

# Frequently Asked Question

Questo capitolo ha lo scopo di dare le principali risposte alle domande più frequentemente poste in merito alla piattaforma eXtraWay.



La pagina è in continua evoluzione e verrà presto integrata con altra documentazione attualmente disponibile alla URL:

[http://intra.3di.it/manuali/xw\\_techref/faq/html/](http://intra.3di.it/manuali/xw_techref/faq/html/)

## Architettura delle Directory

### I files di un archivio

Essi sono

Estensione	Descrizione
<b>.stat.xml</b>	E' il file che contiene lo <i>Status</i> dell'archivio. In esso vengono registrati il numero complessivo dei documenti presenti, quanti di essi sono cancellati, quanti indicizzati in modo parziale e/o totale. Viene registrato il numero totale delle chiavi ed alcuni valori seriali. Nelle prime versioni del server conteneva anche l'elenco di tutti i valori seriali assegnati ora migrato nel file <i>.ser.xml</i> .
<b>.stat</b>	 <b>Not yet implemented/released</b> Le versioni successive sostituiranno il file XML con una versione binaria per ottenere maggior stabilità e prestazioni in fase di salvataggio degli archivi modificati.
<b>.lck</b>	File funzionale a regolare le concorrenze d'accesso. Con l'introduzione del file <i>.stat</i> al posto del file <i>.stat.xml</i> è presumibile che esso sia destinato a sparire.
<b>.conf.xml</b>	E' uno dei files più importanti dell'archivio ed attorno al quale un archivio può essere generato. Dichiarare tutte le informazioni necessarie per riconoscere e mappare le Unità Informative, creare per ciascuna componente di esse un sistema di vocabolari ampio e versatile e compiere combinazioni speciali di chiavi. Dichiarare tutte le <i>short hand</i> che le applicazioni possono utilizzare per velocizzare l'accesso alle chiavi stesse, le regole di univocità sulle Unità Informative e le politiche d'archivio per compiere sul <i>file system</i> uno storage delle Unità Informative in files XML logicamente organizzati, denominati e distribuiti in <i>directories</i> . Consente anche l'impostazione di profili specifici per il tuning d'archivio che consentano ad esempio maggiori performance o l'uso di funzionalità customizzate tramite, ad esempio, librerie dinamiche.
<b>.ser.xml</b>	Introdotta da diverso tempo, contiene lo stato di tutti i valori seriali emessi così da poter rapidamente determinare quale sia il prossimo valore candidato <sup>1)</sup> .
<b>.udp/.udd</b>	E' la coppia di files che rappresenta la vera e propria <i>Mappa</i> dell'archivio. In essi sono contenuti tutti i riferimenti ai files XML nei quali sono collocate le singole <i>Unità Informative</i> con indicazioni della posizione e della dimensione. Il file <i>.udd</i> viene chiamato in causa se e solo se ci sono componenti supplementari <sup>2)</sup> ovvero se il percorso del file XML sia troppo ampio per essere collocato nel file <i>.udp</i> <sup>3)</sup> . Nelle versioni meno recenti in tale file trovano collocazione anche i <i>Titoli</i> <sup>4)</sup> delle Unità Informative che sono stati migrati nelle versioni più recenti nei files <i>.tip/.tit</i> .
<b>.idx/.vcb/.ref</b>	Terna di files che contiene i veri e propri indici. Il file <i>.idx</i> contiene tutti i vocabolari e quindi tutte le chiavi che possono essere utilizzate in selezione. Per ciascuna di esse esiste un record descrittivo del file <i>.vcb</i> che a sua volta indica nel file <i>.ref</i> dove si trovi e quali siano le caratteristiche della catena di riferimenti che conduce ai singoli documenti identificabili da ogni singola chiave.
<b>.idx.bpt/.idx.edt</b>	 <b>Not yet implemented/released</b> Le versioni successive sostituiranno l'attuale BTree con un B+Tree per incrementare stabilità e prestazioni.
<b>.lck</b>	File funzionale a regolare le concorrenze d'accesso. Con l'introduzione del file <i>.stat</i> al posto del file <i>.stat.xml</i> è presumibile che esso sia destinato a sparire.
<b>.ths/.thv</b>	Thesaurus dell'archivio. Si prestano alla conservazione e gestione di piani di classificazione nidificati, vocabolari vincolati ed altre combinazioni di termini secondo domini e criteri detti relazioni. Utilizzato anche per comporre Thesaurus della lingua italiana <sup>5)</sup> come supporto alle estensioni linguistiche delle ricerche.
<b>.tip/.tit</b>	(Re)Introdotti recentemente per il trattamento dei Titoli dei documenti precedentemente descritto in merito al file <i>.udd</i> . L'uso di questa coppia di files risulta più vantaggioso e performante quindi sarà sovente preferito al precedente. Nelle versioni più recenti l'uso di questi files è stato reso dinamico, e quindi separato e distinto dal trattamento delle Unità Informative, per accrescere ulteriormente le performance.  A questi titoli viene affiancata inoltre un'ulteriore <i>cache</i> prodotta con l'utilizzo di files XSL applicati al documento, da non confondere con l'estensione XSL del documento stesso che ha finalità e risultati diversi. Dal momento che lo sviluppo di questa componente è ancora in corso non si entra ancora nel dettaglio.

