



Dettagli sui comandi del Server eXtraWay

Di seguito vengono elencati i dettagli più importanti per fare uso dei comandi del server eXtraWay. Si ricorda che i comandi del server si dividono in due categorie, quelli di “vecchio ordinamento”, di tipo binario, proprietario, e quelli di “nuovo ordinamento” che invece sono stati realizzati come comandi XML¹⁾.

In particolare questo documento si prefigge di dare comprensibilità ai comandi di “nuovo ordinamento”.

Introduzione

Perché si possa comprendere di cosa questo capitolo tratta si tenga in considerazione che il comando XML che il server interpreta deve avere una struttura nota ed alcune componenti obbligatorie. Esso è caratterizzato da un elemento “cmd” che deve contenere per forza di cose un attributo “c” ed un attributo “bits”. Il primo indica il comando, o macro comando, ed il secondo i suoi modificatori. Il comando può contenere a sua volta un'area XML (o meno) contenuta in un elemento “bst”. Ecco quindi che avremo un comando semplice...

```
<?xml version="1.0"?>
<cmd c="..." bits="..."/>
```

...ed un comando complesso...

```
<?xml version="1.0"?>
<cmd c="..." bits="...">
  <bst>
    ...
  </bst>
</cmd>
```

Se l'attributo **c** indica il comando è può assumere un solo valore, l'attributo **bits** può assumere valori diversi ma soprattutto *combinazioni* di valori. Avremo quindi uno di questi scenari

Bitmap	Il valore espresso è una <i>maschera di bit</i> ovvero va interpretato come una sequenza binaria di '1' e '0' ciascuno dei quali assume un particolare significato ovvero un modificatore. Ad esempio il valore '7' corrisponde, per l'espressione binaria alla somma dei bits corrispondenti a '4' + '2' + '1' ⇒ bin'111'. Analogamente il valore '5' sarà '4' + '1' ⇒ bin'101'. Il primo indica che sono attive 3 opzioni che possono essere combinate ed il secondo indica che ne sono attive solo due
Sub-Comando	Il valore espresso è un valore secco in un elenco di sub-comandi. In questo caso valori come '1', '2' e '3' non indicano la combinazione binaria '01', '10' e '11' come nel caso precedente ma indicano 3 diversi sub-comandi del tutto distinti
Sub-Comando + Bitmap	Si tratta di una combinazione dei due casi precedenti, in vero piuttosto rara. Quando si presenta solitamente si sfruttano i 'bit alti' per il Bitmap e quelli bassi per il sub comando. Un esempio potrebbe essere il valore 0x82000005 che indica che sono attivi il bit '0x80000000', il bit '0x02000000' e che il sub comando assume valore '5'

Ai comandi possono poi essere associati altri attributi che assumono, di volta in volta, significati diversi.

E' convenzione comune che i valori numerici possano essere espressi in diverso modo, sarà il server a stabilire autonomamente se il valore indicato debba considerarsi decimale o esadecimale²⁾ secondo il seguente criterio.

Prefisso al valore	Tipo di valore identificato
h H 0x 0X x X	Valore esadecimale. Vengono presi in esame tutti i successivi caratteri compresi tra '0' e '9', tra 'a' ed 'f' e tra 'A' ed 'F' <i>Si noti che esprimendo un valore esadecimale, il numero di '0' presenti dopo il necessario prefisso e prima dell'espressione del valore effettivo è ininfluente. Ne consegue che così come 0x007f1c3b equivale a h007F1c3B, anche x7f1c3b assume lo stesso valore</i>
Nessun Prefisso	Valore decimale. Vengono presi in esame solo i successivi caratteri compresi tra '0' e '9' <i>Anche in questo caso l'eventuale presenza di '0' in testa al valore è ininfluente se pure usata più raramente</i>

Si ricorda che solo una parte di questi comandi prevede che vengano tornati valori estesi.

Per tutti i comandi in cui non viene espressamente esplicitato si assume che il ritorno sia positivo o negativo ed esso va interpretato come segue

ack e	Significato
1 == 0	Il comando ha dato esito positivo senza segnalare errori
1 != 0	Il comando ha dato esito positivo ma si sono verificati errori non bloccanti. Il codice d'errore è nell'attributo e ³⁾
0 != 0	Il comando ha dato esito negativo. Il codice d'errore è nell'attributo e
0 == 0	Combinazione non prevista

Di seguito tre semplici esempi.



```
<!-- Esito corretto -->
<rsp ack="1" e="0"/>
```

```
<!-- Esito corretto ma con segnalazione di errore latente -->
<rsp ack="1" e="882"/>
```

```
<!-- Esito non corretto. Errore di scrittura -->
<rsp ack="0" e="809"/>
```

Elenco Comandi

L'elenco dei comandi viene posto in ordine numerico progressivo.

Per semplificarne la lettura essi vengono espressi con il loro valore esadecimale.

L'elenco è attualmente provvisorio e verrà completato man mano che sorgerà l'esigenza di commentare un singolo comando.

Comando 0x00000004 (Informazioni generiche)

Questo comando consente di effettuare delle operazioni generiche

Avremo quindi...

c	0x04 (4)
bits	Può assumere uno dei seguenti valori 6 → elenca i record bloccati da un utente o quale sia l'utente che ha bloccato un record 7 → ottenere l'elenco degli archivi disponibili

Esempio del comando con **bits** valorizzato **7**

```
<?xml version="1.0" encoding="windows1252"?>
<cmd c="4" bits="7"></cmd>
```

Esempio risposta del comando con **bits** valorizzato **7**

```
<?xml version="1.0" encoding="windows-1252" ?>
<rsp ack="1" e="0">
  <dtl dtype="info" dval="3dgestio" />
  <dtl dtype="info" dval="acl" />
  <dtl dtype="info" dval="acque" />
  <dtl dtype="info" dval="albo" />
  <dtl dtype="info" dval="armi" />
  <dtl dtype="info" dval="articles" />
  <dtl dtype="info" dval="ascomunali" />
</rsp>
```

Comando 0x00000005 (Amministrazione Archivi)

(Documentazione in fase di completamento)

Questo comando, di fatto, racchiude una serie di comandi diversi, molti dei quali hanno scopi del tutto disgiunti.

Il valore di **bits** associato ad essi è quindi una particolare combinazione di valori progressivi ed a bit.

Nella *High Word* vengono usati i bits per modificare il singolo comando⁴⁾ mentre nella *Low Word* si trova il comando numerico.

Di seguito i comandi disponibili (In fase di allestimento)

Etichetta	Valore	Significato
CMDMGR_ARCADMIN_FULLCLONEARC	0x20	Comando per mezzo del quale si richiede che venga realizzato un archivio <i>Clone</i> dell'archivio indicato.
CMDMGR_ARCADMIN_REMOVEMINTENANCE	0x21	Comando per mezzo del quale si richiede che venga abbattuto lo stato di archivio in manutenzione.
CMDMGR_ARCADMIN_GETSERIAL	0x25	Comando per mezzo del quale si richiede vengano notificati i valori dei seriali interni al serve, quelli che vengono usati per l'attribuzione del prossimo nome di file XML ⁵⁾ ovvero il prossimo nome di file di <i>Attachment</i> .
CMDMGR_ARCADMIN_SETSERIAL	0x26	Comando per mezzo del quale si richiede venga impostato il valore di uno dei seriali interni al server, gli stessi indicati al punto precedente,.

Tra essi vale la pena ricordare una batteria di comandi solitamente utilizzata per realizzare degli archivi che sono una copia clone, totale o parziale, di altri archivi esistenti. Tale copia può poi essere realizzata secondo i criteri per il suo utilizzo su supporto ottico o meno.

Per semplificare la vita agli utenti che si prefiggono tale obiettivo è stato creato un particolare comando che consente, con una sola operazione, di selezionare i documenti necessari e produrre un archivio completo ed utilizzabile direttamente su supporti ottici etc.



etc.

Tale comando unisce il codice di bits 0x00000020 ad una serie di ulteriori flags.

Etichetta	Valore	Significato
CMDMGR_ARCADMIN_FCS_USEQUERY	0x80000000	Richiede che la clonazione d'archivio abbia luogo sulla base di una frase di selezione che si trova nell'elemento <bst>. In assenza di questo bit la funzione cerca gli attributi num e num2 per identificare gli estremi dei documenti dai quali produrre il nuovo archivio ovvero assume che la clonazione sia integrale.
CMDMGR_ARCADMIN_FCS_FORCESTAT	0x40000000	Richiede che si forzi la sovrascrittura del file <nomearchivio>.conf.xml dell'archivio che si intende clonare anche qualora esso sia già presente nella directory preposta alla clonazione. In assenza di questo bit il server usa il file di configurazione dell'archivio originale se manca uno specifico.
CMDMGR_ARCADMIN_FCS_RETITLE	0x20000000	Richiede che vengano calcolati i titoli dell'archivio clonato che, per definizione, viene clonato senza titoli.
CMDMGR_ARCADMIN_FCS_EXTREP	0x10000000	Richiede che l'archivio clonato usi un repository <i>Buddy File</i> anziché dei files XML. Incompatibile con CMDMGR_ARCADMIN_FCS_EXTREZIP.
CMDMGR_ARCADMIN_FCS_EXTREZIP	0x08000000	Richiede che l'archivio clonato usi un repository <i>Buddy File</i> compresso anziché dei files XML. Incompatibile con CMDMGR_ARCADMIN_FCS_EXTREP.
CMDMGR_ARCADMIN_FCS_LICENCE	0x04000000	Richiede il calcolo di una licenza di autoconsultazione. Un serve anche non abilitato riuscirà comunque ad accedere e gestire l'archivio. Incompatibile con CMDMGR_ARCADMIN_FCS_LICENCECD.
CMDMGR_ARCADMIN_FCS_LICENCECD	0x02000000	Richiede il calcolo di una licenza di autoconsultazione per supporti ottici. Un serve anche non abilitato riuscirà comunque ad accedere all'archivio. Incompatibile con CMDMGR_ARCADMIN_FCS_LICENCE.

