurazione

1. Impostazioni Generali

1. `type="codepage"`. [String] Obsoleto, caduto in disuso.

2. Titoli

1. `type="title"`. [Stringa] Regola di composizione dei titoli per tutte le unità informative dell'archivio. Ad esso può affiancarsi una specifica per ogni tipologia di unità informativa ma poiché questa seconda modalità non consente di avvalersi del titolo prodotto per operazioni, ad esempio, di ordinamento, essa è scarsamente utilizzata.
2. `type="title.mixed"`. [Boolean, default=false] Obsoleto, di futura abolizione. Indica, se impostato a “true”, che gli elementi citati nella regola di cui al punto precedente devono essere considerati in modalità “container” ovvero si



deve considerare come loro il contenuto anche degli elementi in essi nidificati. Serve appunto per gestire doverosamente gli XML che hanno una conformazione *mixed*.

3. `type="title.hier"`. [String] Regola di composizione titolo da applicare nella visualizzazione gerarchica dei documenti. Pressoché inutilizzato.
4. `type="title.separ"`. [Char] Indica il carattere da utilizzare, in sede di generazione del titolo di un documento, per separare le diverse istanze di un elemento ripetibile che risulti presente nel titolo.
5. `type="title.lazy"`. [Boolean, default=false] Indica che i titoli dei documenti non devono essere realizzati *on line* ovvero all'atto di inserimento o modifica del documento, bensì essere rimandati ad un momento successivo, regolato da un servizio appartenente al Server Master. E' facoltà di tale servizio rimandare ad esempio ad un periodo notturno il calcolo di tali titoli.
6. `type="title.delay"`. [Boolean, default=true] Indica che i titoli dei documenti non devono essere realizzati *on line* ovvero all'atto di inserimento o modifica del documento, bensì in corrispondenza della prima operazione che ne fa richiesta.

Questa voce di profilo viene abolita nelle versioni 24 e successive in quanto diviene il comportamento di default.



Not yet implemented/released.

7. `type="title.xsl"`. [String, default=""] Nome del file XSL²⁸⁾ da utilizzarsi per produrre una speciale estensione della Cache dei titoli dei documenti. Tale estensione NON collabora alla realizzazione di titoli, ordinamento o indici ed è espressamente concepita per produrre un output XML da usarsi come forma arricchita del titolo, e quindi della *cache*, semplificando la vita all'applicazione che di tale *cache* deve avvalersi.
L'esito della lavorazione, sino alla realizzazione di comandi specifici, sarà prodotto nel titolo stesso richiedendo in esso il canale TITXSL.



Not yet implemented/released.

3. Estensione XSLT

1. `type="cache.xslt"`. [String] Nome del file, completo di percorso assoluto o da considerarsi relativo alla collocazione dell'archivio, per mezzo del quale compiere un'elaborazione del XML del documento al fine di realizzare un semilavorato, sempre XML, da utilizzarsi per particolari costituzioni di titoli, vocabolari di chiavi o per gli ordinamenti. Si applica a tutte le unità informative se non è stata data esplicita dichiarazione in esse del file XSL da utilizzare.

4. Trattamento degli allegati.

1. `type="attach.maxperdir"`. [Number, default=100] Nella dislocazione degli allegati in directory numerate, numero massimo di file per directory.
2. `type="attach.substext"`. [String] Elenco delle estensioni, rappresentate col punto in testa e separate da spazi, per le quali è ammessa la sostituzione degli allegati.
3. `type="attach.aliasext"`. [String] Elenco delle estensioni, rappresentate col punto in testa e separate da spazi, degli allegati il cui contenuto viene considerato testuale al pari dei files .txt.
4. `type="attach.skipext"`. [String] Elenco delle estensioni, rappresentate col punto in testa e separate da spazi, degli allegati il cui contenuto non dev'essere sottoposto ad indicizzazione.
5. `type="attach.idhome"`. [String] Il valore è rappresentato da una sequenza di path separati da virgola. In tali percorsi, partendo dal primo, si vanno a cercare tutti gli allegati identificabili per mezzo di un ID numerico quando essi non sono rilevabili presso la loro collocazione naturale, vale a dire nella directory denominata analogamente al file XML che ospita il record che li riferenzia. Ciascun path può essere seguito da una label racchiusa tra parentesi tonde atta ad indicare su quale repository eventualmente rimovibile si sta tentando di operare²⁹⁾.
6. `type="attach.varhome"`. [String] Il valore è rappresentato da una sequenza di path separati da virgola. In tali percorsi, partendo dal primo, si vanno a cercare tutti gli allegati espressi con un percorso relativo e NON con un ID numerico. Il percorso relativo viene combinato con il path indicato e così si compie la ricerca dell'allegato.
ATTENZIONE: i percorsi da indicare in questa voce devono essere assoluti. Se viene introdotto un percorso a sua volta relativo, ad esempio per semplificare la portabilità dell'archivio, tutti i percorsi successivamente indicati verranno ignorati.
7. `type="attach.remove"`. [Boolean] Consente, se impostata a valore true, che il thread che si occupa della rimozione degli allegati non assegnati si occupi di quest'archivio.
8. `type="attach.allowdel"`. [Boolean] Indica, se impostata a valore true, che nel presente archivio gli allegati non più associati ad alcun documento verranno fisicamente rimossi dal file system.
9. `type="attach.longnames"`. [Boolean, default='false'] Indica che la gestione degli allegati deve adottare la denominazione estesa introdotta dal server della versione 25. Consente di utilizzare una politica di distribuzione degli allegati che tutela maggiormente dal rischio di *affaticamento* del file system dovuto al riempimento di cartelle con un elevato numero di files.