### Il compattamento Dati ed Indici

#### Compattamento Dati

Pagina in allestimento



## Compattamento Indici

Il compactamento degli indici è una procedura che ha senso di essere effettuata fondamentalmente per guadagnare prestazioni nell'accesso agli indici. Questo avvantaggia gli archivi intensamente consultati ma, in qualche modo, confligge con archivi altresì modificati in inserimento e modifica, quindi archivi dinamici.

Quando tutti gli indici di un archivio sono nell'ordine in cui essi vengono consultati dai diversi vocabolari, il server riesce ad accedere ad essi molto più rapidamente. Quando però un archivio è dinamico, quindi soggetto quanto meno a modifiche, lo stato di compactamento degli indici degrada rapidamente e non si può dire che l'ordine degli indici rappresenti un effettivo vantaggio in fase di modifica.

Ecco quindi che si suggerisce di ricorrere al compactamento degli indici principalmente negli archivi storici a sola consultazione. Non di meno, il compactamento degli indici è una procedura che viene compiuta pressoché in modo automatico<sup>6)</sup> ogni qual volta si procede alla ricostruzione integrale degli indici<sup>7)</sup>.

Vediamo come si comporta il server in questo caso e quali siano i possibili problemi che la procedura può incontrare.

Iniziamo col dire che il server genera, compiendo il compactamento, 3 files che sostituiranno gli originali e da loro denominazione modificando il primo carattere del nome dell'archivio originario. Se il nostro archivio si chiama 'abc', i files frutto del compactamento si chiameranno '@bc' ed altri files di servizio che vedremo poi si chiameranno '#bc'. I files interessati sono quelli che rappresentano i soli indici, quindi .idx, .vcb e .ref già descritti in precedenti paragrafi.

Partendo dai files *nomearchivio.idx*, *nomearchivio.vcb* e *nomearchivio.ref*, il server produrrà prima una terna di files *@omearchivio.idx*, *@omearchivio.vcb* e *@omearchivio.ref* aventi lo stesso contenuto ma nell'ordine richiesto.

Giunti a questo punto sarà necessario compiere una sorta di *rotazione* dei files che consenta agli altri server che già operano con i files stessi di trovarsi aggiornati nei contenuti senza rendersi conto dell'avvenuto compactamento. Perché questo avvenga e non si rischino perdite di contenuti, la procedura è la seguente:

1. La terna di files originali *nomearchivio* viene 'copiata' in files *#omearchivio*.
2. La terna di files compactati *@omearchivio* viene 'copiata' sui files originali.
3. Una volta avvenuta questa doppia copia, i files *@omearchivio* e *#omearchivio* possono essere rimossi.

