

**ExtGASoai - OAI-PMH gateway per ExtGAS:
organizzazione logica del software**

giugno 2007

Giuseppe A. Romano
Istituto di Scienza e Tecnologie dell'Informazione,
e.mail giuseppe.romano@isti.cnr.it

Indice

| | | |
|---|-----|----|
| O.Premessa | pag | 4 |
| 1.Il software | pag | 5 |
| 1.0 I driver | pag | 6 |
| 1.0.0 ExtGASoai.java | pag | 6 |
| 1.0.1 ExtGASoai_v_Id.java | pag | 8 |
| 1.0.2 ExtGASoai_v_LI.java | pag | 9 |
| 1.0.3 ExtGASoai_v_LS.java | pag | 10 |
| 1.0.4 ExtGASoai_v_LMF.java | pag | 12 |
| 1.0.5 ExtGASoai_v_LR.java | pag | 14 |
| 1.0.6 ExtGASoai_v_GR.java | pag | 15 |
| 1.1. I servizi | pag | 16 |
| 1.1.0 Il modello di selezione e accesso ai dati | pag | 16 |
| 1.1.0.1 L'organizzazione della lista dei risultati: | pag | 18 |
| 1.1.0.2 La struttura dell'OAI-PMH identifier | pag | 18 |
| 1.1.0.3 Il modello di rappresentazione delle date richiesto dal gateway | pag | 19 |
| 1.1.0.4 Il modello di restituzione di Identifiers/Records | pag | 19 |
| 1.1.1 Il restart dell'accesso ai dati | pag | 20 |
| 1.1.2 Il software che gestisce i servizi | pag | 23 |
| 1.1.2.0 ExtGASoaiIsisQuery.java | pag | 25 |
| 1.1.2.1 ExtGASoaiWriteResults.java | pag | 27 |
| 1.2. Le utility | pag | 31 |
| 1.2.0 ExtGASoai.Util.java | pag | 31 |
| 1.3. L'accesso alle API CDS/ISIS | pag | 33 |
| 1.4. Le definizioni del sistema | pag | 37 |
| 1.3.0 ExtGASoaiSysTable.java | pag | 37 |
| 2 .Bibliografia | pag | 39 |

O.Premessa

ExtGASoai è il gateway OAI-PMH per il sistema **ExtGAS/IsisGAS**. **ExtGAS/IsisGAS** è composto dal sistema **IsisGAS** descritto in "Isis/GAS : un sistema multi piattaforma per l'accesso a basi di dati testuali via Internet - ISTI-B4-28, 2003" e **ExtGAS** che accede a sistemi catalografici memorizzati su database system diversi da CDS/ISIS; vedi : " ExtGAS un sistema multipiattaforma per l'accesso ai cataloghi di sistemi biblioteconomici eterogenei - ISTI-B4-15,2005 ". Naturalmente il gateway si applica alla parte **IsisGAS** del sistema **ExtGAS/IsisGAS**.

Quello che segue è un manuale tecnico che descrive l'organizzazione logica del software OAI-PMH.

Quasi tutti gli esempi fanno riferimento alla directory **<distrib>/unix**, per ambiente Unix, ma sono validi anche per ambiente Windows. Non esiste un ambiente preferenziale.

Si presuppone che il lettore abbia conoscenza di OAI-PMH Protocol version 2.0 del 2002-06-14 e del sistema CDS/ISIS, delle sue procedure di generazione, indicizzazione e ordinamento, e conosca, inoltre, il linguaggio di stampa/formattazione dei dati (vedi Bibliografia).

Nel seguito si parlerà indifferentemente di:

1. OAI System File
2. OAI System Table
3. Database System Table
4. Database System File
5. Configuration File

Intendendo il file di configurazione che definisce l'applicazione **ExtGASoai** verso un particolare database CDS/Isis. Lo stesso software può gestire più di un gateway OAI/database per volta. Da ciò consegue che lo OAI System File definisce il database da utilizzare, le modalità di selezione e accesso ai dati e le modalità di restituzione secondo lo standard OAI-PMH Protocol version 2.0 del 2002-06-14.

1. Il software

Il software ExtGASoai è costituito da un insieme di moduli java che implementano i "verbi" OAI e da una API CDS/ISIS per accedere ai dati e produrre i risultati secondo lo standard OAI:

1. **ExtGASoai.java** è il driver principale del sistema
2. **ExtGASoai_v_Id.java** gestisce il verbo Identify
3. **ExtGASoai_v_LI.java** gestisce il verbo ListIdentifiers
4. **ExtGASoai_v_LS.java** gestisce il verbo ListSets
5. **ExtGASoai_v_LMF.java** gestisce il verbo ListMetadataFormats
6. **ExtGASoai_v_LR.java** gestisce il verbo ListRecords
7. **ExtGASoai_v_GR.java** gestisce il verbo GetRecord
8. **ExtGASoaiUtil.java** è un insieme di Utility generali
9. **ExtGASoaiIsisQuery.java** costruisce le query e esegue gli accessi a CDS/ISIS
10. **ExtGASoaiWriteResults.java** produce i risultati da restituire via protocollo OAI-PMH
11. **ExtGASoaiSysTable.java** descrive e definisce i parametri generali del sistema in modo che sia possibile la personalizzazione del Gateway OAI-PMH. Nel seguito verranno referenziati i parametri più importanti, mentre la lista completa e la discussione relativa verranno trattati nel manuale di installazione del Gateway
12. **AccessIsis.java** accede, mediante l'API CDS/ISIS, ai dati.
13. **<wwwisis>** che è la API CDS/ISIS per accedere al database. Esso corrisponde al programma eseguibile `www-isis`, distribuito free nella versione 3.0, per piattaforme unix-linux, unix-solaris, unix-ibmaix, unix-hp e windows, da www.unesco.org e prodotto da BIREME www.bireme.br

Nel complesso il software è costituito da:

1. I driver: da 1 a 7 nella lista precedente
2. I servizi: da 9 a 10 nella lista precedente
3. Le Utility: 8 nella lista precedente
4. Le definizioni: 11 nella lista precedente
5. L'API: 13 nella lista precedente

1.0 I Driver

I driver pilotano la gestione degli OAI verb.

Esiste un driver principale **ExtGASoai.java** che attiva volta per volta i driver di verbo e mantiene la comunicazione con l'Harvester Client.

1.0.0 ExtGASoai.java

ExtGASoai.java è il driver che viene attivato dal servlet ogni qualvolta viene eseguito un accesso al gateway ExtGAS OAI-PMH. Esso viene attivato tramite un proxy-server in modo da rendere trasparente all'utente finale l'attivazione del gateway. In altra maniera è possibile attivare il servlet OAI indicando l'OAI System File da utilizzare e l'indirizzo IP del client chiamante.

Di regola l'accesso al gateway ExtGAS OAI-PMH viene fatto indicando il nome dell'applicazione oai relativa:

`http://<hostname>/<oai-application-name>`

Nell'esempio:

http://leonardo.isti.cnr.it/oai_cnrpub

Il proxy-server leonardo.isti.cnr.it possiede una regola di riscrittura del tipo:

```
RewriteCond %{REQUEST_URI} =/oai_cnrpub
RewriteCond %{QUERY_STRING} ^(.+)$
RewriteRule ^.*
http://metabsrv.isti.cnr.it:8080/ExtGASoai/servlet/oai?Raddr=%{REMOTE_ADDR}&Conf=/export/home/
metaopac/mpisa/oai_cnrpubConf/oai.system.file&%1 [P,L]
```

```
RewriteCond %{REQUEST_URI} =/oai_cnrpub
RewriteCond %{QUERY_STRING} !^(.+)$
RewriteRule ^.*
http://metabsrv.isti.cnr.it:8080/ExtGASoai/servlet/oai?Raddr=%{REMOTE_ADDR}&Conf=/export/home/
metaopac/mpisa/oai_cnrpubConf/oai.system.file [P,L]
```

che accede al servizio sul server OAI metabsrv.isti.cnr.it:8080 assegnando il relativo file di configurazione e in Raddr l'indirizzo IP del client chiamante. In questa maniera è possibile accettare o meno le richieste del client a seconda che sia o meno autorizzato l'indirizzo IP relativo.

`http://metabsrv.isti.cnr.it:8080/ExtGASoai/servlet/oai?Raddr=146.48.82.119&Conf=/export/home/metaopac/mpisa/oai_cnrpubConf/oai.system.file`

L'indirizzo IP può essere: *allowed*, *denied*, *restricted*. Per default è **restricted**, il che permette di accedere, listare, farsi restituire gli OAI identifier/record dei primi `MinObjectsToBeDisplayed` oggetti selezionati dal database e ordinati in ordine crescente

di data. `MinObjectsToBeDisplayed` è definito di default a 5.

Nel caso si volesse evitare l'uso delle regole di riscrittura via proxy-server il gateway `ExtGAS OAI-PMH` può essere attivato nella seguente maniera:

```
http://leonardo.isti.cnr.it:8080/ExtGASoai/servlet/oai?Conf=/export/home/metaopac/mpisa/oai_cnrpubConf/oai.system.file
```

o ancora meglio come

<http://leonardo.isti.cnr.it:8080/ExtGASoai/servlet/oai>

dove si presuppone che il `OAI System File` sia `/oai/oaiConf/oai.system.file` per ambiente unix e `C:\oai\oaiConf\oai.system.file`

Attenzione: il `OAI System File` **deve** esistere a meno di non mandare in crash il gateway.

Sarà compito del driver del servlet calcolare l'indirizzo IP del client chiamante e garantire così la sicurezza dell'accesso.