Il comando tipo, per realizzare un archivio per CD rom, sarà quindi il seguente:

```
<?xml version="1.0" encoding="iso-8859-1"?>
<cmd c="5" bits="0xB2000020" arc="arcclone">
  <bst>([/doc/@anno]=2009)</bst>
</cmd>
```

Questo esempio crea un archivio clone dell'archivio corrente, che si chiamerà, come nome logico “arcclone” e sarà collocato parallelamente a quello di origine. La richiesta prevede che si clonino in questo secondo archivio solo i documenti del 2009.

Comandi per la gestione dei seriali.

Richiesta dei seriali

```
<?xml version="1.0" encoding="iso-8859-1"?>
<cmd c="5" bits="0x25"/>
```

Ottiene come risposta la seguente

```
<?xml version="1.0" encoding="iso-8859-1"?>
<rsp ack="1" e="0">
<dtl type="xml" value="0"/>
<dtl type="attach" value="183034"/>
</rsp>
```

Richiesta di impostare un seriale interno

```
<?xml version="1.0" encoding="iso-8859-1"?>
<cmd c="5" bits="0x26" serial="attach" num="183035"/>
```

I valori ammessi per l'attributo *serial* sono “xml” e “attach”.

Comando 0x00000007 (Comandi 'Inter Server')

(Documentazione in fase di completamento)

Questo comando consente di inviare a tutti i server (compreso o escluso il server *Master*) un comando da svolgere tra quelli di tipo amministrativo.

Il valore di bits indica le modalità con le quali il comando dev'essere svolto mentre il valore di num indica quale comando si intenda svolgere..

Di seguito i bits disponibili (In fase di allestimento)



Etichetta	Valore	Significato
CMDMGR_INTERSRV_WAIT	0x00000001	Indica che il comando deve produrre una risposta ed eventualmente interrompere la propria attesa allo scadere di un <i>time out</i> Documentazione incompleta.
CMDMGR_INTERSRV_BROADCAST	0x00000002	Indica che il comando dev'essere distribuito a tutti i processi eXtraWay ad esclusione del processo <i>Master</i> .
CMDMGR_INTERSRV_FULLBROADCAST	0x00000004	Indica che il comando dev'essere distribuito a tutti i processi eXtraWay includendo il processo <i>Master</i> .

I bits CMDMGR_INTERSRV_BROADCAST e CMDMGR_INTERSRV_FULLBROADCAST sono mutualmente esclusivi. In loro assenza dev'essere presente un attributo pid che indichi il *Process ID* del processo cui si vuole inviare il comando.

Di seguito i comandi (attributo num) disponibili (In fase di allestimento)

Etichetta	Valore	Significato
INTERSRV_CMD_RELOADCFG	0x00000001	Richiede che vengano ricaricate le configurazioni generali del server.
INTERSRV_CMD_RELOADARCCFG	0x00000002	Richiede che vengano ricaricate le configurazioni dell'archivio sul quale è stato richiesto il comando.
INTERSRV_CMD_SHUTDOWNARC	0x00000004	Richiede che l'archivio sul quale è stato richiesto il comando venga chiuso. Si pone come alternativa al comando precedente qualora esso non riesca a rinfrescare tutte le configurazioni necessarie.
INTERSRV_CMD_RELOADUSERCFG	0x00000020	Richiede che vengano ricaricate le configurazioni generali dell'utente che richiede il comando.

Esempi:

Richiede che tutti i server, tranne il Master, ricarichino le configurazioni dell'archivio sul quale si esegue il comando

```
<?xml version="1.0" encoding="iso-8859-1"?>
<cmd c="7" bits="2" num="2">
</cmd>
```

Richiede che tutti i server, Master compreso, chiudano l'archivio sul quale si esegue il comando

```
<?xml version="1.0" encoding="iso-8859-1"?>
<cmd c="7" bits="4" num="4">
</cmd>
```

Comando 0x00000008 (Esportazione)

Questo comando *Esportazione* consente di effettuare una esportazione di diverse componenti della base dati ma, principalmente, dei documenti della stessa.

Il comando principale (8) prevede un sub-comando, espresso nel campo *bits* che può prevedere a sua volta dei modificatori. E' quindi il classico caso di un comando che ha un sub-comando + bitmap

Vediamo di seguito come questi possano essere utilizzati.

Avremo quindi...

c 0x08 (8)	Comandi di esportazione in senso generale
------------	---

...ed avremo in 'bits' il codice del sub-comando richiesto. Si ricorda che esso potrà essere poi combinato con altri modificatori assumendo un valore misto.

Sub Comando (in 'bits')	Significato
0x00000001	Esportazione della Mappa dei documenti
0x00000002	Esportazione di una raccolta
0x00000003	Esportazione di una raccolta privata
0x00000004	Esportazione delle chiavi di un vocabolario
0x00000005	Esportazione delle chiavi di un Thesaurus
0x00000006	Esportazione di unità informative(documenti) in forma <i>complessa</i>
0x00000007	Esportazione di unità informative in forma <i>semplificata</i>
0x00000008	Esportazione di unità informative in modalità nidificata
0x00000009	Esportazione dei metadati delle unità informative
0x0000000A	Conversione di contenuti CSV in XML pronti per l'acquisizione via WatchDoc
0x0000000B	Conversione di contenuti CSV in XML ed acquisizione diretta via WatchDoc
0x0000000C	Realizzazione di un sub-archivio partendo dall'archivio corrente

In assenza di un modificatore di 'bits' che impone che l'esportazione avvenga senza utilizzare un file (0x80000000), tutti i comandi di esportazione prevedono la presenza di un attributo fn che deve definire il nome richiesto per l'esportazione. Tale valore può essere assoluto o relativo. Qualora relativo il suo punto d'origine si considera essere la directory dei temporanei del server.



Qualora tale nome non venga indicato, il server provvederà a generarne uno autonomamente⁶⁾

A questi sub-comandi possono applicarsi diversi ulteriori modificatori che verranno valutati di volta in volta.

Esportazione Mappa

Documentazione in fase di realizzazione

Esportazione Raccolte Pubbliche e Private

Documentazione in fase di realizzazione

Esportazione Chiavi Vocabolario e Thesaurus

Documentazione in fase di realizzazione

L'esportazione delle chiavi di un vocabolario richiede la presenza nel comando di alcuni attributi.

In particolare avremo

Attributo	Significato
id	Nome del canale da esportare. Esso può essere espresso con il percorso XML (e l'indicazione del canale XML o UD) ovvero in forma di <i>search alias</i> ovvero di etichetta configurata nel <i>file di configurazione dell'archivio</i> .
fn	Nome del file in cui compiere l'esportazione. Se assente il server ne assegna uno d'ufficio, del quale tornerà la denominazione nella risposta del comando. Indicando un'estensione .gz o .zip il server produce un file compresso secondo la modalità richiesta

Esistono altri parametri, meno frequentemente utilizzati.

Attributo	Elemento	Significato
key / key2		prima ed ultima chiave da esportare dal vocabolario indicato
num / num2		Indica la frequenza massima e minima che si richiede abbiano le chiavi da esportare. In assenza di esplicita dichiarazione le chiavi vengono esportate senza limitazione di frequenza quindi con una soglia massima pari alla dimensione dell'archivio. Si ricorda altresì che alcune versioni di server hanno un valore di default minimo impostato a '0' e quindi potrebbero esportare più chiavi del desiderato.
sel		Indica di operare in modalità di <i>Analisi Spettrale</i> esportando solo le chiavi che sono selettive per la ricerca indicata e con la frequenza ad essa relativa.
	query⁷⁾	Alternativo all'attributo sel consente di indicare direttamente la frase di selezione da effettuare ed applicare come analisi spettrale.