Per quanto la procedura non nasconda particolari complicazioni, qualcosa può andare storto, specialmente se lo spazio disponibile su disco non risulta sufficiente a contenere la doppia copia dell'archivio stesso<sup>8)</sup>. La procedura può limitarsi a presentare errore<sup>9)</sup> oppure indicare che l'operazione ha avuto successo ma ci sono stati problemi di copia. In questo caso avremo l'archivio in una condizione incompleta che può ricadere in uno dei seguenti casi.

- Esiste la terna *nomearchivio* ed esiste la terna *@omearchivio* ma la terna *#omearchivio* è incompleta o i files non sono stati copiati per intero. Mancava spazio nel fare la copia di sicurezza. L'archivio risulta funzionante con gli indici che sono ancora "non compactati". La soluzione preferibile è quella di rimuovere i files *@omearchivio* e gli eventuali *#omearchivio* lasciando l'archivio non compactato<sup>10)</sup>. Se il problema era effettivamente lo spazio su disco si può cercare di renderne disponibile una maggior quantità e compiere nuovamente il compactamento.
- Esistono tutte e 3 le terne di files ma la copia s'è interrotta, credibilmente, durante la migrazione da *@omearchivio* a *nomearchivio*. Si può procedere in due modi:
  - Rimuovere la terna di files *#omearchivio* e completare manualmente la sostituzione dei files *nomearchivio* con i files *@omearchivio*. Il risultato sarà l'archivio compactato.
  - Rimuovere la terna di files *@omearchivio* e ripristinare manualmente i files *nomearchivio* grazie alle copie di sicurezza rappresentate dalla terna *#omearchivio*. Il risultato sarà l'archivio nella sua forma originaria prima del compactamento. In linea di massima le due soluzioni si equivalgono ma questa seconda, come nei casi precedenti in cui si torna allo stato originario dell'archivio, è da preferire.

Esistono versioni del server che possono compiere il compactamento anche in modalità "non sicura" ovvero senza avvalersi dalla terna di files *#omearchivio*. In tal caso se la copia dei files *@omearchivio* in *nomearchivio* non è avvenuta completamente, essa può essere completata manualmente.

## Organizzazione degli Allegati

Ogni documento può contenere riferimenti ad uno o più allegati.

Rif. implicito ed esplicito.

Pagina in Allestimento

### Cosa contiene la directory <nomearchivio>.file

La directory <nomearchivio>.file contiene, in senso assolutamente generico, allegati ai documenti dell'archivio in esame, ma di fatto assolve a più compiti e tali allegati sono organizzati in maniera differente tra loro.

Perché le successive spiegazioni siano chiare al lettore si assume che abbia già conoscenza del capitolo [Organizzazione degli Allegati](#) che identifica gli allegati referenziati in modo implicito da quelli referenziati in modo esplicito.

In prima approssimazione possiamo dire che la directory <nomearchivio>.file contiene:

1. Nella sua radice i files che vengono detti [in area di parcheggio](#)
2. Nelle sue directory numerate (es.: 000000\...) i files frutto della migrazione da una precedente architettura ovvero di un'organizzazione intenzionalmente svincolata dai documenti XML. Tali files vengono comunque referenziati in modo

implicito.

3. In ogni altra directory, files referenziati in modo esplicito con percorso relativo.

## Perchè vi è un proliferare di files nella directory <nomearchivio>.file

La directory <nomearchivio>.file svolge il compito di area di parcheggio nell'ambito del [trattamento degli allegati degli archivi eXtraWay](#).