Si faccia attenzione che in tutti i casi il gateway `OAI-PMH` richiede almeno un verbo definito. Nel caso il verbo mancasse si avrebbero risposte di questo tipo:

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--
OAIpmh gateway versus CDS/ISIS, November 2006
By Beppe Romano giuseppe.romano@isti.cnr.it
-->
<OAI-PMH xmlns="http://www.openarchives.org/OAI/2.0/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/ http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd">
<responseDate>2007-06-20T17:05:56+0200</responseDate>
<request>http://leonardo.isti.cnr.it/oai</request>
<error code="badVerb"/>
</OAI-PMH>
```

Il driver `ExtGASoai.java`:

1. carica dalla sua locazione la `OAI System Table` con la completa descrizione della applicazione.
2. verifica l'esistenza del verbo `OAI` e passa il controllo al modulo relativo.

Ogni modulo viene chiamato sempre con gli stessi tre parametri:

1. `HttpServletRequest`
2. La `OAI System Table`
3. Il `PrintWriter` che ritorna i risultati al `HttpServletResponse`

1..0.1 ExtGASoai_v_Id.java

Il modulo ExtGASoai_v_Id.java gestisce il verbo **Identify**:

1. Se è attivato il TRACE invia un messaggio segnalando l'accesso al servizio.
2. Viene eseguito il log della richiesta specificandone:
 1. L'indirizzo IP del client
 2. I diritti del cliente (allowed,restricted,denied): (**ExtGASoaiUtil/r_Host**)
 3. I parametri specificati nella richiesta (**ExtGASoaiUtil/LogRequest**). Si tenga presente che il parametro **Conf** e **Raddr** sono aggiunti di default dall'implementazione ma non fanno parte del protocollo OAI-PMH
3. Se nell'HttpServletRequest sono contenuti solo i parametri **verb**, **Conf**, **Raddr**:
 1. Viene inviato l'OAI PMHheader contenuto nel parametro **ExtGASoaiSysTable/OAI PMHheader** di OAI System Table,
 2. Viene inviato il ResponseDateHeader contenuto nel parametro **ExtGASoaiSysTable/ResponseDateHeader** di OAI System Table,
 3. Viene inviato l'OAI time e time-zone secondo le specifiche del protocollo.
 4. Viene inviato il ResponseDateTrailer contenuto nel parametro **ExtGASoaiSysTable/ResponseDateTrailer** di OAI System Table,
 5. Vengono inviate le informazioni che identificano il repository e le sue caratteristiche che sono contenute nel parametro **ExtGASoaiSysTable/Identify** di OAI System Table,
 6. Viene inviato l'OAI PMHtrailer contenuto nel parametro **ExtGASoaiSysTable/OAI PMHtrailer** di OAI System Table,
4. Altrimenti invia il messaggio di errore "BadArgument" (**ExtGASoaiWriteResults/WriteBadArgument**)
5. Torna a **ExtGASoai.java**

1.0.2 ExtGASoai_v_LI.java

Il modulo ExtGASoai_v_LI.java gestisce il verbo **ListIdentifiers**:

1. Se è attivato il TRACE invia un messaggio segnalando l'accesso al servizio.
2. Esegue il log come specificato in 1.0.1/2
3. Se nell'HttpServletRequest esiste il parametro **resumptionToken**
 1. sono accettati solo i parametri **verb, Conf, Raddr, resumptionToken**
 2. Altrimenti invia il messaggio di errore "BadArgument" (**ExtGASoaiWriteResults/WriteBadArgument**) e torna a **ExtGASoai.java**
4. Se non è stato specificato il parametro **metadataPrefix**: invia il messaggio di errore "BadArgument" (**ExtGASoaiWriteResults/WriteBadArgument**) e torna a **ExtGASoai.java**
5. Altrimenti se non esiste un **metadataPrefix** uguale a quello richiesto (**ExtGASoaiUtil/ExistsMetadataPrefix**): invia messaggio di errore "cannotDisseminateFormat" e torna a **ExtGASoai.java**
6. Se è stato specificato il parametro **Set** verifica la sua esistenza: (**ExtGASoaiUtil/ExistsSet**)
 1. Se il **Set corrispondente** non è stato definito invia messaggio di errore "noSetHierarchy" e torna a **ExtGASoai.java**
7. Se nell'HttpServletRequest esiste il parametro **resumptionToken**
 1. Esegue una ResumptionTokenQuery (**ExtGASoaiIsisQuery/ResumptionTokenQuery**) restituendo gli **identifier** dei record selezionati.
8. Altrimenti è una query iniziale:
 1. Costruisce e esegue una Query iniziale (**ExtGASoaiIsisQuery/BuildQuery**) restituendo gli **identifier** dei record selezionati.
9. Torna a **ExtGASoai.java**

1.0.3 ExtGASoai_v_LS.java

Il modulo ExtGASoai_v_LS.java gestisce il verbo **ListSets**, produce la lista dei Set che sono stati definiti all'OAI-PMH gateway per quella specifico OAI-PMH System Table. I Set sono definiti mediante il parametro **ExtGASoaiSysTable/SetDefinition** che può contenere un massimo di **ExtGASoaiSysTable/MaxSetDefined** (di default è 100) Set. Ogni set è separato dal successivo dai successivi da uno dei caratteri delimiter contenuti nel parametro **ExtGASoaiSysTable/SetDefinitionDelimiter** (di default vale ";"). Ogni Set contiene 4 sottoparametri separati dai caratteri delimiter contenuti nel parametro **ExtGASoaiSysTable/InternalSetDefinitionDelimiter** (di default vale "+%#\$"):

1. Nome del Set
2. Internal code del Set
3. Descrizione del Set
4. Descrizione del qualifier che ne determina l'appartenenza: suffisso da applicare alla **CDS/Isis Query** per qualificarla.

L'internal code del Set è per ora inutilizzato.

Il driver **ExtGASoai_v_LS.java**:

1. Invia all'Harvester client
 1. L'**OAIpMHheader** dal parametro **ExtGASoaiSysTable/OAIpMHheader**
 2. Il **ResponseDateHeader** dal parametro **ExtGASoaiSysTable/ResponseDateHeader**
 3. L'**OAItime** mediante l'utility **ExtGASoaiUtility/oaiTime**
 4. Il **ResponseDateTrailer** dal parametro **ExtGASoaiSysTable/ResponseDateTrailer**
2. Se è attivato il TRACE invia un messaggio segnalando l'accesso al servizio.
3. Esegue il log come specificato in 1.0.1/2
4. Se non esistono Set definiti:
 1. viene inviato il messaggio di errore "noSetHierarchy"
 2. Torna a **ExtGASoai.java**
5. Altrimenti invia all'Harvester client il ListSetHeader contenuto in **ExtGASoaiSysTable/ListSetHeader**
6. Per ogni Set definito in **ExtGASoaiSysTable/SetDefinition**
 1. Se è attivato il TRACE visualizza, commentando l'XML, l'array dei 4 sottoparametri
 2. Invia all'Harvester client il ListSetHeader contenuto in **ExtGASoaiSysTable/SetHeader**
 3. Invia all'Harvester client il SetSpecHeader contenuto in **ExtGASoaiSysTable/SetSpecHeader**
 4. Invia all'Harvester client il primo parametro dell'array che descrive il set (**Nome del Set**)
 5. Invia all'Harvester client il SetSpecTrailer contenuto in

- ExtGASoaiSysTable/SetSpecTrailer**
6. Invia all'Harvester client il SetNameHeader contenuto in **ExtGASoaiSysTable/SetNameHeader**
 7. Invia all'Harvester client il terzo parametro dell'array che descrive il set (**Descrizione del Set**)
 8. Invia all'Harvester client il SetNameTrailer contenuto in **ExtGASoaiSysTable/SetNameTrailer**
 9. Invia all'Harvester client il SetTrailer contenuto in **ExtGASoaiSysTable/SetTrailer**
 7. Invia all'Harvester client il ListSetTrailer contenuto in **ExtGASoaiSysTable/ListSetTrailer**
 8. Invia all'Harvester client l'**OAI PMH Trailer** definito nel parametro **ExtGASoaiSysTable/OAI PMH Trailer**
 9. Torna a **ExtGASoai.java**

1.0.4 ExtGASoai_v_LMF.java

Il modulo ExtGASoai_v_LMF.java gestisce il verbo **ListMetadataFormats**, produce la lista dei MetadataFormat che sono stati definiti all'OAI-PMH gateway per quella specifico OAI-PMH System Table. I MetadataFormat sono definiti mediante il parametro **ExtGASoaiSysTable/ListMetadataFormats**. Poiché, secondo la corrente implementazione del gateway, i record contenuti nel CDS/Isis database non contengono, al loro interno, la qualifica dei **metadataPrefix** supportati, in modo da avere record dello stesso database che possano essere disseminati in modo diverso, la definizione dei **metadataFormat** è estesa a tutti i record del database e quindi la definizione dei **metadataFormat** diviene statica ed è realizzata secondo il seguente modello:

```
ListMetadataFormats <ListMetadataFormats>
ListMetadataFormats <metadataFormat>
ListMetadataFormats <metadataPrefix>oai_dc</metadataPrefix>
ListMetadataFormats <schema>http://www.openarchives.org/OAI/2.0/oai\_dc.xsd </schema>
ListMetadataFormats <metadataNamespace>http://www.openarchives.org/OAI/2.0/oai\_dc/ </metadataNamespace>
ListMetadataFormats </metadataFormat>
ListMetadataFormats <metadataFormat>
ListMetadataFormats <metadataPrefix>oai_rfc1807</metadataPrefix>
ListMetadataFormats <schema>http://www.openarchives.org/OAI/1.1/rfc1807.xsd </schema>
ListMetadataFormats <metadataNamespace>http://www.w3.org/2001/XMLSchema-instance </metadataNamespace>
ListMetadataFormats </metadataFormat>
ListMetadataFormats </ListMetadataFormats>
```

I **metadataFormat** così descritti vengono caricati rigidamente nell'ordine di descrizione e utilizzati per il verbo **ListMetadataFormats** e per l'identificazione del **metadataPrefix** che poi individua il formato di stampa CDS/Isis da usare per la restituzione dei record. Non è pensabile, quindi, che esista un ExtGAS OAI-PMH gateway verso CDS/Isis che non possieda almeno un **metadataPrefix** definito all'interno di un **ListMetadataFormats**.

Si tenga presente che la sequenza:

```
<metadataPrefix>oai_dc</metadataPrefix>
```

e

```
<metadataPrefix>oai_rfc1807</metadataPrefix>
```

è rigidamente **non modificabile**, nel senso che non sono ammessi spazi.

Il driver **ExtGASoai_v_LMF.java**:

1. Se è attivato il TRACE invia un messaggio segnalando l'accesso al servizio.
2. Esegue il log come specificato in 1.0.1/2
3. Se nell'HttpServletRequest esiste il parametro **identifier**
 1. Seleziona l'**identifier** utilizzando il servizio **ExtGASoaiIsisQuery/BuildQueryIdentifier**
 2. Se la **ResultList** ottenuta indietro non contiene il CDS/Isis Master File record Number invia il messaggio di errore "idDoesNotExist" mediante il servizio **ExtGASoaiWriteResults/WriteIdDoesNotExists** e Torna al driver di primo livello

4. Invia all'Harvester client: L'**OAIpMHeader** dal parametro **ExtGASoaiSysTable/OAIpMHeader**
5. Invia all'Harvester client:Il **ResponseDateHeader** dal parametro **ExtGASoaiSysTable/ResponseDateHeader**
6. Invia all'Harvester client:L'**OAItime** mediante l'utility **ExtGASoaiUtility/oaiTime**
7. Invia all'Harvester client:Il **ResponseDateTrailer** dal parametro **ExtGASoaiSysTable/ResponseDateTrailer**
8. Invia all'Harvester client: il contenuto di **ExtGASoaiSysTable/ListMetadataFormats** così come definiti.
10. Invia all'Harvester client l'**OAIpMTrailer** definito nel parametro **ExtGASoaiSysTable/OAIpMTrailer**
9. Torna al driver di primo livello

1.0.5 ExtGASoai_v_LR.java

Il modulo ExtGASoai_v_LR.java gestisce il verbo **ListRecords**:

1. Se è attivato il TRACE invia un messaggio segnalando l'accesso al servizio.
2. Esegue il log come specificato in 1.0.1/2
3. Se nell'HttpServletRequest esiste il parametro **resumptionToken**
 1. sono accettati solo i parametri **verb, Conf, Raddr, resumptionToken**
 2. Altrimenti invia il messaggio di errore "BadArgument" (**ExtGASoaiWriteResults/WriteBadArgument**) e torna a **ExtGASoai.java**
4. Se non è stato specificato il parametro **metadataPrefix**: invia il messaggio di errore "BadArgument" (**ExtGASoaiWriteResults/WriteBadArgument**) e torna a **ExtGASoai.java**
5. Altrimenti se non esiste un **metadataPrefix** uguale a quello richiesto (**ExtGASoaiUtil/ExistsMetadataPrefix**): invia messaggio di errore "cannotDisseminateFormat" e torna a **ExtGASoai.java**
6. Se è stato specificato il parametro **Set** verifica la sua esistenza: (**ExtGASoaiUtil/ExistsSet**)
 1. Se il **Set corrispondente** non è stato definito invia messaggio di errore "noSetHierarchy" e torna a **ExtGASoai.java**
7. Se nell'HttpServletRequest esiste il parametro **resumptionToken**
 1. Esegue una ResumptionTokenQuery (**ExtGASoaiIsisQuery/ResumptionTokenQuery**) restituendo i record (**ExtGASoaiWriteResults/Write_OAI_record**) selezionati.
8. Altrimenti è una query iniziale:
 1. Costruisce e esegue una Query iniziale (**ExtGASoaiIsisQuery/BuildQuery**) restituendo i record selezionati (**ExtGASoaiWriteResults/Write_OAI_record**).
9. Torna a **ExtGASoai.java**

1.0.6 ExtGASoai_v_GR.java

Il modulo ExtGASoai_v_GR.java gestisce il verbo **GetRecords**:

1. Se è attivato il TRACE invia un messaggio segnalando l'accesso al servizio.
2. Esegue il log come specificato in 1.0.1/2
3. Se nell'HttpServletRequest non esiste il parametro **identifier**
 1. Invia il messaggio di errore "BadArgument" (**ExtGASoaiWriteResults/WriteBadArgument**) e torna a **ExtGASoai.java**
4. Se non è stato specificato il parametro **metadataPrefix**: invia il messaggio di errore "BadArgument" (**ExtGASoaiWriteResults/WriteBadArgument**) e torna a **ExtGASoai.java**
5. Altrimenti se non esiste un **metadataPrefix** uguale a quello richiesto (**ExtGASoaiUtil/ExistsMetadataPrefix**): invia messaggio di errore "cannotDisseminateFormat" e torna a **ExtGASoai.java**
6. Se è stato specificato il parametro **Set** verifica la sua esistenza: (**ExtGASoaiUtil/ExistsSet**)
 1. Se il **Set corrispondente** non è stato definito invia messaggio di errore "noSetHierarchy" e torna a **ExtGASoai.java**
7. Altrimenti è una query iniziale:
 1. Costruisce e esegue una Query utilizzando un **identifier** (**ExtGASoaiIsisQuery/BuildQueryIdentifier**) restituendo il record selezionato (**ExtGASoaiWriteResults/Write_OAI_record**).
8. Torna a **ExtGASoai.java**

1.1 I Servizi

I servizi sono utilizzati per accedere ai dati, preparare le liste di restituzione, preparare gli identifier da restituire e i record da restituire per mezzo degli specifici formati (metatadatPrefix).

1.1.0 Il modello di selezione e accesso ai dati

Poiché per il protocollo OAI-PMH la selezione degli oggetti può essere fatta esclusivamente applicando i limiti temporali (parametri **from** e **until**) e l'appartenenza o meno a un **Set** e la ricerca sui campi data non esiste in CDS/ISIS, è stato necessario applicare un particolare algoritmo per la selezione degli oggetti via campo data (in formato carattere) utilizzando o meno il carattere di mascheramento a destra ('\$') per ottenere la **ResultList** di tutti quegli oggetti che appartengono a una lista definita con anno di inizio e anno di fine rappresentato come carattere. Successivamente, ottenuta la lista degli oggetti selezionati, essa viene ulteriormente verificata con il valore numerico degli anni mesi e giorni limite tramite il software sviluppato. In questa maniera si ha un primo processo che seleziona gli identifier velocemente accedendo all'inverted file con chiave mascherata o meno che sia, e successivamente ogni identifier viene o meno accettato valutandone il valore numerico della data.

Si tenga presente che i parametri **from** e **until** non sono obbligatori per il protocollo OAI-PMH. Per il funzionamento del Gateway stesso, in caso di mancanza completa di entrambi i parametri o di uno di essi vengono applicati i valori limite sostitutivi che sono memorizzati nella OAI.PMH System Table nei parametri **ExtGASoaiSysTable/ExtGASoaiUpperDate** (default: 20151231) e **ExtGASoaiSysTable/ExtGASoaiLowerDate** (default: 20000101)

Nell'esempio che segue vengono selezionate le pubblicazioni del **Set** "IENIGE" con **from=20060401** e **until=20070331** e **metadataPrefix=oai_dc**.

La seguente è la URL della richiesta:

http://leonardo.isti.cnr.it/oai_cnrpub?verb=ListIdentifiers&from=2006-04-01&until=2007-03-31&set=IENIGE&metadataPrefix=oai_dc

La query CDS/Isis generata dal gateway è la seguente:

((2006\$/(5) or 2007\$/(5)) AND ap/(904) and IENIGE/(900))

I due anni, 2006 e 2007 sono messi in or e mascherati a destra. Vengono, inoltre, qualificati come appartenenti al relativo field 5. La qualificazione si ottiene assegnando il valore relativo al parametro **ExtGASoaiSysTable/ExtGASoaiDateQualifier** (**ExtGASoaiDateQualifier / (5))**

Viene così prodotto una lista temporanea, **ResultList**, che contiene 775 oggetti:

```
<NumberOfSelectedRecords>
775:775
</NumberOfSelectedRecords>
<Start>
<Record>;20060109;Metapub:cnr.ieni/cnr.ieni.ge/1999-A0-008;000492;IENIGE,AP;</Record>
<Record>;20060109;Metapub:cnr.ieni/cnr.ieni.ge/1997-A0-008;000493;IENIGE,AP;</Record>
<Record>;20060109;Metapub:cnr.ieni/cnr.ieni.ge/1991-A0-002;000495;IENIGE,AP;</Record>
.....
.....
<Record>;20060303;Metapub:cnr.ieni/cnr.ieni.ge/1995-A0-009;000771;IENIGE,AP;</Record>
<Record>;20060317;Metapub:cnr.ieni/cnr.ieni.ge/2002-A0-017;000923;IENIGE,AP;</Record>
<Record>;20060410;Metapub:cnr.ieni/cnr.ieni.ge/1999-A2-002;000780;IENIGE,AP;</Record>
<Record>;20060410;Metapub:cnr.ieni/cnr.ieni.ge/1995-A2-001;000784;IENIGE,AP;</Record>
.....
.....
<Record>;20060728;Metapub:cnr.ieni/cnr.ieni.ge/2004-A3-022;001048;IENIGE,AP;</Record>
<Record>;20060728;Metapub:cnr.ieni/cnr.ieni.ge/2004-A3-023;001049;IENIGE,AP;</Record>
<Record>;20070426;Metapub:cnr.ieni/cnr.ieni.ge/2006-A0-023;000466;IENIGE,AP;</Record>
<Record>;20070426;Metapub:cnr.ieni/cnr.ieni.ge/2006-A0-019;000903;IENIGE,AP;</Record>
<Record>;20070426;Metapub:cnr.ieni/cnr.ieni.ge/2006-A0-008;001050;IENIGE,AP;</Record>
.....
.....
<Record>;20070606;Metapub:cnr.ieni/cnr.ieni.ge/2006-A6-017;001184;IENIGE,AP;</Record>
<Record>;20070606;Metapub:cnr.ieni/cnr.ieni.ge/2007-PR-002;001185;IENIGE,AP;</Record>
<Record>;20070606;Metapub:cnr.ieni/cnr.ieni.ge/2006-PR-013;001186;IENIGE,AP;</Record>
</Start>
```

Tale lista viene successivamente ridotta a 472 elementi mediante il test di inclusione dei valori numerici dei parametri **from** e **until** confrontati con la data di pubblicazione/aggiornamento dei metadati:

```
Resume from [cnr.1182440921475.0.RT]
Complete List Size : 472
Cursor : 0
Expiration Date : 2007-06-21T17:48:41+0200
```

e viene reso disponibile il primo identifier valido come qui in esempio.

```
identifier : Metapub:cnr.ieni/cnr.ieni.ge/1999-A2-002
datestamp : 2006-04-10
setSpec : IENIGE
setSpec : AP
```

La costruzione della lista in esempio viene realizzata utilizzando il formato:

```
<ExtGASoaiArchiveName>_li.pft
```

Il cui nome è calcolato per default dal sistema con il parametro
ExtGASoaiSysTable/ListIdentifiersExtGASoaiFormat

L'esempio che segue è il formato `cnr_li.pft`:

```
mhl,if val(v1001) = val(v1031) then '<NumberOfSelectedRecords>',/,F((val(v1032)-
val(v1031)+1),3,0),':',v1002,','</NumberOfSelectedRecords>',/,'<Start>',/, mhl,ref(['cnr']val(v1^m),if
p(v5^b) then '<Record>',v5^b,if p(v1) then ';Metapub:',v1,',';mf,',';v900,',';v904,','</Record>',/,fi,fi)
```

Il formato suddetto utilizza un formato di ordinamento, per restituire i record ordinati dal più vecchio al più nuovo, che è definito nel parametro

ExtGASoaiSysTable/ExtGASoaiSortPftFile il cui valore volutamente assegnato è "@Sdate.pft" (mentre per default è null) che è così definito:

```
mhu,if p(v5^b) then '^t',v5^b,' ^m',mf,fi
```

essendo il campo 5 costituito da due sottocampi: ^a che contiene la data in formato legale e ^b che contiene invece il formato numerico della stessa data:

```
^a31/12/2006^b20063112
```

1.1.0.1 L'organizzazione della lista dei risultati

La struttura dell'elemento della lista è costituita da 6 elementi separati dal parametro **ExtGASoaiSysTable/ExtGASoaiDataRecordDelimiter** che per default vale carattere/stringa ";" (punto_e_virgola):

1. "<Record>" stringa costante: **obbligatoria**
2. data in formato numerico del tipo yyyymmdd, yyyymm, yyyy : **obbligatoria**
3. OAI-PMH Identifier: **obbligatorio**
4. CDS/ISIS Master File Record Number (viene usato per il retrieval) : **obbligatorio**
5. Lista dei **Set**, separati da un **SetIdentifierSetDelimiter** che vale di default "," (virgola), cui appartiene l'OAI Identifier. Se non esistono Set inserire il valore "NullSet": **obbligatorio**
6. "</Record>" stringa costante: **obbligatoria**

L'OAI-PMH identifier è costituito da una stringa di caratteri separati da separatori specifici tale da identificare univocamente il record nel repository.

1.1.0.2 La struttura dell'OAI-PMH identifier

1. **IdentifierPrefix** che di default vale null.
2. Set o lista di Set, separati gerarchicamente dal carattere ':' (due_punti) così come specificato nel protocollo OAI-PMH, e separati dall'effettivo identifier da **SetIdentifierDelimiter** che di default vale "::::". Tali Set, se esistono, vengono utilizzati per qualificare l'effettivo identifier al momento della sua selezione per ottenere i metadati.
3. Effettivo identifier: **obbligatorio**

1.1.0.3 Il modello di rappresentazione delle date richiesto dal gateway

Sono accettati come modelli di date i seguenti mapping:

1. `yyyymmdd`
2. `yyyy-mm-dd`
3. `yyyymm`
4. `yyyy-mm`
5. `yyyy`

Il default definito nella OAI SystemTable mediante il parametro **ExtGASoaiSysTable/DateExtGASoaiFormat** è `yyyymmddd`.

Escludendo il mapping 5 tutti i rimanenti richiedono, per poter essere selezionati, un carattere "\$" per il mascheramento a destra. In questa maniera verranno selezionati tutti gli oggetti che hanno data di modifica che inizia per "yyyy" che può determinare una espansione della query mettendo in "or" tutti gli anni compresi e uguali tra i valori di **from** e **until**.

Lo stesso si applica al caso 5 senza mascheramento

1.1.0.4 Il modello di restituzione di Identifiers/Records

Il modello di accesso e selezione descritto sopra viene utilizzato per i verbi ListIdentifiers e ListRecords. Successivamente nel caso di:

1. ListIdentifiers: leggendo la lista degli oggetti selezionati, vengono inviati gli OAI identifiers e se necessario costruito il resumptionToken restart file e inviato il ResumptionToken pointer (vedi più avanti nell'esempio)
2. ListRecords: leggendo la lista degli oggetti selezionati, mediante il CDS/ISIS Master File Record Number, vengono acceduti i record e prodotte le restituzioni, in accordo al metadataPrefix specificato, per un numero massimo di record pari a **ExtGASoaiSysTable/MaxObjectsToBeDisplayed**. Nel caso sia necessario viene costruito il resumptionToken restart file e inviato il ResumptionToken pointer.

Nel caso del verbo GetRecord l'accesso viene fatto tramite l'OAI identifier del record. L'OAI identifier viene smembrato nei suoi componenti (vedi sopra) e viene costruita una particolare query per identificare univocamente il record. Nel caso del seguente OAI identifier:

`isti:CNR:oai:Genova::cnr.ieni/cnr.ieni.ge/2003-A0-001`

dove:

1. **ExtGASoaiSysTable/IdentifierPrefix** è definito come `isti:CNR:oai:`
2. **ExtGASoaiSysTable/SetIdentifierDelimiter** definito come `::` da cui deriva che, dato l'**IdentifierPrefix** in 1, esiste anche, definito nell'identifier, un Set qualificato di valore **Genova**.

3. Un Set definito a livello generale con il parametro **ExtGASoaiSysTable/SetDefinition** di valore:

Genova+x_ienige+cnr Genova publications+AND ap/(904) and (IENIGE/(900) or IBFGE/(904));

Dove **InternalSetDefinitionDelimiter** vale uno dei seguenti caratteri "+%# \$" e **SetDefinitionDelimiter** vale ';' (punto_e_virgola)

4. Un effettivo identifier di valore **cnr.ieni/cnr.ieni.ge/2003-A0-001** su cui è definito il qualifier **ExtGASoaiIdentifierQualifier** (ExtGASoaiIdentifierQualifier /(1)) e l'operatore **ExtGASoaiDataIdentifierOperator** impostato a (ExtGASoaiDataIdentifierOperator " \$ ") (dollaro_spazio_dollaro)

La query risultante è:

((cnr/(1) \$ ieni/(1) \$ cnr/(1) \$ ieni/(1) \$ ge/(1) \$ 2003/(1) \$ A0/(1) \$ 001/(1)) AND ap/(904) and IENIGE/(900))

dove il Field 1 rappresenta la identificazione univoca del documento nel suo contesto:

cnr.ieni/cnr.ieni.ge/2003-A0-001

Essa viene usata per realizzare una lista di risultati come per i verbi ListIdentifiers e ListRecords, ma composta da una unica entry di tipo <Record>.....</Record>. Successivamente tramite il Master File record Number viene prodotto il record da inviare al client secondo il metadataPrefix specificato. Nell'esempio:

```
<NumberOfSelectedRecords>
1:1
</NumberOfSelectedRecords>
<Start>
<Record>;20051129;isti:CNR:oai:Genova::cnr.ieni/cnr.ieni.ge/1993-A0-001;000404;IENIGE,AP,Genova;</Record>
</Start>
```

Si tenga presente che nel formato di restituzione delle liste di OAI-PMH identifier il blocco:

```
<NumberOfSelectedRecords>
n:m
</NumberOfSelectedRecords>
```

non è assolutamente obbligatorio.

1.1.1 Il restart dell'accesso ai dati

Per assicurarsi la completa certezza dello scaricamento è opportuno che il server OAI invii all'Harvester un numero limitato di record/identifier per transazione. L'OAI server invierà all'Harvester piccoli blocchi di record/identifier dando sempre la possibilità di restarting, a certi livelli, della transazione. Per ottenere questo risultato ogni qualvolta

vengono inviate piccole quantità di dati all'Harvester viene inviato il nome di un pointer che rende disponibile il blocco successivo di informazioni e che ha una "expiration date" tale da permettere a qualunque livello il caricamento parziale dei dati persi durante la trasmissione.

Il restart pointer viene chiamato ResumptionToken (identificatore che permette il ripristino/continuazione dell'operazione di scaricamento).

Il resumptionToken viene identificato da un nome univoco che identifica un file memorizzato su una particolare directory del server che è definita nel parametro **ExtGASoaiSysTable/ResumptionTokenDir** che è di default caricato come:

1. /oai/RT/ in ambiente unix
2. C:\oai\RT\ in ambiente Windows

Il nome del file è costruito da **ExtGASoaiUtil/GetResumptionTokenFileName** nella seguente maniera:

1. **ResumptionTokenDir**
2. un prefisso contenuto nel parametro **ExtGASoaiSysTable/ExtGASoaiResumptionTokenFilePrefix** che di default è caricato con il valore del parametro **ExtGASoaiSysTable/ExtGASoaiArchiveName**
3. "." : il carattere punto.
4. data e ora espressa in millisecondi a partire dal 1 gennaio 1974
5. "." : il carattere punto seguito ad un valore numerico calcolato dinamicamente dal sistema per rendere univoco il nome del file
6. "." : il carattere punto.
7. un suffisso contenuto nel parametro **ExtGASoaiSysTable/ExtGASoaiResumptionTokenFileSuffix** che di default vale la stringa "RT"

Il seguente è un esempio di resumptionToken file/directory:

```
/home/metaopac/mpisa/OAISAVE/cnr.1182529863184.0.RT  
/home/metaopac/mpisa/OAISAVE/cnr.1182529876784.0.RT  
/home/metaopac/mpisa/OAISAVE/cnr.1182529882474.0.RT
```

dove **ExtGASoaiSysTable/ResumptionTokenDir** è /home/metaopac/mpisa/OAISAVE/

Per default il resumptionToken ha un Expiration Date espressa in millisecondi nel parametro **ExtGASoaiSysTable/ResumptionTokenTTL** che vale 3600000 (un'ora).

Il resumptionToken File contiene un massimo di 10 parametri così come definito nel parametro **ExtGASoaiSysTable/MaxResumptionTokenParms** separati da un separatore in scrittura che è **ExtGASoaiSysTable/ResumptionTokenWriteDelim** e vale il carattere "#"

mentre in lettura sono separati da uno dei caratteri definiti in **ExtGASoaiSysTable/ResumptionTokenReadDelim** che vale uno dei seguenti caratteri "%#:,;".

La struttura del **resumptionToken** record è costituita dai seguenti parametri suddivisi dal delimiter **ExtGASoaiSysTable/ResumptionTokenWriteDelim** :

1. Numero totale di record/identifier selezionati
2. Numero di record/identifier di inizio inviato
3. Numero di record/identifier di fine inviato
4. Valore del **dateFrom**
5. Valore del **dateUntil**
6. CDS/Isis query atta a selezionare gli stessi oggetti
7. OAI-PMH verb utilizzato (**ListIdentifiers/ListRecords**)
8. Formato di restituzione corrispondente al **metadataPrefix** selezionato

Il seguente è un esempio di **resumptionToken** file:

2944#11#20#20050101#20070331#2005\$(5) or 2006\$(5) or 2007\$(5)#ListRecords#@oai_dc.pft

La richiesta di esecuzione di un verbo(7) **ListIdentifiers** o **ListRecords** a partire da un **resumptionToken** identifier provoca la riesecuzione completa della richiesta **CDS/Isis**(6) e successivamente la selezione degli oggetti richiesti e l'invio degli oggetti stessi secondo il **metadataPrefix**(8) definito.

1.1.2 Il software che gestisce i servizi

Il software che gestisce i servizi opera nei seguenti modi verso i relativi verbi:

1. ListIdentifiers
 1. Se non esiste resumptionToken
 1. traduce e esegue la richiesta
 2. Analizza la **ResultList** verifica l'accettabilità dei record e restituisce i primi Identifier fino a **ExtGASoaiSystable/MaxObjectTobeDisplayed** così come essi sono presenti in ogni elemento della lista dei risultati accettabili.
 3. Produce, se necessario, il successivo resumptionToken.
 2. Se esiste il resumptioToken
 1. legge il resumptionToken e ne valida l'expiration date
 2. Esegue la richiesta specificata nel resumptionToken
 3. Analizza la **ResultList** validando la accettabilità dei record. Invia gli identifier relativi a partire dal numero di identifier di fine inviato, maggiorato di uno, fino a **ExtGASoaiSystable/MaxObjectTobeDisplayed**
 4. Produce, se necessario, il successivo resumptionToken.
2. ListRecords
 1. Se non esiste resumptionToken
 1. traduce e esegue la richiesta
 2. Analizza la **ResultList** e verifica l'accettabilità dei record e si prepara a restituire i primi record accettati fino a **ExtGASoaiSystable/MaxObjectTobeDisplayed**
 3. Legge il record utilizzando direttamente il suo CDS/Isis Master File Record Number, lo prepara per la restituzione rendendolo compatibile con il metadataPrefix specificato e lo invia al Harvester client.
 4. Produce, se necessario, il successivo resumptionToken.
 2. Se esiste il resumptionToken
 1. legge il resumptionToken e ne valida l'expiration date.
 2. esegue la richiesta già preparata.
 3. Analizza la **ResultList** validando la accettabilità dei record. Si prepara a restituire i record a partire dal numero di record di fine inviato maggiorato di uno fino a **ExtGASoaiSystable/MaxObjectTobeDisplayed**.
 4. Legge il record utilizzando direttamente il suo CDS/Isis Master File Record Number, lo prepara per la restituzione rendendolo compatibile con il metadataPrefix specificato e lo invia al Harvester client.
 5. Produce, se necessario, il successivo resumptionToken.
3. GetRecord
 1. riceve l'OAI-PMH identifier
 2. elimina il prefisso **ExtGASoaiSystable/IdentifierPrefix**, se esiste
 3. se esiste una definizione di **Set** associato all'identifier: lo elimina dalla stringa mediante il **ExtGASoaiSystable/SetIdentifierDelimiter** e restringe la ricerca aggiungendo alla **CDS/Isis Query** la selezione relativa al set specifico (se il set è stato definito e esiste)

4. esegue la richiesta di selezione che produce un solo Record nella **ResultList**.
5. Legge il record utilizzando direttamente il suo CDS/Isis Master File Record Number, lo prepara per la restituzione rendendolo compatibile con il metadataPrefix specificato e lo invia al Harvester client.

1.1.2.0 ExtGASoaiIsisQuery.java

Il modulo IsisGASoaiIsisQuery.java contiene i seguenti servizi:

1. **TotalSelectedRecords**: che data una **CDS/Isis Query** seleziona il numero di record che soddisfano la richiesta e lo rende disponibile al sistema nella OAI-PMH System Table mediante il parametro **ExtGASoaiSysTable/TotalSelectedRecords** che viene inizializzato a null.
 1. Torna al Servizio chiamante
2. **RealSetName**: che dato un **SetName** (nome di Set) ritorna, se esiste, la definizione del relativo **Set** che è stata definita in OAI-PMH System Table.
 1. Torna al Servizio chiamante
3. **RealIdentifier**: che ritorna il reale Identifier del record inteso come insieme di stringhe, derivate dall'OAI-PMH identifier che riescono a qualificare, all'interno del database, in maniera univoca il record.
 1. Torna al Servizio chiamante
4. **BuildQueryIdentifier**: che costruisce utilizzando i servizi (**RealIdentifier** e **RealSetName**) tramite i parametri **ExtGASoaiSysTable/IdentifierPrefix** e **ExtGASoaiSysTable/SetIdentifierDelimiter** la **CDS/IsisQuery** per selezionare il relativo univoco record e costruire la relativa **ResultList** che conterrà il Master File record Number del record da inviare in accordo al **metadataPrefix** selezionato.
 1. Se il verbo è **ListMetadataFormats**
 1. Ritorna il pointer della relativa **ResultList** al driver chiamante di secondo livello (**ExtGASoai_v_LMF**)
 2. Se l'IP address dell'Harvester client è **allowed** produce il risultato per il client stesso attivando il servizio **ExtGASoaiWriteResult/Results**
 3. Altrimenti invia messaggio di errore "idDoesNotExist" tramite il servizio **ExtGASoaiWriteResults/WriteIdDoesNotExists**
 4. Torna al driver chiamante di secondo livello
5. **ResumptionTokenQuery**: che identifica il relativo **resumptionToken** file ne verifica la sua validità formale (**ExtGASoaiResumptionTokenFileSuffix** e **ExtGASoaiResumptionTokenFilePrefix**) e la sua validità temporale (**ResumptionTokenTTL**) e lo concatena al suo **ResumptionTokenDir** e manda in esecuzione la richiesta (**BuildResumptionTokenQuery**)
 1. Torna al driver chiamante di secondo livello
6. **BuildResumptionTokenQuery**: che legge il relativo **resumptionToken**, ne estrae la **CDS/Isis Query** precedentemente memorizzata da un **ListIdentifiers** o un **ListRecords**, e la manda in esecuzione specificando il formato di restituzione che corrisponderà alla costruzione di una **ResultList**.
 1. Produce il risultato per l'Harvester client attivando il servizio

ExtGASoaiWriteResult/Results

2. Torna a **ResumptionTokenQuery**

7. **BuildQuery**: che costruisce la **CDS/Isis Query** relativa ai verbi **ListIdentifiers** e **ListRecords**:
 1. Se non esiste il parametro **from**
 1. viene utilizzato come valore **ExtGASoaiSysTable/ExtGASoaiLowerDate**
 2. se **from** non è formalmente corretto invia il messaggio di errore "BadArgument" (**ExtGASoaiWriteResults/WriteBadArgument**) e torna al driver chiamante di secondo livello
 2. Se non esiste il parametro **until**
 1. viene utilizzato come valore **ExtGASoaiSysTable/ExtGASoaiUpperDate**
 2. se **until** non è formalmente corretto invia il messaggio di errore "BadArgument" (**ExtGASoaiWriteResults/WriteBadArgument**) e torna al driver chiamante di secondo livello
 3. Se **from** e **until** non sono logicamente coerenti con **ExtGASoaiSysTable/ExtGASoaiUpperDate** e con **ExtGASoaiSysTable/ExtGASoaiLowerDate** o **from** è maggiore di **until** invia il messaggio di errore "BadArgument" (**ExtGASoaiWriteResults/WriteBadArgument**) e torna al driver chiamante di secondo livello
 4. Viene identificato il tipo di data da trattare (vedi 1.1.0.3 Il modello.....)
 5. Viene costruita la **CDS/Isis Query**, mascherando a destra con il carattere '\$' se necessario, tutte le stringhe anno comprese tra il valore dell'anno di **from** e quello di **until**: le stringhe sono messe in **or**. Se necessario le stringhe sono qualificate con il parametro **ExtGASoaiSysTable/ExtGASoaiDateQualifier**.
 6. Se esiste un **Set** viene utilizzato il servizio **RealSetName** per applicare alla **CDS/Isis Query** le restrizioni del caso.
 7. La **CDS/Isis Query** viene aggiornata aggiungendo, se esistono i valori dei parametri **ExtGASoaiSysTable/GeneralQueryPrefix** e **ExtGASoaiSysTable/GeneralQuerysuffix**.
 8. Utilizza il servizio **TotalSelectedRecords** per verificare il numero di record che soddisfano la **CDS/Isis Query**: Se esistono record selezionati
 1. Calcola i formati di restituzione per produrre la **ResultList** per entrambi i verbi **ListIdentifiers** e **ListRecords**
 2. Stabilisce i diritti di accesso per l'Harvester client
 1. **allowed** accede al massimo numero di record possibili
 2. **denied** accede a zero record
 3. **restricted** accede al valore contenuto nel parametro **ExtGASoaiSysTable/MinObjectsToBeDisplayed** (default 5)
 3. Accede al database
 9. Produce il risultato per l'Harvester client attivando il servizio **ExtGASoaiWriteResult/Results**
 10. Torna al driver chiamante di secondo livello

1.1.2.1 ExtGASoaiWriteResults.java

Il modulo IsisGASoaiWriteResults.java contiene i seguenti servizi:

11. **WriteEmptyResumptionToken**: invia il **resumptionToken** conclusivo (esso è vuoto e rappresenta la fine dei record/identifier selezionati) all'HarvesterClient e torna al servizio chiamante
12. **WriteResumptionToken**: scrive il **resumptionToken file** e manda all'Harvester client informazioni su di esso.
 1. Calcola il nome univoco del **resumptionToken file** mediante la utility **ExtGASoaiUtility/GetResumptionTokenFileName**
 2. Prepara il **resumptionToken record**, secondo come descritto in "1.1.1 Il restart dell'accesso ai dati" e lo scrive sul **resumptionToken file**.
 3. Invia il **resumptionToken** all'Harvester client
 4. Torna al servizio chiamante
13. **GetWriteSetNames**: dato un **Set** o una **gerarchia** di **Set**; se sono stati definiti con il parametro **ExtGASoaiSysTable/SetDefinition** vengono resi disponibili al servizio chiamante.
14. **Write_OAI_identifier**: invia all'Harvester client il contenuti dell' **OAI_identifier**
 1. Invia all'Harvester client:
 1. l'OAI Header contenuto nel parametro **ExtGASoaiSysTable/HeaderHeader**
 2. l'OAI Identifier contenuto nel parametro **ExtGASoaiSysTable/IdentifierHeader**
 3. L'OAI identifier calcolato
 4. l'OAI TrailerIdentifier contenuto nel parametro **ExtGASoaiSysTable/IdentifierTrailer**
 5. Mediante il servizio **ExtGASoaiWriteResults/GetWriteSetNames** se esistono set associati al Record invia all'Harvester client per ogni set:
 1. L'OAI SetSpecHeader contenuto nel parametro **ExtGASoaiSysTable/SetSpecHeader**
 2. Il **SetName**
 3. L'OAI SetSpecTrailer contenuto nel parametro **ExtGASoaiSysTable/SetSpecTrailer**
 6. l'OAI datestampHeader contenuto nel parametro **ExtGASoaiSysTable/datestampHeader**
 7. Il datestamp calcolato
 8. l'OAI datestampTrailer contenuto nel parametro **ExtGASoaiSysTable/datestampTrailer**
 9. L'OAI HeaderTrailer contenuto nel parametro **ExtGASoaiSysTable/HeaderTrailer**
 2. Torna al servizio chiamante

15. **Write_OAI_record**: invia all'Harvester client il contenuto dell' **OAI_record** in accordo con il **metadataPrefix** selezionato

1. Accede al database con Master File Record Number definito.
2. Invia all'Harvester client:
 1. L'OAI RecordHeader contenuto nel parametro **ExtGASoaiSysTable/RecordHeader**
 2. L'OAI Header contenuto nel parametro **ExtGASoaiSysTable/HeaderHeader**
 3. L'OAI HeaderIdentifier contenuto nel parametro **ExtGASoaiSysTable/IdentifierHeader**
 4. L'OAI identifier calcolato
 5. L'OAI TrailerIdentifier contenuto nel parametro **ExtGASoaiSysTable/IdentifierTrailer**
 6. L'OAI datestampHeader contenuto nel parametro **ExtGASoaiSysTable/datestampHeader**
 7. Il datestamp calcolato
 8. L'OAI datestampTrailer contenuto nel parametro **ExtGASoaiSysTable/datestampTrailer**
 9. Mediante il servizio **ExtGASoaiWriteResults/GetWriteSetNames** se esistono set associati al Record invia all'Harvester client per ogni set:
 1. L'OAI SetSpecHeader contenuto nel parametro **ExtGASoaiSysTable/SetSpecHeader**
 2. Il **SetName**
 3. L'OAI SetSpecTrailer contenuto nel parametro **ExtGASoaiSysTable/SetSpecTrailer**
 10. L'OAI HeaderTrailer contenuto nel parametro **ExtGASoaiSysTable/HeaderTrailer**
 11. L'OAI MetadataHeader contenuto nel parametro **ExtGASoaiSysTable/MetadataHeader**
 12. Il contenuto del record mediante la utility **ExtGASoaiUtil/WriteHTMLline**
 13. L'OAI MetadataTrailer contenuto nel parametro **ExtGASoaiSysTable/MetadataTrailer**
 14. L'OAI RecordTrailer contenuto nel parametro **ExtGASoaiSysTable/RecordTrailer**
3. Torna al servizio chiamante

16. **WriteList**: Invia un identifier o un record all'Harvester client

1. Se è il primo elemento di una lista di ListIdentifiers o ListRecords o è un GetRecord invia all'Harvester client:
 1. L'**OAI PMHeader** dal parametro **ExtGASoaiSysTable/OAI PMHeader**
 2. Il **ResponseDateHeader** dal parametro **ExtGASoaiSysTable/ResponseDateHeader**
 3. L'**OAI time** mediante l'utility **ExtGASoaiUtility/oaiTime**
 4. Il **ResponseDateTrailer** dal parametro

ExtGASoaiSysTable/ResponseDateTrailer

5. La **OAI-PMH request** mediante il servizio **ExtGASoaiWriteResults/BuildRequest**
 2. Se è un verbo **ListIdentifiers**
 1. se è il primo elemento della lista invia il **ListIdentifiersHeader** che è contenuto nel parametro **ExtGASoaiSysTable/ListIdentifiersHeader**
 2. invia l'**OAI_identifier** mediante il servizio **ExtGASoaiWriteResults/Write_OAI_identifier**
 3. Se è un verbo **ListRecords**
 1. se è il primo elemento della lista invia il **ListRecordsHeader** che è contenuto nel parametro **ExtGASoaiSysTable/ListRecordsHeader**
 2. invia l'**OAI_record** mediante il servizio **ExtGASoaiWriteResults/Write_OAI_record**
 4. Se è un verbo **GetRecord**
 1. invia il **GetRecordsHeader** che è contenuto nel parametro **ExtGASoaiSysTable/GetRecordsHeader**
 2. invia l'**OAI_record** mediante il servizio **ExtGASoaiWriteResults/Write_OAI_record**
 5. Torna al servizio chiamante.
17. **WriteIdDoesNotExists**: invia il messaggio di errore "idDoesNotExist" e torna la servizio chiamante
18. **WriteNoRecordFound**: invia il messaggio di errore "noRecordsMatch" e torna la servizio chiamante
19. **WriteBadArgument**: invia il messaggio di errore "BadArgument" e torna la servizio chiamante
20. **WriteBadResumptionToken**: invia il messaggio di errore "BadResumptionToken" e torna la servizio chiamante
21. **BuildRequest**: scrive la **OAI-PMH request** per l'Harvester client e torna la chiamante
22. **Results**: che restituisce gli identifier e i record a seconda delle richieste servite
1. Se l'Harvester client ha diritti **Denied** invia il messaggio di errore "noRecordsMatch" (**ExtGASoaiWriteResults/WriteNoRecordFound**) e torna al driver chiamante.
 2. Accede il **ResultList** file e legge i record fino al loro esaurimento
 1. Accetta di elaborare solo record che iniziano per **<Record>** e finiscono per **</Record>**
 2. Se la data del documento è maggiore o uguale a **from** e minore o uguale a **until** l'identifier/record viene inviato all'Harvester client tramite il servizio **ExtGASoaiWriteResults/WriteList**

3. Se nessun record è stato accettato
 1. Se era verbo ListIdentifiers o ListRecords invia messaggio di errore "noRecordsMatch" tramite il servizio **ExtGASoaiWriteResults/WriteNoRecordFound** e torna al driver chiamante.
 2. Se era verbo GetRecord invia messaggio di errore "idDoesNotExist" tramite il servizio **ExtGASoaiWriteResults/WriteIdDoesNotExists** e torna al driver chiamante.
4. Altrimenti se il verbo è ListIdentifiers
 1. Se ci sono ancora identifiers da inviare scrive il **resumptionToken** tramite il servizio **ExtGASoaiWriteResults/WriteResumptionToken**
 2. Altrimenti se la HttpServletRequest contiene un **resumptionToken** scrive un **EmptyresumptionToken** tramite il servizio **ExtGASoaiWriteResults/WriteEmptyResumptionToken**
 3. Scrive un **ExtGASoaiSysTable/ListIdentifiersTrailer** che è definito dal parametro avente lo stesso nome
5. Altrimenti se il verbo è ListRecords
 1. Se ci sono ancora record da inviare scrive il **resumptionToken** tramite il servizio **ExtGASoaiWriteResults/WriteResumptionToken**
 2. Altrimenti se la HttpServletRequest contiene un **resumptionToken** scrive un **EmptyResumptionToken** tramite il servizio **ExtGASoaiWriteResults/WriteEmptyResumptionToken**
 3. Scrive un **ExtGASoaiSysTable/ListRecordsTrailer** che è definito dal parametro avente lo stesso nome
6. Altrimenti se il verbo è GetRecord
 1. Scrive un **ExtGASoaiSysTable/GetRecordTrailer** che è definito dal parametro avente lo stesso nome.
7. Invia l'**ExtGASoaiSysTable/OAIPMHtrailer** che è contenuto nel parametro avente lo stesso nome e definito nella OAI-PMH System Table
8. Torna al servizio chiamante.

1.2. Le Utility

Le Utility sono raccolte in una unica classe chiamata **ExtGASoai.Util** e vengono usate per una serie di operazioni specifiche a supporto dei Servizi e dei Driver,

1.2.0 ExtGASoai.Util.java

Di seguito verranno elencate le utility senza entrare nel merito della loro particolare funzione. Quelle senza commento sono, per il momento obsolete.

1. **ParseQueryString**
2. **ResetOurServer**
3. **GetResumptionTokenFileName** Calcola un nome univoco per il ResumptionToken File
4. **GetFileName** Calcola un nome univoco per un particolare specifico file
5. **DeleteResumptionFiles** cancella i resumptionToken file expired appartenuti a una specifica sessione.
6. **DeleteFile** cancella un generico file.
7. **LogRequest** scrive nel file di log la OAI-PMHrequest
8. **r_Host** verifica se l'Harvester client è rispettivamente allowed, denied e restricted e prepara una stringa con l'address e i suoi diritti scritti in maniera esplicita (da usare nel log del sistema)
9. **HostRights** verifica l'IP address dell'Harvester client confrontandolo con la lista degli AllowedHosts (**ExtGASoaiSysTable/AllowedHosts**) e DeniedHosts (**ExtGASoaiSysTable/DeniededHosts**) se non appartiene a nessuna delle 2 liste il client è restricted.
10. **HostAllowed** verifica se l'IP address dell'Harvester client è contenuto nella lista **ExtGASoaiSysTable/AllowedHosts**
11. **HostDenied** verifica se l'IP address dell'Harvester client è contenuto nella lista **ExtGASoaiSysTable/DeniedHosts**
12. **IPAddressAllowed** verifica l'IP address dell'Harvester client contro il mapping di definizione degli address: [0-9]\.[0-9]\.[0-9]-[0-9]\.[0-9]-[0-9]

13. **ExistsSet** verifica l'esistenza nella definizione dei set **ExtGASoaiSysTable/SetDefinition** il set specificato in un **ListIdentifiers**, **ListRecords** e **GetRecord** via identifier
 1. Se il Set non è definito invia il messaggio di errore "noSetHierarchy"
 2. Torna al servizio chiamante

14. **ExistsMetadataPrefix** verifica l'esistenza nella definizione dei metadataFormat **ExtGASoaiSysTable/ListMetadataFormats** il metadataPrefix specificato in un **ListIdentifiers**, **ListRecords** e **GetRecord**
 1. Se il metadataPrefix non è definito invia il messaggio di errore "cannotDisseminateFormat"
 2. Torna al servizio chiamante

15. **oaiTime** produce la data corrente secondo l'OAI-PMH protocol nel formato yyyy-MM-dd'T'HH:mm:ssZ

16. **StringTranslate** applica una translate table a una stringa di caratteri

17. **WriteHTMLline** invia all'Harvester client i risultati in formato XML compatibile

18. **LoadTable** carica una tabella di coppie di valori in un hashtable.

19. **localhost** ritorna il nome a dominio della URL ricevuta

20. **GetField**

21. **WriteLog** scrive un record informativo sul file di log dell'applicazione. Il nome del logfile è contenuto in **ExtGASoaiSysTable/ExtGASoaiLogFile** e assume, di default, i seguenti valori:
 1. /tmp/ExtGASoaiLogFile per ambiente unix
 2. C:/TEMP/ExtGASoaiLogFile per ambiente Windows

22. **IsDigit** verifica che la stringa sia numerica

23. **IsLetter** verifica che la stringa contenga solo lettere dell'alfabeto e numeri

24. **IsDate** verifica che la stringa sia una data in formato short gg mm aaaa per l'Italia

25. **parseName**

1.3. L'accesso alle API CDS/ISIS

Le API CDS/ISIS sono utilizzate dal sistema per accedere alle informazioni contenute nel database CDS/ISIS. In realtà esiste un unico applicativo che interfaccia il database ed è il programma `wwwisis` che viene distribuito free dall'Unesco (www.unesco.org) e da BIREME (www.bireme.br). Il programma può ricevere in input il nome di un file che contiene i parametri da utilizzare per l'accesso al database e produrre i relativi risultati sul file di output. L'attivazione di `wwwisis` viene eseguita dal servizio **AccessIsis.java**.

Complessivamente il programma `wwwisis` viene attivato per eseguire due tipi di operazione:

1. L'operazione **search** per selezionare i record che soddisfano la richiesta (**IsisSearchAPI**).
2. L'operazione **get** per acquisire il contenuto del record e inviarne il contenuto, in accordo con il **metadataPrefix** selezionato, all'Harvester client (**IsisGetAPI**).

In realtà il servizio **AccessIsis** attiva una chiamata a una applicazione esterna definita in **ExtGASoaiSysTable/ExtGASoaiProgramName** passando come parametro:

```
in=<filename>
```

Nello standard è:

```
wwwisis in=<filename>
```

Al ritorno dalla chiamata i risultati sono resi disponibili in un file il cui nome e la locazione sono noti al programma chiamante.

Genericamente è possibile chiamare un generico script che a sua volta può eseguire una serie di operazioni prima e dopo la chiamata alle API CDS/ISIS. Di seguito un esempio attivato dalla chiamata

```
access_isis.sh in=<filename>

#!/bin/bash
# this is a script example to change results after CDS/ISIS access
# input is access_isis.sh in=<filename>
# echoes parmvalue
echo $1
# execute ISIS query
/export/home/metaopac/mpisa/BiremePrograms/bin/wwwisis $1
# get input file name and echoes it
f=${1:3}
echo $f
# load parm with pft name and echoes it
parm=`grep pft= $f`
echo $parm
# if pft name is pft=@cnr_li.pft execute external program
$if [ $parm == pft=@cnr_li.pft ]
```

```
echo -e "executing external program"
$endif
exit 0
```

Di seguito sono analizzate le chiamate alle API generate a fronte di:

1. ListIdentifiers

verb=ListIdentifiers&from=2006-12-31&metadataPrefix=oai_dc&set=IENIGE

1. **IsisSearchAPI** per selezionare il numero massimo di record che soddisfa la richiesta:

```
db=cnr
bool=( ( 2006$(/5) or 2007$(/5) or 2008$(/5) or 2009$(/5) or 2010$(/5) or 2011$(/5) or 2012$(/5) or
2013$(/5) or 2014$(/5) or 2015$(/5) ) AND ap/(904) and IENIGE/(900) )
h1=1
h2=1
prolog=@prolog.cnr
epilog=@epilog.cnr
pft=mhl,v1002,/
cipar=/export/home/metaopac/mpisa/oai_cnrpubConf/cnr.txt
out=/tmp/ExtGASoai.1183710556093.0.query.src.out
```

che produce