Esempio

Richiediamo l'esportazione in forma compressa .zip del canale nDomanda nel file omonimo.

```
<?xml version="1.0"?>
<cmd c="x8" bits="x4" id="nDomanda" fn="nDomanda.zip"/>
```

La risposta evidenzia nel dettaglio il nome esteso del file (collocato nella directory dei temporanei di eXtraWay).

```
<?xml version="1.0" encoding="iso-8859-1"?>
<rsp ack="1" e="0"><dtl dtype="file" dval="/opt/it-3di/extraway/xw/tmp/hwtemp/nDomanda.zip"/>
</rsp>
```

Esportazione Unità Informative

Il sub-comando per l'esportazione di unità informative(documenti) può assumere 3 diversi valori (modificatori esclusi

bits	Significato
0x00000006	Esportazione di unità informative(documenti) in forma <i>complessa</i>
0x00000007	Esportazione di unità informative in forma <i>semplificata</i>
0x00000008	Esportazione di unità informative in modalità nidificata
0x0000000D	Esportazione di unità informative in forma <i>molto⁸⁾ semplificata</i>
0x0000000E	Replica delle unità informative su sistema di Replica ⁹⁾

Si può richiedere di esportare tutte le unità informative dell'archivio non indicando nulla di particolare ovvero esportare sottoinsiemi noti.

Per esprimere uno di tali sottoinsiemi si possono dichiarare:

Attributi	Significato
sel	Identificativo del file di selezione che contiene i documenti che si desidera esportare
query	Frase di ricerca da applicare per selezionare i record da esportare ¹⁰⁾



Attributi	Significato
num num2	Numero fisico del primo e dell'ultimo documento che si intende esportare, entrambe compresi. L'assenza del primo o del secondo valore comporta che esso assuma il valore estremo in tale direzione (in assenza di num esso vale '1', in assenza di num2 esso vale quanto il totale dei documenti in archivio) N.B.: Si noti che questi due valori vengono presi in esame esclusivamente se l'attributo <code>se1</code> non esiste o non è valorizzato

Come detto tali valori sono alternativi ed il valore assegnato all'attributo `se1` ha la priorità sugli estremi numerici. Non è possibile quindi combinare le due impostazioni, ad esempio, per dichiarare un intervallo di numeri di documenti riferito alla selezione e non all'intero archivio. In tal caso la selezione dev'essere preventivamente ridotta a ciò che si intende esportare e nulla più.

Oltre a questo è anche possibile indicare un ulteriore attributo...

se12	Identificativo del file di selezione che contiene i documenti che non si vuole vengano esportati. Quale che sia il metodo definito per scegliere cosa esportare, i documenti presenti in questa seconda selezione verranno saltati
-------------	--

Oltre a questo si può richiedere di applicare un foglio di stile XSLT ai documenti prima di sottoporli ad esportazione. Ciò, ovviamente, impatta esclusivamente su quanto esportato lasciando invariato il contenuto dell'archivio.

Per scegliere questa funzionalità si possono seguire due metodi:

- Impostare un attributo `fmode` contenente il nome del file XSLT da applicare in fase di esportazione.

ovvero

- Introdurre nel comando XML un elemento `bst` all'interno del quale riportare integralmente il contenuto del file XSLT del quale si richiede l'applicazione.

La modalità complessa prevede che l'intero file ed ogni unità informativa esportata siano incapsulati in elementi di esportazione corredati da una batteria di attributi che indica una serie di informazioni.

Per l'archivio esse sono: Il nome dell'archivio dal quale si è compiuta l'esportazione, l'eventuale nome del file XSLT che è stato applicato in esportazione, il nome di file di selezione che ha regolato i documenti da esportare ovvero il primo ed ultimo numero di documento esportato.

Per il documento viene esportato un valore etichettato come `nrecord` che corrisponde, di fatto, al numero fisico del documento.

Nella modalità semplificata, questi elementi e corrispondenti attributi vengono trasformati in più sintetici commenti XML così da non impattare sulla struttura definitiva del file esportato. Questo è ancor più vero se l'esportazione viene effettuata in **modalità nidificata**, operazione che dovrebbe tentare la ricostruzione di XML complessivo che rappresenta un singolo archivio (o porzione) quando esso ha relazioni gerarchiche tra i documenti.

Quale che sia la modalità di esportazione dei documenti, essi vengono sempre esportati senza la *signature*, ovvero il *check sum* calcolato da eXtraWay per ogni documento, valore necessario a comprendere se esso abbia subito modifiche manuali o meno.

L'esportazione di unità informative prevede una serie di modificatori di seguito descritti:

Modificatore	Significato
0x40000000	Richiede l'esportazione degli allegati dei documenti in esame. L'esportazione avviene secondo due modalità che si legano strettamente ad altri parametri. Se si richiede di esportare l'archivio sotto forma di file .zip, il file prodotto conterrà un unico XML contenente tutti i documenti e gli allegati organizzati secondo le directory che essi effettivamente occupano nell'archivio dal quale si compie l'esportazione. Se non viene richiesto una simile esportazione gli allegati vengono introdotti in un file definito <code>simultar¹¹⁾</code> che potrà essere interpretato da altre procedure di acquisizione.
0x20000000	Richiede che l'esportazione avvenga con compressione gzip. Quale che sia il file che viene richiesto, esso verrà compresso e generato con estensione supplementare '.gz'. N.B.: Questo risultato si ottiene anche esplicitando direttamente l'estensione '.gz' nel nome del file di esportazione di cui all'attributo 'fn'.
0x08000000	Richiede che l'esportazione abbia luogo creando un file zip. Quale che sia il file che viene richiesto, esso verrà compresso e generato con estensione '.zip'. N.B.: Questo risultato si ottiene anche esplicitando direttamente l'estensione '.zip' nel nome del file di esportazione di cui all'attributo 'fn'. N.B.: Questa funzionalità è disponibile dalla versione 21.0.1.133 o superiore e richiede l'aggiornamento delle librerie terze parti in quanto utilizza libzip-1.dll versione 0.8c/ter o superiore
0x04000000	Contrariamente a quanto precedentemente espresso, prevede che l'esportazione degli allegati abbia luogo privando gli stessi del percorso cui appartengono ed esportandoli tutti insieme, senza distribuzione nel file .zip in directory diversificate. Richiede infatti che sia attivo il bit 0x40000000 per l'esportazione degli allegati ed il bit 0x08000000 (o un opportuno nome di file con estensione .zip). N.B.: Questa funzionalità è disponibile dalla versione 21.1.4.44 o superiore
0x02000000	Bit che si applica sempre all'esportazione dei documenti in forma di file .zip. Si oppone al bit precedente (ed è mutualmente esclusivo con esso) e richiede che nello .zip finale, gli allegati vengano espressi con il loro percorso assoluto. ATTENZIONE: Dalla sua introduzione la documentazione di questo bit dichiarava che i codici di allegati venivano convertiti in percorsi relativi all'atto dell'esportazione. Ciò non è mai stato vero ed è stato introdotto un ulteriore bit per questa specifica funzionalità. N.B.: Questa funzionalità è disponibile dalla versione 21.1.4.198 o superiore



Modificatore	Significato
0x01000000	Bit introdotto ma non ancora utilizzato. La sua finalità è quella di consentire al richiedente di esplicitare, al posto di export, il nome del <i>root element</i> del file in cui l'esportazione ha luogo. N.B.: Questa funzionalità è disponibile dalla versione 21.2.0.0 o superiore
0x00800000	Bit che indica che nello .zip che si intende produrre dev'essere presente anche il file di configurazione dell'archivio. Introdotto nell'ambito dello sviluppo di specifiche funzionalità di <i>clonazione</i> archivio. Non ha molto senso che se ne faccia un uso diretto. N.B.: Questa funzionalità è disponibile dalla versione 21.2.0.0 o superiore
0x00400000	Bit che si applica all'esportazione di documenti che citano allegati in forma implicita, ovvero nel formato <numero>.<estensione>. I documenti esportati presenteranno al posto del semplice codice numerico, il percorso relativo ¹²⁾ dell'allegato. Si ricorda in proposito che la directory candidata alla conservazione degli allegati è quella nota come <i>area di parcheggio</i> ed identificabile come <nomearchivio>.file. I percorsi relativi sono quindi da intendersi relativi a tale collocazione. N.B.: Questa funzionalità è disponibile dalla versione 22.2.3.1 e 23.0.2.0 o superiore

Il comando di esportazione sicuramente più utilizzato, in quanto produce un file XML più simile all'originale, è il comando 0x00000007 ovvero l'esportazione in formato semplificato.

Come detto esso può prevedere che si esprima direttamente il foglio di stile XSLT da applicare nel comando stesso. Se ad esempio abbiamo il seguente XSLT...