Per comprendere bene cosa si intenda per area di parcheggio è opportuno comprendere prima il flusso operativo che consente di associare un allegato ad un documento nella piattaforma eXtraWay. Il flusso è identificabile con i seguenti punti:

1. Ad eccezione degli allegati di tipo esplicito (Vds. [Organizzazione degli Allegati](#)), quelli implicitamente referenziati non sono proprietà del server eXtraWay. L'applicazione che intende associare un allegato ad un documento nuovo o esistente dovrà quindi inviare per ciascuno di essi una copia al server il quale provvederà a dare loro un identificatore numerico univoco e collocarli nell'*area di parcheggio* ovvero nella directory <nomearchivio>.file. Il server risponde all'applicazione indicando quale identificativo numerico è stato assegnato a ciascun allegato.
2. L'applicazione completa il documento (in fase di inserimento o modifica) elencando, tra gli altri, anche i riferimenti implicito a tutti gli allegati che ha precedentemente inviato al server<sup>11)</sup>. Il documento viene inviato al server per il salvataggio.
3. Il server che riceve un documento da salvare, che si tratti di inserimento o modifica, ne verifica i contenuti e stabilisce quali identificatori di allegati si riferiscano a files che si trovano o meno in *area di parcheggio*. Quando avrà deciso [dove compiere il salvataggio del documento XML](#) sposterà presso di esso tutti gli allegati referenziati rimuovendo quindi quelli ancora in *area di parcheggio* e posizionandoli doverosamente.

Esiste quindi uno scollamento temporale tra il momento in cui l'applicazione invia gli allegati al server perché associ ad essi<sup>12)</sup> un identificatore univoco ed il momento in cui gli allegati inviati vengono effettivamente associati al documento, momento che corrisponde al salvataggio dello stesso.

Va da se che molte cose possono avvenire tra questi due momenti. L'operatore può rinunciare a salvare le modifiche effettuate oppure il contenuto del documento è incompleto o replicato e quindi il sistema si rifiuta di salvarlo o una qualsiasi altra attività simile.

In ogni caso se la seconda attività, quella di effettiva assegnazione di un allegato ad un documento, non ha luogo l'allegato che si trova in *area di parcheggio* non sarà più utilizzato da alcun documento ne il suo numero univoco potrà essere recuperato.

Potremmo quindi asserire che i files presenti in area di parcheggio che presentino una denominazione del tipo

```
<numero in 6 cifre>.<estensione del documento>
```

si possano rimuovere una volta che la loro datazione evidenzia che non può essere in corso un'attività d'assegnazione che li riguardi e quindi essi siano rimasti in quest'area a causa di attività non condotte a termine.

Nella precedente affermazione si è volutamente usato il *condizionale* in quanto l'unica garanzia che questo avvenga può darcela l'engine ed è per questo che esiste una specifica [procedura di eliminazione degli allegati non assegnati](#).

## Come posso rimuovere gli allegati non associati

Come annunciato nel capitolo inerente [l'area di parcheggio](#) in essa possono trovarsi files che si aveva l'intenzione di associare a documenti del database ma che sono rimasti inutilizzati. Per assicurarsi di rimuovere solo ed esclusivamente quelli che non risultano referenziati da alcun documento è stata implementata una [procedura automatica che svolge quest'attività in modo sicuro](#).

1)

Il server si riserva comunque di verificare la validità del valore desunto da tale file prima di compiere applicazione

2)

Ad esempio, l'estensione XSL di una unità informativa

3)

Il quale ha record a dimensione fissa

4)

Vengono ancora chiamati Titoli pur rappresentando una sorta di *cache* del documento utilizzata tanto in fase di visualizzazione sintetica dell'esito di una selezione, quanto in ordinamento ed altri procedimenti

5)

o altre lingue

6)

può essere evitata ma essa viene compiuta per *default*

7)

specialmente quando associata anche alla ricostruzione del catalogo

8)

Il server compie una valutazione preventiva dello spazio necessario ma essa è empirica ed il disco potrebbe essere contemporaneamente utilizzato e quindi riempito da altri programmi. Inoltre il compattamento degli indici è, di fatto, un ordinamento

9)

Ed in quel caso provvede a rimuovere i semilavorati non utilizzabili

10)

Se si sono riprese le attività gli indici potrebbero aver subito modifiche e quindi quelli compattati in *@omearchivio* potrebbero non essere più affidabili

11)



L'invio avviene per singoli files, uno alla volta

[12\)](#)

Per la precisione alla copia che eXtraWay trattiene presso l'archivio