5. Trattamento degli separatori.

1. `type="separs.default"`. [String] Consiste in un'elencazione di valori numerici corrispondenti al carattere che dev'essere impostato come separatore. Essi si *aggiungono* ai separatori di default³⁰⁾ Si possono indicare singoli valori separati da spazi o intervalli di valori rappresentati da due numeri uniti da un trattino ('-').
2. `type="separs.erase"`. [String] Consiste in un'elencazione di valori numerici corrispondenti al carattere che non devono essere impostati come separatori. In sostanza si contrappone al precedente profilo e serve a rendere

operativi separatori definiti tra quelli di default. Si possono indicare singoli valori separati da spazi o intervalli di valori rappresentati da due numeri uniti da un trattino ('-').

6. Trattamento del correttore ortografico.
 1. `type="spellchk.rule"`. [String] Regola di spell checking.
7. Trattamento del catalogo.
 1. `type="catalog.strict"`. [Boolean] Indica, se `true`, che nel costituire un catalogo verranno riconosciute solo le unità informative dichiarate nel file `<nomearchivio>.conf.xml`. In caso contrario, tutti gli elementi figli dell'elemento radice verranno considerati come valida Unità Informative.
8. Trattamento delle estensioni in ricerca.
 1. `type="extension.top"`. [Num] Limite oltre il quale non compiere estensioni delle chiavi in modo parallelo per ragioni prestazionali.
9. Trattamento degli ordinamenti.
 1. `type="sort.defaultrule"`. [String] Regola di ordinamento applicata automaticamente a tutte le ricerche.
 2. `type="sort.defaultlimit"`. [Num] Numero documenti oltre il quale non effettuare l'ordinamento di default. Se non impostato l'ordinamento si compie sempre.
10. Trattamento degli documenti in editing.
 1. `type="doctail.maxsize"`. [Num] Dimensione massima della coda bianca da aggiungere in salvataggio ai documenti per ridurre il rischio di spostamento in successive modifiche.
 2. `type="doctail.minsize"`. [Num] Percentuale massima (rapportata alla dimensione del documento) della coda bianca da aggiungere in salvataggio ai documenti per ridurre il rischio di spostamento in successive modifiche.
 3. `type="doctail.maxperc"`. [Num] Percentuale massima (rapportata alla dimensione del documento) della coda bianca da aggiungere in salvataggio ai documenti per ridurre il rischio di spostamento in successive modifiche.
 4. `type="doctail.growonsite"`. [Boolean] Indicazione se la crescita di un documento oltre lo spazio per lui riservato nel file XML debba avvenire sul luogo (`true`) o gestita tramite spostamento dello stesso (`false`).
11. Trattamento dei check sum dei documenti.
 1. `type="fixity.signmode"`. [String] Indica l'algoritmo da utilizzare per compiere la firma dei documenti.
12. Trattamento delle impostazioni d'archivio.
 1. `type="arc.allownonportable"`. [Boolean] Se impostato a `true` indica che i files dell'archivio possono avere nomi non portabili da una piattaforma ad un'altra³¹⁾
 2. `type="arc.allownonstrict"`. [Boolean] Richiede, se impostato a `true`, che i controlli sulla congruenza dei record in sede di modifica vengano rilassati. Consente alle applicazioni un procedimento più snello ma introduce il rischio di sovrascrittura di record con dati che non tengano conto di precedenti modifiche. Quando non abilitato, ovvero in condizione di default, richiede che la configurazione `metadata` non sia disabilitata, pena errore in sede di apertura dell'archivio.
Default `false` al fine di garantire la massima sicurezza dei dati.