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format" xmlns:xw="http://www.3di.it/ns/xw-200303121136" >
<xsl:template match="//scheda_A/csm_def">
  <xsl:call-template name="fai_qualcosa"><xsl:with-param name="parametro" select="'para'"
/></xsl:call-template>
</xsl:template>
</xsl:stylesheet>
```

...esso potrà essere indicato sia come nome del file nell'attributo fmode sia direttamente nell'elemento bst così com'è stato espresso.

Note:

L'esportazione è attualmente un comando che può essere richiesto direttamente dalla *Console*. Esso produce un'esportazione in formato semplificato, già filtrata per la selezione corrente. In tal caso, però, indipendentemente dal nome del file che si richiederà, la *Console* epura tale nome da ogni forma di percorso e quindi l'esportazione avverrà *sempre* nella directory dei temporanei di eXtraWay Server.

Esempi Esportazione

Esportazione in modalità semplificata senza allegati con file compresso .zip in modo implicito.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<cmd c="0x8" bits="0x00000007" fn="myexport.zip"/>
```

La stessa cosa di cui all'esempio precedente può essere espressa in modo esplicito alzando il bit opportuno. Esportazione in modalità semplificata senza allegati con file compresso .zip in modo esplicito

```
<?xml version="1.0" encoding="iso-8859-1"?>
<cmd c="0x8" bits="0x08000007" fn="myexport.xml"/>
```

In ambo i casi i file risultante avrà estensione .zip e conterrà un file omonimo con estensione .xml

Esportazione in modalità semplificata con compressione zip ed indicazione della selezione che funge da filtro dell'esportazione. Richiesta degli allegati in forma normale(distribuzione in percorsi nel file .zip)

```
<?xml version="1.0" encoding="iso-8859-1"?>
<cmd c="0x8" bits="0x48000007" sel="3sea09ef721ab.tmp" fn="myexport.xml"/>
```

Esportazione in modalità semplificata con compressione zip ed indicazione dell'intervallo di documenti da esportare. Richiesta degli allegati senza percorso(nel file zip verranno esportati tutti nella radice dello zip stesso).

```
<?xml version="1.0" encoding="iso-8859-1"?>
<cmd c="0x8" bits="0x44000007" num="100000" num2="200000" fn="myexport.zip"/>
```

Esportazione in modalità semplificata di documenti selezionati con conversione degli ID d'allegato in percorsi relativi. Non vengono esportati allegati.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<cmd c="0x8" bits="0x00400007" sel="3se4104582abe17601.tmp" fn="DCW_2008.zip"></cmd>
```



Esportazione MetaDati

Documentazione in fase di realizzazione

Conversione dati CVS per import da WatchDoc

Documentazione in fase di realizzazione

Realizzazione di un sub-archivio

Documentazione in fase di realizzazione

Comando 0x0000000B (Gestione degli Allegati)

I comandi per la gestione degli allegati si ottengono esprimendo un particolare valore nell'attributo *bits*, come da tabella seguente.

Valore	Significato
0x00000001	Richiede di compiere la verifica, totale o parziale, di quanti e quali allegati, se pure da sottoporre al processo di conversione file, non siano stati ancora convertiti. Ovviamente, se la conversione non può aver luogo, il <i>File Conversion Agent</i> dovrebbe modificare comunque il documento per abbattere i valori che indicano che la conversione è ancora pendente.
0x00000002	Richiede che si calcoli l'impronta di un set di allegati.
0x00000004 0x00000008	Funzione speciale realizzata per sanare particolari casi di erronea assegnazione di un allegato a più documenti. La funzione provvede alla duplicazione dell'allegato ed all'assegnazione a ciascun documento di allegati univoci.
0x00000010	Funzione che richiede la lista degli <i>Storages</i> disponibili come da configurazione d'archivio.
0x00000020	Funzione che richiede la migrazione degli allegati da uno <i>Storage</i> ad un'altro.
0x00000080	Enterprise Only Funzione che richiede la riorganizzazione degli allegati di un archivio abolendo la distribuzione <i>next to xml</i> .

Il comando 0x00000020 (Migrazione Allegati)

N.B.: Questo comando esiste solo dalla versione 23.0.2.0 *candidate* ovvero 23.0.3.0.

Il comando per la migrazione degli allegati da uno *Storage* ad uno differente dev'essere completato con ulteriori parametri rappresentati da attributi del comando.

Innanzitutto si deve poter esprimere l'intervallo di documenti su cui operare. Esso può essere espresso come intervallo di numeri fisici di documenti ovvero come selezione secondo il seguente schema.

- Si testa la presenza dell'attributo *sel*. Se esiste ed ha un valore non nullo si assume che esso sia l'*id* della selezione da utilizzare;
- Se la selezione non è stata espressa correttamente si cerca la presenza degli attributi *num* e *num2* atti ad indicare gli estremi numerici entro cui operare.
In assenza di un valore valido per *num* si parte dal primo documento dell'archivio.
In assenza di un valore valido per *num2* si arriva all'ultimo documento dell'archivio.
Ovviamente, se i due valori non vengono espressi, si opera sull'intero archivio.

In secondo luogo si deve indicare lo *Storage* di destinazione. Esso può essere indicato secondo il seguente schema.

- Si verifica la presenza di un attributo *dir*. Se esso esiste ed ha un valore valido si confronta il contenuto con la configurazione dell'archivio. Il valore espresso può essere tanto un percorso assoluto quanto una label (Vds. [Esempi](#)). Se non viene riscontrato alcun valore si segnala l'errore e la procedura non prosegue.
- In assenza di un valore valido per *dir* si verifica la presenza dell'attributo *id*. Esso deve rappresentare un numero in base '0' da interpretarsi come segue:
 - 0 = Storage di primo livello, in pratica le directory ove sono collocati naturalmente gli allegati in prossimità degli XML che li referenziano;
 - 1 = Storage di secondo livello (il primo storage configurato nel file <nomearchivio>.conf.xml);
 - 2 = Storage di terzo livello (il secondo storage configurato nel file <nomearchivio>.conf.xml);
 - 3 ... etc. etc.

E' utile sapere che:

1. Il server può indicare nel proprio log alcune informazioni utili quali:
 1. **"Attachment <nome> in doc <numero> doesn't belong to a valid storage"**: L'allegato non risulta essere collocato in alcuno degli storages configurati nel server. Questo avviene, ad esempio, quando l'allegato è referenziato con un percorso assoluto o relativo. L'allegato viene ignorato;
 2. **"Error <codice> moving attachment <nome> in doc <numero> to <nuovo nome>"**: Si tratta di un vero e proprio errore di copia. Potrebbe dipendere dalla mancanza di spazio sullo storage di destinazione o alla presenza di altro file con pari denominazione;
 3. **"Attachment <nome> in doc <numero>, moved to <nuovo nome>"**: Indica il corretto spostamento dell'allegato;
 4. **"Attachment <nome> in doc <numero> already in required storage"**: Indica che l'allegato si trova già nello storage richiesto. L'allegato rimane immodificato;



5. "Attachment <nome> in doc <numero> can't be found": Non è possibile reperire l'allegato indicato, esso viene ignorato;
2. Vengono inoltre ignorati e mai rimossi dallo storage di livello '0' tutti gli allegati che siano collocati nella vecchia modalità di storage (secondo lo schema <directory nomearchivio>.file<numero>\<numero>.<estensione>), gli allegati originali che siano oggetto "diretto" di indicizzazione¹³⁾ nonché quelli considerati allegati di servizio. Si dicono allegati di servizio tutti quelli che sono il frutto di una trasformazione compiuta tramite il *File Conversion Service*. In questo caso si compie un distinguo.
 1. Gli allegati di tipo testuale prodotti da *FCS* che siano necessari all'indicizzazione non vengono rimossi dal livello '0' perché la loro presenza può essere fondamentale;
 2. Gli allegati di altra natura prodotti da *FCS* che non siano necessari all'indicizzazione possono essere spostati su storage diversi e seguono le sorti degli allegati standard¹⁴⁾.
3. Al termine dell'operazione (Vds. [Esempi](#)) viene tornato un breve elenco che indica quanti allegati siano stati spostati, quanti ignorati e quanti lasciati immodificati in quanto già in posizione corretta.

Esempi Gestione Storages

Supponiamo che un archivio sia stato configurato per usare altri storages oltre a quello principale.

Il suo file <nomearchivio>.conf.xml dovrà contenere una configurazione come la seguente:

```
...
<profile type="attach.idhome" value="d:\xw\db\dcw\docwaydoc\storage1\[Level
1],d:\xw\db\dcw\docwaydoc\storage2\[La casa di
Pippo],d:\xw\db\dcw\docwaydoc\docwaydoc\storage3\[Level 2]"/>
...
```

La configurazione consente anche di indicare percorsi relativi alla collocazione dell'archivio.

La richiesta dell'elenco degli storages disponibili si esprime come segue:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<cmd c="xb" bits="0x00000010"/>
```

Risposta

```
<?xml version="1.0" encoding="iso-8859-1"?>
<rsp ack="1" e="0">
  <storages num="3">
    <storage level="1" dir="d:\xw\db\dcw\docwaydoc\storage1\" label="Level 1"/>
    <storage level="2" dir="d:\xw\db\dcw\docwaydoc\storage2\" label="La casa di Pippo"/>
    <storage level="3" dir="d:\xw\db\dcw\docwaydoc\docwaydoc\storage3\" label="Level 2"/>
  </storages>
</rsp>
```

Si noti che la *label* deriva da quanto indicato espressamente nel file di configurazione d'archivio mentre l'ordine di valutazione, dato dal *level*, corrisponde all'ordine in cui l'elenco è stato alimentato.

Richiesta di migrazione degli allegati.

Richiesta con indicazione della selezione e dell'etichetta desiderata

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<cmd c="xb" bits="0x00000020" sel="3se1203fa78439b01" dir="Level 2"/>
```

Richiesta con indicazione della selezione e del percorso voluto

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<cmd c="xb" bits="0x00000020" sel="3se1203fa78439b01" dir="d:\xw\db\dcw\docwaydoc\storage2"/>
```

Questa richiesta è in tutto equivalente alla precedente in quanto l'*id* del livello indicato conduce allo stesso storage.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<cmd c="xb" bits="0x00000020" sel="3se1203fa78439b01" id="2"/>
```

Richiesta con indicazione della selezione e dello storage richiesto.

Questa richiesta riporta alla collocazione originaria tutti gli allegati dei documenti selezionati che fossero collocati in storages differenti.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<cmd c="xb" bits="0x00000020" sel="3se1203fa78439b01" id="0"/>
```



A tutti i precedenti comandi la risposta avrà il seguente aspetto.

```
<rsp ack="1" e="0">
  <dtl dtype="moved" dval="81"/>
  <dtl dtype="unchanged" dval="6"/>
  <dtl dtype="skipped" dval="18"/>
</rsp>
```

Il comando 0x00000080 (Riorganizzazione Allegati)

Prevede un singolo parametro `id` che può assumere i valori:

Valore	Significato
1	Effettua la riorganizzazione dei soli allegati collocati in prossimità dei file XML contenenti i record.
2	Effettua la riorganizzazione dei soli allegati collocati nella directory <code>nomearchivio.file</code> .
3	Effettua entrambe le precedenti operazioni, partendo dalla cartella <code>nomearchivio.file</code> e, di seguito, con gli allegati <i>near to xml</i> .

Se il parametro non viene indicato il valore default è 1.

Le versioni antecedenti la 1.0.7, se il valore impostato è pari a 3, eseguono prima il comando 2 e poi il comando 1. Purtroppo questo comportamento non è del tutto corretto quindi, per versioni di server antecedenti si suggerisce di inviare separatamente i due comandi, prima con valore 1 e poi con valore 2 mentre quest'accorgimento non è necessario per le versioni 1.0.7 e superiori.

Esempi Riorganizzazione Allegati Enterprise

Richiesta di riorganizzazione di tutti gli allegati .

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<cmd c="xb" bits="0x00000080" id="1"/>
```

La risposta avrà il seguente aspetto.

```
<rsp ack="1" e="0">
  <dtl dtype="total" dval="103683"/>
  <dtl dtype="moved" dval="103677"/>
  <dtl dtype="skipped" dval="3"/>
  <dtl dtype="rmDirs" dval="1127"/>
  <dtl dtype="faults" dval="3"/>
  <dtl dtype="fault"
dval="c:/xwee/db/dcw/xdocwaydoc/xdocwaydoc/2001/3dinbol/fascicoli/00_00/2001-00_00.00110.file/0053
22.txt"/>
  <dtl dtype="fault"
dval="c:/xwee/db/dcw/xdocwaydoc/xdocwaydoc/2001/3dinbol/fascicoli/00_00/2001-00_00.00110.file/0055
68.txt"/>
  <dtl dtype="fault"
dval="c:/xwee/db/dcw/xdocwaydoc/xdocwaydoc/2004/3dinbol/fascicoli/00_00/2004-00_00.00022.file/0000
00/000089.tif"/>
</rsp>
```

Comando 0x00000013 (Rilocalazione unità informative)

Questo comando consente la ridistribuzione delle unità informative nei files XML dell'archivio quando vengono cambiate le regole di `file_location` ovvero quando, pur avendo impostato una regola variabile, non è stato impostato il parametro `'move_always="true"'`:

Si può indicare l'insieme dei documenti si cui operare introducendo l'attributo `se1` valorizzato con l'ID della selezione che li racchiude ovvero indicare l'intervallo di documenti fisici introducendo gli attributi `num` e `num2`. In assenza di entrambe le dichiarazioni, l'operazione di rilocalazione avverrà per tutte le unità informative.

Ci sono inoltre dei modificatori che possono essere impostati nell'attributo `bits`

Valore	Singificato
0x01	Richiede, in presenza di una serie di DTD che sono evolute nel tempo, che l'unità informativa venga condotta alla forma prevista dalla DTD più recente con l'ausilio del XSLT correlato.
0x02	Richiede che non vengano rettificati gli indici dei documenti. A seguito dell'operazione di rilocalazione sarà quindi necessario rifare gli indici dell'archivio. Va detto, infatti, che per ogni singolo documento non vengono modificati gli indici inerenti il suo contenuto ma quelli riferiti alla componente UD, ovvero informazioni quali il nome del file XML nel quale l'unità informativa si trova, il time stamp del file ed altri. Tali operazioni vengono fatte singolarmente, un documento alla volta, con un dispendio di tempo considerevole. L'impostazione di questo <i>bit</i> evita il ritocco indici che però andrà fatto a posteriori.



Avremo quindi...

```
<cmd c="0x13" bits="0"/>
```

...per una rilocalizzazione completa dell'archivio...

```
<cmd c="0x13" bits="0" sel="3sel12934ad4901"/>
```

...per la rilocalizzazione di un ben preciso set di documenti ottenuto tramite una selezione ...

```
<cmd c="0x13" bits="0" num="1" num2="10000"/>
```

...per la rilocalizzazione di un set di documenti descritto da un range di numeri fisici di record...

```
<cmd c="0x13" bits="2"/>
```

...per una rilocalizzazione completa dell'archivio senza indici cui far seguire un *reindex*.

Comando 0x00000016 (Estensione XSLT)

Questo comando *Estensione XSLT* consente di associare ad ogni documento della banca dati (distinguendoli per singole *primary nodes*) un'elaborazione del contenuto del documento stesso sulla base di un foglio di stile XSLT. Il risultato può quindi essere utilizzato per di versi scopi come:

- Alimentare vocabolari con il contenuto di aree non altrimenti identificabili/componibili
- Sfruttare questi elaborati per la composizione dei titoli/cache dei documenti
- Ordinamento dei documenti sulla base di simili elaborati

Per ottenere questo risultato bisogna innanzitutto accoppiare ad ogni unità informativa un particolare file .xslt

```
<primary_node ud_name="doc" ud_container="list_of_doc" ud_xslt="change.xslt">
  ...
</primary_node>
```

Ovviamente il file dev'essere collocato in una posizione su disco corrispondente al percorso assoluto o relativo che viene indicato. Se il percorso è relativo, come avviene normalmente, la directory di riferimento è quella ove si trova l'archivio stesso. Nel nostro esempio, il file *change.xslt* si troverà nella stessa directory del file *<nomearchivio>.conf.xml* dal quale viene tratto l'esempio.

Avremo quindi...

c	0x16 (22)
bits	Può assumere uno dei seguenti valori 1 ⇒ Indica che si richiede di avere in risposta per i documenti indicati il contenuto dell'estensione calcolata per il documento 2 ⇒ Indica che si richiede il ricalcolo dell'estensione XSLT dei documenti indicati (tipicamente perché sono state apportate modifiche al foglio di stile XSLT associato all'archivio o al primary node). L'operazione viene effettuata in modalità "non bloccante", rettifica la componente XSLT ma non agisce sugli indici che quindi al termine dell'operazione non possono essere considerati affidabili. 3 ⇒ Come il comando 2 ma con attività da svolgere in modalità <i>batch</i> con un blocco sull'archivio che, per il tempo del comando, risulta non disponibile ad operazioni modificanti. Diversamente dal caso 2, però, gli indici vengono rifatti al termine delle operazioni. 4 ⇒ Simile al comando 1 ma finalizzato esclusivamente all'indagine del rapporto tra numeri logici di una selezione e numeri fisici A questi comandi (solo al 'bit' 3) può essere associato anche il bit 0x80000000 ⇒ Richiede che nel ricalcolare le estensioni XSLT dei documenti non si compia la rettifica degli indici (che risulta non necessaria o che verrà effettuata massivamente in seguito)

In presenza di comando '1' o '4' l'elemento **bst** deve contenere una serie di elementi **item**, uno per ogni documento del quale si richiede il ritorno dell'elaborato XSLT. Il documento richiesto sarà riconoscibile per mezzo dell'attributo **d**. E' ammessa l'assenza dell'elemento **bst** e del suo contenuto se viene esplicitato l'attributo "sel" mostrato in seguito. In tal caso l'operazione viene compiuta a tappeto su tutti i documenti della selezione indicata.

Inoltre, per identificare l'insieme di documenti su cui operare, possono essere indicate i seguenti attributi

sel	Nome del file di selezione contenente i documenti su cui operare. Per il comando '1' e '4' indica che il numero documento richiesto deve considerarsi come indice nella selezione e non come numero assoluto. In assenza dell'elemento 'bst' si assumono tutti i documenti referenziati dalla selezione.
num num2	utilizzabile solo per i comandi '2' e '3'. Se non è stato indicato sel si interpretano questi due valori che non devono essere obbligatoriamente presenti. Essi esprimono gli estremi, ovviamente compresi, del <i>range</i> di documenti su cui operare. In assenza di un valore valido per num si assume che esso sia pari ad 1 per dare inizio alla lavorazione dal principio dell'archivio ed in assenza di un valore valido per num2 si assume che esso sia pari al numero totale dei documenti presenti in archivio



Esempi

Richiesta di tornare al chiamante l'esito dell'estensione XSLT per i primi 5 documenti di una selezione