Not yet implemented/released.

3. `type="arc.bestbtnodesize"`. [Num, default=4] Indica la dimensione da adottare per i nodi dei BTree. La misura è in Kb quindi 0 corrisponde alla misura di default ovvero il nodo standard di 512 bytes.
4. `type="arc.flushdatasteps"`. [Num] Indica con quale frequenza compiere il flush effettivo dei dati per migliorare le prestazioni. Si riferisce al numero di salvataggi.
5. `type="arc.flushidxsteps"`. [Num] Indica con quale frequenza compiere il flush effettivo degli indici per migliorare le prestazioni. Si riferisce al numero di salvataggi. Se non indicato vale quanto espresso con 'arc.flushdatasteps'.
6. `type="arc.flushdatadelay"`. [Num] Sovrascrive ogni impostazione a `step`³²⁾. Indica ogni quanti secondi compiere il flush effettivo dei dati per migliorare le prestazioni.
Un valore negativo disabilita il *flush* dei dati senza ricadere nel caso a *step*.



Not yet implemented/released.

7. `type="arc.flushidxdelay"`. [Num] Indica ogni quanti secondi compiere il flush effettivo degli indici per migliorare le prestazioni.
Un valore negativo disabilita il *flush* degli indici senza ricadere nel caso a *step*.
Se non indicato o indicato inferiore al precedente, vale quanto espresso con 'arc.flushdatadelay'.



Not yet implemented/released.

8. `type="arc.attribfreeorder"`. [Boolean] Se `true` indica di non tener conto dell'ordine degli attributi di un elemento per migliorare le performance in modifica documenti.
9. `type="arc.repository"`. [Urn] Se si utilizza un repository esterno³³⁾ indica quale repository si stia usando e con quali caratteristiche.
10. `type="arc.exclude_xreg.strong"`. [String] Codice prodotto da 3D Informatica per inibire in modo forte la stesura del registro di sistema per l'archivio in esame.
11. `type="arc.exclude_xreg.light"`. [String] Codice prodotto da 3D Informatica per inibire in modo semplice la stesura del registro di sistema per l'archivio in esame.
12. `type="arc.wide_xreg"`. [Boolean] Se impostato (il valore di default è 'false') impone la registrazione di un set



supplementare di comandi nel registro³⁴⁾.

In particolare registra il caricamento documenti (operazione 22), il caricamento allegati (operazione 23) e l'aggiunta di nuovi allegati (operazione 24).

13. `type="arc.no_idx_cache"`. [Boolean] In via sperimentale, se impostato a `true` inibisce l'uso della cache in indicizzazione *On-Line*..
14. `type="arc.no_idx_attachment"`. [Boolean default=false] Per funzionalità di debug, se impostato a `true` inibisce l'indicizzazione dei contenuti testuali degli allegati.
15. `type="arc.arc.no_attachment_conversion"`. [Boolean default=false] Se impostato a `true` inibisce la creazione da parte del server *eXtraWay* dei file utilizzati dal *File Conversion Agent* per attivarsi e compiere le conversioni richieste sugli allegati³⁵⁾.
16. `type="arc.lazyindex"`. [Boolean] Se `true`, e se l'archivio è stato opportunamente configurato per avere chiavi primarie e secondarie, compie *On-Line* l'indicizzazione delle chiavi primarie e *Near-On-Line* quella delle chiavi secondarie.
17. `type="arc.lazyindex.off"`. [Boolean] Applica lo stesso criterio della voce `arc.lazyindex.off` all'indicizzazione *Off-Line*.
18. `type="arc.allow_moreunique"`. [Boolean] Consente, nel modificare un documento, che si violi l'univocità di un altro documento oltre a trovare corrispondenza piena col documento in modifica.
19. `type="arc.trust_indexes"`. [Boolean, default=true] In sede di inserimento, modifica, cancellazione documenti e/o di inserimento nuovi allegati, consente di non rendere bloccante l'operazione anche in presenza di indici non corretti affidando alle registrazioni dei seriali il compito di garantire univocità e serialità.



Not yet implemented/released.

20. `type="arc.test_wellformed"`. [Boolean, default=false] Se impostato a `false` inibisce i controlli di correttezza formale dei documenti dell'archivio in fase di salvataggio.
 21. `type="arc.test_encoding"`. [Boolean, default=true] Se impostato a `false` inibisce i controlli di correttezza dell'encoding dichiarato in fase di salvataggio. Disponibile dalla versione 22.1.3.6 del server *eXtraWay*.
 22. `type="arc.locktime.long"`. [Num] Tempo massimo in secondi per compiere locking e quindi accesso ad un archivio. Serve a sovrascrivere il default di 15 secondi.
 23. `type="arc.empty_is_null"`. [Boolean] Parametro che indica la modalità di calcolo della chiave nulla e se sia necessario o meno compiere il distinguo tra chiave *nulla* e chiave *vuota*.
13. Trattamento delle impostazioni di debug.
1. `type="debug.chkref"`. [Boolean] Verifica le catene dei riferimenti delle chiavi interessate dagli inserimenti e modifiche nel momento stesso dell'operazione.
14. Trattamento delle relazioni.
1. `type="rel.logicalid"`. [XPath] Percorso XML che indentifica il valore logico da utilizzarsi ad esempio in fase di esportazione.
15. Trattamento delle selezioni probabilistiche.
1. `type="probab.hitweight"`. [Num] Indica quanta percentuale del peso complessivo dato ad un unità informativa debba essere valutata sulla base del numero di termini selezionati e non sul peso ponderato della chiave coinvolta. Sovrascrive le impostazioni del file `xw.ini`.
 2. `type="probab.ranksize"`. [Num] Indica il numero massimo di unità informative da tornare.
 3. `type="probab.minranksize"`. [Num] Indica il numero minimo di unità informative da tornare.
 4. `type="probab.minrank"`. [Num] Indica la percentuale di somiglianza minima, al di sotto della quale non tornare documenti.
 5. `type="probab.similartuning"`. [Num, default=100] Nella ricerca per documenti simili la scelta dei termini da utilizzare per la ricerca si basa su un determinato grado di selettività. Sopra tale soglia, ovvero se il termine è troppo poco selettivo, la parola viene scartata. Per evitare che siano scartate parole anche significative su un archivio ove i termini hanno particolare distribuzione, si può intervenire su questa voce di configurazione. Le parole accettate sono quelle che selezionano 1/<numero> documenti in archivio.
16. Trattamento della ricerca.
1. `type="search.skipiw"`. [Boolean] Nel compiere le selezioni si richiede di produrre, per default, catene di riferimenti inerenti i documenti selezionati in cui le *iwords* vengono ignorate. Causa l'inversione del default usato solitamente dal server per velocizzare le selezioni.
 2. `type="search.wildcardprefix"`. [Num] La selezione può essere effettuata con l'uso di *Wild Cards*. L'abuso di questa modalità può condurre a selezioni eccessivamente complesse, solitamente frutto di un uso errato del sistema e quindi non realmente utili o significative. Per tutelarsi dagli abusi o dall'uso improprio delle *Wild Cards* in ricerca si può impostare questo valore che indica di quanti caratteri debba essere la radice priva di *Wild Cards* di una chiave perché la stessa possa considerarsi valida in questo tipo di ricerche.
 3. `type="search.similarprefix"`. [Num] La selezione può essere effettuata indicando la somiglianza tra termini ovvero selezionando termini che somiglino a quello impostato perché hanno caratteri in più, in meno o semplicemente diversi. L'abuso di questa modalità può condurre a selezioni eccessivamente complesse, solitamente frutto di un uso errato del sistema e quindi non realmente utili o significative. Per tutelarsi dagli abusi o dall'uso improprio delle somiglianze in ricerca si può impostare questo valore che indica di

quanti caratteri debba essere la radice, ovvero la parte iniziale comune a tutte le chiavi che non viene sottoposta al controllo di similitudine in modo da restringere sensibilmente il campo d'azione.

4. `type="search.englishsint"`. [Boolean] Per richiedere che l'archivio operi solo con sintassi inglese per AND, OR e NOT.
5. `type="search.maxSearchTime"`. [Num] Indica il tempo massimo, in secondi, oltre il quale una selezione viene interrotta perché troppo onerosa. Si riferisce agli utenti standard.
6. `type="search.maxLongSearchTime"`. [Num] Indica il tempo massimo, in secondi, oltre il quale una selezione viene interrotta perché troppo onerosa. Si riferisce agli utenti autorizzati a svolgere ricerche di "lungo corso".
7. `type="phono.flags"`. [String] Identificazione delle regole di default, ovvero flags, nell'estensione fonetica dei termini in ricerca.
8. `type="phono.charlimit"`. [Num] Numero massimo di caratteri di una chiave oltre il quale non procedere nella ricerca delle somiglianze fonetiche.
9. `type="phono.rule"`. [String] Etichetta che identifica, una ad una, le regole fonetiche impostate per un dato archivio. Dev'essere seguita da un numero arabo progressivo in base 1.