```
<cmd c="x16" bits="1" sel="3se440f508cb838a01">
  <bst>
    <item d="1"/> <!-- Ovviamente questo non è il primo documento dell'archivio bensì della
selezione -->
    <item d="2"/>
    <item d="3"/>
    <item d="4"/>
    <item d="5"/>
  </bst>
</cmd>
```

L'esito avrà l'aspetto seguente

```
<rsp ack="1" e="0">
  <bst>
    <item d="1" dln="123"> <!-- Viene tornato anche il numero fisico del documento -->
      <!-- Frammento XML corrispondente all'esito della lavorazione XSLT -->
    </item>
    <item d="2" dln="234">
      ...
    </item>
    <item d="3" dln="235">
      ...
    </item>
    <item d="4" dln="236">
      ...
    </item>
    <item d="5" dln="367">
      ...
    </item>
  </bst>
</rsp>
```

Richiesta ricostruzione *on-line* di tutti gli elaborati XSLT dei documenti in archivio senza intervento sugli indici

```
<cmd c="x16" bits="2"/>
```

Richiesta ricostruzione *batch* dell'elaborato XSLT per i primi 1000 documenti dell'archivio senza ricostruzione indici grazie al flag aggiuntivo.

```
<cmd c="x16" bits="0x80000003" num2="1000"/>
```

Richiesta traduzione dei numeri logici di selezione in numeri fisici

```
<cmd c="x16" bits="4" sel="3se440f508cb838a01"/>
```

Il risultato avrà un aspetti simile al seguente:

```
<rsp ack="1" e="0">
  <bst>
    <item d="1" dln="54948"/>
    <item d="2" dln="55169"/>
    <item d="3" dln="57247"/>
    <item d="4" dln="58002"/>
    <item d="5" dln="58478"/>
    <item d="6" dln="63914"/>
    <item d="7" dln="100592"/>
    <item d="8" dln="101196"/>
    <item d="9" dln="104929"/>
    <item d="10" dln="131387"/>
    <item d="11" dln="188857"/>
    <item d="12" dln="198055"/>
    <item d="13" dln="199931"/>
  </bst>
</rsp>
```

```
<item d="14" dln="200636"/>
<item d="15" dln="206653"/>
</bst>
</rsp>
```

Comando 0x00000017 (Search & Replace)

Il comando di *Search & Replace* consente di identificare un valore contenuto in un componente dell'unità informativa XML (che si tratti di attributo o elemento) e sostituirlo con altro valore. Il confronto può essere effettuato sia in forma *case sensitive* che in forma *case insensitive* e la sostituzione può interessare tanto l'intero valore del componente quanto una sequenza di caratteri al suo interno eventualmente ripetuta più volte.

Avremo quindi...

c	0x17 (23)
bits	Si ottiene per combinazione a bit dei seguenti valori 1 ⇒ Indica che il confronto deve essere fatto in modalità <i>case insensitive</i> mentre il comportamento di default è <i>case sensitive</i> 2 ⇒ Indica che il confronto verrà effettuato su porzioni del testo del componente indicato ¹⁵⁾ mentre il comportamento di default è che il test avvenga per l'intero contenuto del componente Avremo quindi le seguenti combinazioni possibili:
0	Test Case Sensitive sull'intero contenuto del componente
1	Test Case Insensitive sull'intero contenuto del componente
2	Test Case Sensitive su porzioni del testo del componente
3	Test Case Insensitive su porzioni del testo del componente
id	Percorso XML, assoluto ¹⁶⁾ o parziale, che identifica il componente (elemento o attributo) sul quale compiere il confronto
key	Valore di confronto da utilizzare in modo <i>sensitive</i> o <i>insensitive</i> sul componente o una sua parte secondo quanto indicato dall'attributo bits
key2	valore col quale compiere la sostituzione. Esso verrà utilizzato <i>as is</i> indipendentemente dal fatto che il test sia stato effettuato in modalità <i>case sensitive</i> o meno

Inoltre, per identificare l'insieme di documenti su cui operare, possono essere indicate i seguenti attributi

sel	Nome del file di selezione contenente i documenti su cui compiere confronti e sostituzioni
num	Se non è stato indicato sel si interpretano questi due valori che non devono essere obbligatoriamente presenti. Essi esprimono gli estremi, ovviamente compresi, del <i>range</i> di documenti su cui operare. In assenza di un valore valido per num2 num si assume che esso sia pari ad 1 per dare inizio alla lavorazione dal principio dell'archivio ed in assenza di un valore valido per num2 si assume che esso sia pari al numero totale dei documenti presenti in archivio

Il comando prevede uno stato di avanzamento che può essere sfruttato dal chiamante.

Nota Bene	L'implementazione al 15/01/2009 non prevede (ancora) di stabilire se l'applicazione del foglio di stile XSLT abbia effettivamente apportato trasformazioni o meno. Questo comporta il fatto che i documenti vengono risalvati comunque indipendentemente dal fatto che siano realmente cambiati o meno
Attenzione	Per tutelarsi dal rischio che possa essere espresso un foglio di stile che non sia corretto e che, quindi, una volta applicato ai documenti causi una perdita di dati, esso è in sostanza <i>hard coded</i> nel server e si adegua al fatto che sia stato richiesto di agire su un elemento o su di un attributo. La condizione perché il comportamento del foglio di stile sia corretto è però che i documenti XML contenuti in Banca Dati non presentino elementi " <u>mixed</u> ", ovvero elementi che possono contenere tanto del testo quanto altri elementi. Se utilizzato su elementi di questo tipo, il comando danneggia il documento producendo un documento che può essere incompleto ovvero presentare aree di test replicate in punti non coerenti

Il comando torna un acknowledge positivo o negativo ed implementa uno *stato d'avanzamento*.

Informazioni fornite

L'esito del comando produce un output xml composto da una serie di elementi `dt17)` aventi due attributi, `dtype` e `dval`. Di seguito una panoramica dei valori che si possono ottenere in risposta.

dtype	dval
total	Numero totale dei documenti assoggettati a <i>Search & Replace</i>
done	Numero dei documenti sui quali si è compiuto effettivamente un <i>Search & Replace</i> con successo
skipped	Numero dei documenti sui quali non si è compiuto un <i>Search & Replace</i> con successo ¹⁸⁾
erased	Numero dei documenti su cui si sarebbe dovuto procedere che sono risultati cancellati
nested	Numero dei documenti su cui si sarebbe dovuto procedere che sono risultati nidificati
notOwned	Numero dei documenti su cui si sarebbe dovuto procedere ma dei quali il server non ha la proprietà ¹⁹⁾
notWellformed	Numero dei documenti su cui si sarebbe dovuto procedere che sono risultati malformati
xsltFailure	Numero dei documenti su cui si sarebbe dovuto procedere che hanno presentato un errore nell'elaborazione dell'XSLT di <i>Search & Replace</i>
lockFailure	Numero dei documenti su cui si sarebbe dovuto procedere che sono risultati bloccati da terzi



dtype	dval
saveFailure	Numero dei documenti su cui si sarebbe dovuto procedere la cui lavorazione era del tutto corretta ma che hanno causato un errore in fase di salvataggio
docFailed-Numero Documento	Codice d'errore rilevato durante il tentativo di salvataggio del documento. Si presenta un simile codice per ogni documento che non è stato possibile salvare. Essi sono i documenti totalizzati alle voci lockFailure e saveFailure.

Esempi

Richiesta di sostituzione sull'intero archivio dell'intero valore di un attributo in modo *case insensitive*

```
<cmd c="x17" bits="0"
      id="/iter/@d_abbreviata"
      key="regio decr. leg."
      key2="regio D.L."/>
```

Richiesta di sostituzione su alcuni documenti selezionati in un archivio un valore contenuto anche più volte in un elemento in modo *case insensitive*