17. Triggers.

1. `type="trigger.user"`. [String] Nome del *Package* dal quale acquisire le funzioni esterne da chiamare come *Trigger* al verificarsi di determinati eventi. Vds. [Stored Procedures & Triggers](#).
2. `type="trigger.auth"`. [String] Nome del *Package* dal quale acquisire le funzioni esterne da chiamare come *Trigger* di autorizzazione al verificarsi di determinati eventi. Vds. [Stored Procedures & Triggers](#).
3. `type="trigger.auth.mode"`. [String, default=strong] Modalità di Autenticazione da applicare ai *Trigger* di cui al punto precedente. Prevede i valori strong e weak. Vds. [Stored Procedures & Triggers](#).

1)

Il case in cui i valori sono espressi è del tutto influente

2)

bensì acquisite da un XML esistente senza intervenire sul formato

3)

in grigio valori facoltativi e di default

4)

Si consulti la parte del presente documento che tratta la `file_location rule`

5)

Nome assoluto ma ancor meglio percorso relativo alla locazione naturale dei files XML

6)

documentata in altra sede

7)

Ci si riferisce al momento dell'inserimento del documento

8)

Non dimentichiamo che stiamo parlando di *Unità Informative* in XML

9)

Con la presenza di // nel mezzo

10)

Si applica su canali ove si voglia avere una chiave univoca indipendentemente dall'ampiezza del contenuto

11)

Per non correre il rischio di confondere le chiavi e consentire la loro consultazione distinta, le chiavi mono valore prodotte in questo caso, e solo nei campi indicizzati "double", viene prefissata ad uno spazio

12)

In questo caso la presenza dell'elemento/attributo privo di ogni valore ovvero la sua assenza si equivalgono e producono comunque la chiave vuota

13)

A meno che non siano stati configurate in modo esplicito chiavi che li dichiarano

14)

In indicizzazione la *Stop List* non viene utilizzata e tutti i termini producono chiavi che poi, di fatto, non vengono utilizzate se non in casi particolari

15)

Dalla quale deriva quella delle `file_location rules`

16)

Le chiavi speciali vengono solitamente distinte dagli elementi e dagli attributi per mezzo di un carattere '#'

17)

Si considera vuoto un campo che esiste ma non contiene nulla ovvero ha un contenuto che per la natura dell'indicizzazione che vi si applica, ad esempio nel caso di sola interpunzione in un campo multi valore, non produce chiavi.

18)

Non si prevede che due chiavi della stessa *Primary Node* possano essere accoppiate alla stessa regola.

19)

Che ha luogo imponendo nel campo un punto('.') o un prefisso seguito dal punto

20)

Server eXtraWay 25.5.0 o superiori

21)

Ovvero liste chiuse o semi chiuse

22)

Se si tratta del nome di default può bastare un punto ('-')



23)

Seguendo la stessa sintassi dell'attributo 'name' di 'key'

24)

Tali etichette sono elencate nel file .dsp del thesaurus stesso, documentato separatamente

25)

Vedasi il successivo capitolo sui profili d'archivio

26)

Per questa ragione l'attributo che contiene questa sintassi è solitamente racchiuso tra singoli apici per semplificarne la stesura e lettura

27)

Vds. documentazione specifica

28)

relativo alla posizione dell'archivio

29)

Formato: path[(label)] , ...

30)

Facendo riferimento alla tabella codici nota come *Windows-1252* i separatori di default sono identificati dalla seguente sequenza numerica: 1-47 58-63 91-96 123-130 132-137 139 141 143-153 155 157 160-191 247 255

31)

Ci si riferisce, in sostanza, al Case

32)

Vedi punti precedenti

33)

I documenti possono essere salvati in “contenitori” diversi dai files XML tra cui dei files binari proprietari della piattaforma eXtraWay oppure in Oracle

34)

Disponibile dalla versione 22.2.14.0 e 24.0.0

35)

Dalla versione EE 1.4.10