```
<cmd c="x17" bits="2"
      id="/iter/rtf/contenitore/testo"
      key="M.C.T.C."
      key2="Motorizzazione Civile"
      sel="3se4415f8502b0ea01">
</cmd>
```

Comando 0x00000018 (Index Drop, Build & Rebuild)

Il comando in esame nasce per occuparsi di attività da svolgere su singoli indici, in particolare la loro rimozione dall'attuale vocabolario (Drop) ovvero la loro generazione, anche ex-novo (Build-Rebuild). Esso è disponibile dalla versione **23.0.0.1 e successive**.

In un secondo momento è stato ampliato per occuparsi, di fatto, di tutti gli indici di un archivio in modo cumulativo.

Si ricorda che in questo caso, parlando di *Indice* ci si riferisce a quello che solitamente è noto anche come *Vocabolario* ovvero la collezione di tutte le chiavi che si riferiscono ad un singolo canale di ricerca²⁰⁾ e le relative catene di riferimenti che consentono la selezione dei documenti.

Analogamente, di seguito di indicherà con *Drop* l'abbattimento di indice esistente, con *Rebuild* la ricostruzione dell'indice esistente e precedentemente svuotato e con *Build* la generazione di un indice in precedenza mai esistito, quindi un vocabolario del tutto nuovo.

Per quanto concerne il rifacimento complessivo di tutti gli indici siamo soliti riferirci al concetto di *Catalogazione o Indicizzazione Off Line* che in questo caso viene definita *Fast*.

Avremo quindi...

c	0x18 (24)
bits	Si ottiene per combinazione a bit dei seguenti valori 0x00000001 ⇒ Indica che si richiede di compiere il <i>Drop</i> della chiave indicata 0x00000002 ⇒ Indica che si richiede <i>Build o Rebuild</i> della chiave indicata 0x00000004 ⇒ Indica che si richiede il <i>Drop & Rebuild</i> di tutte le chiavi e non va combinato con altri bits. Avremo quindi le seguenti combinazioni possibili:
1	<i>Drop</i> della chiave indicata
2	<i>Build</i> di una nuova chiave sino ad ora inesistente o <i>Rebuild</i> di una chiave esistente ²¹⁾
3	<i>Drop & Rebuild</i> della chiave indicata
4	<i>Drop & Rebuild</i> di tutti gli indici dell'archivio, in sostanza è la modalità <i>Indicizzazione Off Line Fast</i>
0x80000000	<i>Reset Maintenance</i> . Bit supplementare, da combinare con i precedenti, che indica di abbattere lo stato di <i>Archivio in Manutenzione</i> per poter procedere con l'indicizzazione. Lo stato di <i>Archivio in Manutenzione</i> viene automaticamente impostato ogni volta che si avvia un processo di manutenzione degli indici non bloccante, come sono appunto quelli del comando in esame. Se l'operazione viene interrotta brutalmente, lo stato rimane attivo e va necessariamente abbattuto prima di procedere con un'operazione successiva.

I suddetti valori di **bits** possono essere combinati col valore 0x80000000 per effettuare l'operazione di indicizzazione/drop/reindicizzazione di uno o più indici ignorando l'eventuale presenza dello stato di "Maintenance in corso". Questo bit è da utilizzarsi quando una precedente operazione di manutenzione straordinaria è stata troncata maldestramente e lo stato di manutenzione permane. In alternativa, ovviamente, si può far ricorso al [comando amministrativo 0x5](#).

Oltre al comando "c" ed ai parametri indicati con l'attributo "bits" è necessario identificare su quale *Vocabolario* si intende operare. Per farlo si possono usare due diversi metodi basati sugli attributi "id" e "key".

Ovviamente nel caso di ricostruzione integrale degli indici questo parametro viene ignorato.

I casi più comuni e meglio "coperti" da questa funzionalità sono quelli in cui si compia il *Drop & Rebuild* di una singola chiave già



esistente senza apportare significativi interventi di modifica al conf che la riguarda, ovvero si provveda alla ricostruzione integrale degli indici dell'archivio. Questo in quanto il calcolo delle posizioni dei termini internamente al documento potrebbe causare condizioni non allineate se si fa, ad esempio, l'indice di una chiave che precedentemente era *skippata* o più semplicemente si converte in *multi valore* una chiave che era precedentemente *mono valore*²²⁾. In tal caso, pur garantendo in linea di principio le adiacenze²³⁾ ci sono diversi aspetti tra cui il reverse dei termini che non può essere effettuato correttamente.

Alla luce di questa considerazione, la lavorazione di una singola chiave ha senso quando solo essa risulti danneggiata e debba essere ricostruita minimizzando gli impatti su un archivio in esercizio e/o quando ci si trovi in stato di sviluppo di un'applicazione e si debbano rapidamente cambiare le carte in tavola per quanto concerne una chiave precedentemente non indicizzata o indicizzata con criteri differenti, a patto d'accettare che l'archivio necessiti comunque di un'attività di reindicizzazione integrale.

Si ricorda in fine che attività che comportino un *Drop* di una o più chiavi di un archivio possono causare una crescita della dimensione degli indici apparentemente non giustificata. Essa può essere sanata con un semplice compattamento degli indici in quanto l'operazione di *drop annulla* gli indici che vanno buttati senza rimuoverli fisicamente per ridurre il tempo dell'operazione allo stretto necessario. L'operazione di *Drop*, quando invocata singolarmente e non unitamente ad una richiesta di *Rebuild*, provvede ad impostare la chiave del quale si richiede il *Drop* in modo che essa risulti da non indicizzare nel file `<nomearchivio>.conf.xml` come nell'esempio che segue.

```
...
<key name="..." key_style="skip"/>
...
```

Dichiarazione per mezzo dell'attributo "id"

La dichiarazione della chiave per mezzo dell'attributo "id" comporta che in esso debba essere presente il percorso XML completo della chiave per la quale si richiede di compiere il *Drop* o il *Build-Rebuild*. Tale percorso può essere corredato dal prefisso che ne indica il macro-canale eXtraWay²⁴⁾ ma se esso viene omesso si identifica d'ufficio il macro-canale 'XML'.

Questa dichiarazione si presta sia per la funzionalità di *Drop* che per quella di *Build* o *Rebuild*.

Questa dichiarazione, inoltre, è da considerarsi prioritaria nei confronti della successiva. Se questa è presente, l'attributo "key" viene ignorato.

Dichiarazione per mezzo dell'attributo "key"

La dichiarazione della chiave per mezzo dell'attributo "key" comporta che in esso debba essere presente il codice di chiave di cui fare *Drop* o *Rebuild*. Mancando la definizione del macro-canale esso è, per definizione, 'XML'.

Il valore della chiave può essere espresso sia in forma decimale che esadecimale (prefissandolo con un '0x' o una 'x') e può essere espresso indipendentemente sia con il suo valore "secco" sia con il suo valore di chiave già mascherato col valore 0x8080²⁵⁾.

In questo modo, per una chiave avente codice '60' i seguenti valori sono tutti leciti ed equivalenti.

	Decimale	Esadecimale
Senza Maschera	60	0x3c
Con Maschera	32956	0x80BC

Mancando la dichiarazione del percorso di chiave è evidente che questa dichiarazione non sia applicabile al caso del *Build* ma solo al *Rebuild* in quanto la chiave non potrebbe essere creata ed anzi deve per forza riferirsi a chiave esistente.

Casi particolari

Se pure questo set di comandi sorge nell'intenzione di sanare singoli indici applicativi che risultassero danneggiati e/o costruire indici nuovi o rigenerare indici preesistenti secondo nuovi criteri, esso può essere utilizzato anche per le chiavi di servizio, per meglio intenderci quelle appartenenti al macro-canale UD.

In questo caso, viene ammesso il *Drop* ed il *Rebuild* mentre è assolutamente proibito il *Build* in quanto tali chiavi sono soggette a regole proprie.

Va inoltre ricordato che questo set di comandi assolve al compito di rielaborare i vocabolari, non di compiere interventi sul file di configurazione dell'archivio, attività per la quale esistono comandi appositi.

In altri termini, se si desidera ricostruire un indice in quanto si variano le sue caratteristiche - passando ad esempio da un indice di tipo `multi` ad un indice di tipo `double` - si assume che l'intervento sul file di configurazione venga compiuto manualmente o sfruttando uno dei suddetti comandi e che tale intervento preceda il *Drop & Rebuild*²⁶⁾.

Esempi Comando

```
<cmd c="x18" bits="1" id="/doc/nrecord"/>
```

Indica l'intenzione di *Droppare* la chiave XML, /doc/nrecord in quanto l'omissione del prefisso XML, rappresenta il default.

```
<cmd c="x18" bits="2" id="UD,/xw/MyData"/>
```

Richiede la creazione di un canale UD, /xw/MyData tra le chiavi di servizio. Secondo quando indicato precedentemente, questo comando è destinato a fallire in quanto non si possono creare nuove chiavi nel macro-canale di servizio UD.

```
<cmd c="x18" bits="3" key="x80Bf"/>
```



Richiede il *Drop & Rebuild* della chiave codificata '0x80bf', vale a dire 'x03f', appartenente al macro-canale 'XML'.

```
<cmd c="x18" bits="3" id="XML,/doc/oggetto" key="x80aa"/>
```

Richiede il *Drop & Rebuild* della chiave XML,/doc/oggetto con macro-canale XML esplicitato.

La dichiarazione del codice di chiave - 0x80aa, ovvero 0x2A, ovvero 42 - viene del tutto ignorata. Non viene compiuto alcun test sul fatto che la seconda dichiarazione combaci o meno con la prima né l'operazione viene in qualche modo inibita a causa di una potenziale ambiguità. La dichiarazione per mezzo dell'attributo "key" viene semplicemente ignorata.

```
<cmd c="x18" bits="4"/>
```

Richiede il *Drop & Rebuild* di tutte le chiavi dell'archivio.

Nell'esempio non vengono riportati gli attributi id e key in quanto, anche qualora presenti, il loro contenuto verrebbe del tutto ignorato.

Esempi Risposta

La risposta fornita dal server, secondo gli standard indicati in testa a questo documento, può assumere uno di questi aspetti:

In risposta al rifacimento di una chiave viene indicato il suo codice di chiave ed il suo percorso.

```
<rsp ack="1" e="0">
  <dtl d_type="key" d_val="x8081"/>
  <dtl d_type="fn" d_val="XML"/>
  <dtl d_type="id" d_val="/doc/oggetto"/>
</rsp>
```

In risposta al rifacimento di tutte le chiavi dell'archivio viene indicato il numero complessivo di documenti elaborati.

```
<rsp ack="1" e="0">
  <dtl d_type="NDoc" d_val="1924382"/>
</rsp>
```

Comando verso Plugin

Con l'introduzione del concetto di Plug-In ci si è posti nella condizione di poter estendere ad un ampio grado di libertà le funzionalità del server.

Evocare direttamente un comando presente in un Plug-In del server è una sorta d'estensione dei comandi XML appena esposti ma diversamente da essi affida ad una più versatile verbosità le sorti del comando e non ad una codifica stringente.

La conformazione del comando è tuttavia molto simile. Ciascun comando è rappresentato da un elemento cmd che deve contenere un attributo c il cui valore DEVE essere 0.

Questo consente al server di distinguere un comando XML standard dai comandi destinati ai Plug-In.

Mentre sparisce l'attributo 'bits', appare un attributo func il cui scopo è presto detto.

Esso si compone di due parti, separate da un punto (.). La prima parte rappresenta il nome della *Classe* da evocare. Tra tutti i Plug-In disponibili il server cercherà, seguendo un criterio d'ordine di valutazione, quello che espone tale *Classe*. A destra del punto, invece, sarà presente il nome del metodo della classe indicata da evocare con una chiamata standard, così come standard sarà la risposta ottenuta. Il tutto viene poi impacchettato nuovamente nel messaggio di risposta ed inviato al richiedente.

Di seguito un esempio di comando per il Plug-In XTree ed il comando treeList

```
<?xml version="1.0"?>
<cmd c="0" func="XTree.treeList" arc="rubricaIBC"/>
```

Si noti che tutti i Plug-In che prevedono un input in forma XML ed un output nella stessa forma, consentono la presenza di diversi parametri nel comando.

Ciascun parametro può essere espresso tanto con un'elemento quanto con un attributo, ed eccezione di c e func che devono essere attributi di cmd. In questo modo si possono anche inviare batterie di parametri per mezzo di elementi ripetibili.

In pratica il seguente comando ha una forma differente ma una sostanza equivalente a quella del precedente.

```
<?xml version="1.0"?>
<cmd c="0" func="XTree.treeList">
  <arc>rubricaIBC</arc>
</cmd>
```

Si presti attenzione al *Case* del nome della classe e del metodo da evocare che dev'essere *sensitive* rispetto a quanto esposto dal Plug-In.



I Comandi "Old Style"

I comandi di vecchio tipo ancora supportati dal server eXtraWay verranno man mano sostituiti da comandi XML ma, sino ad allora e per garantire compatibilità col passato, essi permangono.

Non sarebbe ragionevole né coerentemente fattibile dare evidenza di tali comandi ma, almeno per alcuni di essi, è utile conoscere il variare dei comportamenti al variare dell'utilizzo del comando stesso.

Di seguito alcuni dettagli per questi comandi.

Comando di creazione archivio

Usato tanto per creare un archivio quanto per resettare il contenuto del suo catalogo²⁷⁾ e dei suoi indici, ma non dei dati, fa sì che un nuovo archivio eXtraWay venga realizzato presso il *File System* ovvero che un archivio esistente venga riportato alla sua condizione originaria.

Creare un archivio si può fare ad esempio partendo esclusivamente dal file *nomearchivio.conf.xml*, nel quale si siano già disegnate le caratteristiche delle unità informative e delle chiavi delle stesse, ovvero in totale assenza di un simile file, operazione che comporta la realizzazione di un file di configurazione d'archivio "vuoto" da utilizzare come template per la compilazione definitiva.

Il comando prevede diversi parametri, tesi a garantire che venga registrato l'alias associato all'archivio, alla salvaguardia dei seriali che risultassero già presenti ed altre attività. In particolare esso ha subito un'aumento non indifferente a partire dal [server '23'](#).

Il comando²⁸⁾ prevede i seguenti parametri:

Parametro	Codice	Descrizione
REAL_SAVE	0x01	Obsoleto, non più utilizzato
NO_XML	0x02	Si indica che l'archivio risultante non deve salvare in XML. La modalità di salvataggio dei dati utilizzata sarà, per definizione, un <i>BuddyFile</i> non compresso in cui verranno collocati i frammenti XML.
BLIND_MODE	0x04	Indica che si intende generare un archivio cieco, ovvero pensato per essere privo di configurazione ed "auto configurato"
SET_ALIAS	0x08	Imposta il <i>nome logico</i> dell'archivio nel file di configurazione del programma
SAVE_RESET	0x10	Richiede che si compia reset di un archivio esistente e non la creazione di uno nuovo
SAVE_SERIALS	0x20	Richiede che si mantengano i seriali del file <i>nomearchivio.stat.xml</i> quando si compie il reset di un archivio. Questo flag, in pratica, è ora sempre attivo d'ufficio, ad eccezione del caso in cui sia impostato il bit <i>DROP_DATA</i> .
DROP_DATA	0x40	Unitamente alla funzionalità <i>SAVE_RESET</i> indica che oltre a svuotare l'archivio della sua mappa ed indici, si vogliono rimuovere anche tutti i dati

Oltre a questi valori, il comando prevede due stringhe. Una rappresenta il nome logico dell'archivio, quello al quale si dovrà accoppiare nel file *xw.ini* il nome fisico, ed appunto il nome fisico dell'archivio. Essi devono avere una corretta corrispondenza se già presenti nel file *xw.ini* pena la nullità del comando.

Qualora il comando non possa essere espresso dichiarando i bits indicati sopra, gli stessi possono essere impostati aggiungendo al nome fisico dell'archivio le seguenti etichette.

Etichetta	Parametro Corrispondente
<noxml>	NO_XML
<blind>	BLIND_MODE
<reset>	SAVE_RESET
<serials>	SAVE_SERIALS
<drop>	DROP_DATA
<lucene>	NO_XML BLIND_MODE

In particolare, il concetto di archivio "Lucene" verrà chiarito meglio in [apposita documentazione](#).

¹⁾

se pure incapsulati in un generico comando di vecchio ordinamento per non stravolgere il protocollo di comunicazione *Client-Server* esistente

²⁾

Il sistema ottale non viene preso in esame

³⁾

Questa modalità è molto poco sfruttata in quanto le applicazioni tendono ad ignorarla e considerare il comando come riuscito senza particolari accorgimenti

⁴⁾

Maschera 0xFFFF0000

⁵⁾

In caso di *file_locaton rule* di tipo *single*.

⁶⁾

ovviamente il nome del file esportato verrà sempre e comunque tornato al chiamante nel XML di risposta

⁷⁾

Disponibile a partire dalla versione 25.9.3-SE e 1.1.0-EE



8)

Corrisponde alla semplificata ma esclude dall'output anche i commenti XML più basilari.
Presente dalla versione '24' del Server.

9) , 10)

Disponibile su SE 25.9.3 e EE 1.1.0.

11)

formato proprietario non documentato

12)

se possibile, altrimenti assoluto

13)

Come ad esempio i files .txt allegati direttamente ad un documento

14)

Per essi, in futuro, si potrebbe prevedere un'opzione per mantenerli al livello '0'.

15)

ed in tal caso la sostituzione avverrà più volte in ciascun punto in cui tale valore è stato identificato

16)

In caso di percorso assoluto ci si rifà, come punto d'origine, all'elemento che identifica l'unità informativa

17)

Detail

18)

Sommato a done indica il totale dei documenti elaborati

19)

E quindi non vi può apportare correzioni

20)

Ovvero ad un vocabolario comune a più canali

21)

Era sorto come comando singolo ma poi è stato unificato con la rebuild. Trattandosi di un sistema a Bits, vale quindi sia il 2 che il 3 per ricostruire una chiave.

22)

O vice versa

23)

Sui testi, mentre le adiacenze intese come presenza nello stesso bacino di pescaggio, ovvero nello stesso elemento dichiarato *instance* potrebbe non essere coerente

24)

Vale a dire UD, XML o XSL

25)

Notazione interna del server

26)

Di fatto, in caso si compiano separatamente le due operazioni, è sufficiente che l'intervento venga compiuto prima del rebuild

27)

Ovvero della mappa

28)

Il cui codice interno è 0x68