```
<Prolog>
CNR/Pisa
</Prolog>
775
```

2. **IsisSearchAPI** per selezionare la **ResultList** dei record che soddisfano la richiesta:

```
db=cnr
bool=( ( 2006$(/5) or 2007$(/5) or 2008$(/5) or 2009$(/5) or 2010$(/5) or 2011$(/5) or 2012$(/5) or
2013$(/5) or 2014$(/5) or 2015$(/5) ) AND ap/(904) and IENIGE/(900) )
h1=1
h2=775
prolog=@prolog.cnr
epilog=@epilog.cnr
gizmo=lat2html
freq=@Sdate.pft
pft=@cnr_li.pft
cipar=/export/home/metaopac/mpisa/oai_cnrpubConf/cnr.txt
out=/tmp/ExtGASoai.1183710556093.0.out
```

che produce

```
<Prolog>
CNR/Pisa
</Prolog>
<NumberOfSelectedRecords>
775:775
</NumberOfSelectedRecords>
<Start>
<Record>;20060109;Metapub:cnr.ieni/cnr.ieni.ge/1975-A0-001;000481;IENIGE,AP;</Record>
<Record>;20060109;Metapub:cnr.ieni/cnr.ieni.ge/1991-A0-002;000482;IENIGE,AP;</Record>
.....
```

```
<Record>;20070606;Metapub:cnr.ieni/cnr.ieni.ge/2007-PR-002;000962;IENIGE,AP;</Record>
</Start>
```

da cui poi, applicando la selezione per data viene inviata la lista dei primi
(**ExtGASoaiSysTable/MaxObjectsToBeDisplayed**) **identifier**

2. GetRecord

```
verb=GetRecord&metadataPrefix=oai_dc&identifier=Metapub%3Acnr.ieni%2Fcnr.ieni.ge%2F2006-A0-021
```

1. IsisSearchAPI per selezionare il record che soddisfa la richiesta:

```
db=cnr
bool=cnr/(1) $ ieni/(1) $ cnr/(1) $ ieni/(1) $ ge/(1) $ 2006/(1) $ A0/(1) $ 021/(1)
h1=1
h2=1
epilog=@epilog.cnr
prolog=@epilog.cnr
gizmo=lat2html
freq=@Sdate.pft
pft=@cnr_li.pft
cipar=/export/home/metaopac/mpisa/oai_cnrpubConf/cnr.txt
out=/tmp/ExtGASoai.1183710629457.0.out
```

che produce una **ResultList** di un unico elemento

```
<NumberOfSelectedRecords>
1:1
</NumberOfSelectedRecords>
<Start>
<Record>;20070426;Metapub:cnr.ieni/cnr.ieni.ge/2006-A0-021;000444;IENIGE,AP;</Record>
</Start>
```

2. IsisGetAPI per accedere tramite **ResultList** via Master File Record Number al contenuto del record stesso:

```
db=cnr
from=000444
count=1
gizmo=lat2html
pft=@oai_dc.pft
cipar=/export/home/metaopac/mpisa/oai_cnrpubConf/cnr.txt
out=/tmp/ExtGASoai.1183710629576.0.out
```

che produce il record da inviare all'Harvester client in formato oai_dc

```
<oai_dc:dc
  xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
  http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
  <dc:title>Dynamic Capillary Pressure Measurements in the short Time Range by applying a fast
  growing Drop Technique</dc:title>
  <dc:creator>Makievski Alexander</dc:creator>
  <dc:creator>Kr&#228;gel J&#252;rgen</dc:creator>
  <dc:creator>Pandolfini Piero</dc:creator>
  <dc:creator>Loglio Giuseppe</dc:creator>
  .....
  .....
```

<dc:identifier><http://puma.isti.cnr.it/linkdoc.php?icode=2006-A0-021&authority=cnr.ieni.ge&collection=cnr.ieni&langver=en></dc:identifier>
</oai_dc:dc>

3. ListRecords

verb=ListRecords&from=2006-12-31&metadataPrefix=oai_dc&set=IENIGE

1. **IsisSearchAPI** per selezionare il numero massimo di record che soddisfa la richiesta (come per **ListIdentifier IsisSearchAPI 1**).
2. **IsisSearchAPI** per selezionare la **ResultList** dei record che soddisfano la richiesta (come per **ListIdentifier IsisSearchAPI 2**), da cui poi, applicando la selezione per data vengono accettati i primi record (**ExtGASoaiSysTable/MaxObjectsToBeDisplayed**) da inviare all'Harvester client.
3. **IsisGetAPI** per accedere tramite **ResultList** via Master File Record Number al contenuto dei (**ExtGASoaiSysTable/MaxObjectsToBeDisplayed**) record da inviare (come per **GetRecord IsisGetAPI 2**)

Per la comprensione dei parametri delle **IsisSearchAPI** e delle **IsisGetAPI** si faccia riferimento a:

1. BIREME - Centro Latino-Americano e do Caribe de Informação em Ciências da Saúde - WWWISIS: a world-wide web server for ISIS-databases Version 3.0, URL <http://leonardo.isti.cnr.it/metaopac/wwwisis.pdf>
2. Giuseppe A. Romano - ExtGas/IsisGAS revisione 2.0: manuale di installazione e d'uso - ISTI-B4-16, 2006 URL <http://leonardo.isti.cnr.it/metaopac/ExtGASOmoDbInstall.pdf>

1.4. Le definizioni del sistema

Le definizioni sono contenute nel modulo **ExtGASoaiSysTable.java** e servono per configurare il sistema in modo da avere più gateway OAI-PMH che interfacciano più database e/o più gateway OAI-PMH che interfacciano, in modo diverso, lo stesso database.

Nel contesto di questo manuale le definizioni non verranno descritte. Verranno, invece, descritte e discusse nel manuale di installazione e configurazione del software ExtGASoai.

Si rimanda il lettore a questo manuale o al modulo **ExtGASoaiSysTable.java** tenendo conto che ogni definizione è in realtà una piccolissima utility che scarica da una hashtable caricata in memoria il relativo parametro. La hashtable è caricata dinamicamente dall'OAI-PMH System Table che descrive la specifica applicazione. Ognuna di queste utility può avere o non avere definito un valore; e nel caso non lo avesse ritorna un default-value.

In tutti i casi molti di questi parametri sono stati menzionati nei paragrafi precedenti.

Bibliografia

1. Giuseppe A. Romano - Isis/GAS : un sistema multi piattaforma per l'accesso a basi di dati testuali via Internet - ISTI-B4-28, 2003 - URL: <http://leonardo.isti.cnr.it/metaopac/IsisGas.pdf>
2. Giuseppe A. Romano - Isis/GAS : manuale di installazione e d'uso - ISTI-B4-31, 2003 - URL: <http://leonardo.isti.cnr.it/metaopac/IsisGasInstall.pdf>
3. Giuseppe A. Romano - Il MetaOpac Pisano: l'organizzazione logica e il mapping fisico - ISTI-B4-02, 2003 - URL: <http://leonardo.isti.cnr.it/metaopac/MetaOpacLogic.pdf>
4. Giuseppe A. Romano - Il MetaOpac Pisano: descrizione tecnica e procedure di recovery - ISTI-B4-30, 2003 - URL: <http://leonardo.isti.cnr.it/metaopac/MetaopacPisano.pdf>
5. Giuseppe A. Romano - Il metaopac: controllo e recovery del sistema - ISTI-B4-07, 2004 - URL: http://leonardo.isti.cnr.it/metaopac/IsisGAS_BriefMan.pdf
6. Biagioni S., Giannini S., Giuseppe A. Romano - Il Metaopac del CNR di Pisa e la sua collocazione nel Metaopac pisano - ISTI-B4-14, 2004 - URL: <http://leonardo.isti.cnr.it/metaopac/OpacCNR.pdf>
7. Pierlugi Niccolai, Enrico Venturini - TECA: Applicazione di CDS/ISIS per i dati bibliografici - Titivillus , Firenze 1993
8. Stefania Biagioni, Carlo Carlesi, Daniela Musa - Sistema Atlas: catalogazione derivata da registrazioni in formato UNIMARC - IEI-B4-30 : Istituto di Elaborazione della Informazione, settembre 1994
9. CDS/ISIS - software, tools, descriptions - URL: http://portal.unesco.org/ci/ev.php?URL_ID=2071&URL_DO=DO_TOPIC&URL_SECTION=201&reload=1082372136&PHPSESSID=7462f4450cf4d320877976df1ae94292
10. BIREME - CISIS reference manual - URL: <http://productos.bvsalud.org/html/en/home.html>
11. BIREME - wwwisis version 3.0 - URL: <http://productos.bvsalud.org/html/en/home.html>
12. LIBERO - software - URL: <http://www.infologic.it/ita/libero.htm>
13. Antonio Scolari - Unimarc, Roma, AIB, 2000 Enciclopedia Tascabile ETIS
14. Giuseppe A. Romano - Il MetaOpacPisano : organizzazione, controllo e recovery del sistema per l'anno 2005 - ISTI-B4-17,2005 URL: http://leonardo.isti.cnr.it/metaopac/MOP_BriefMan.pdf
15. Giuseppe A. Romano - ExtGAS un sistema multipiattaforma per l'accesso ai cataloghi di sistemi biblioteconomici eterogenei - ISTI-B4-15,2005 - URL: <http://leonardo.isti.cnr.it/metaopac/ExtGas.pdf>

16. BIREME - Centro Latino-Americano e do Caribe de Informação em Ciências da Saúde - WWWISIS: a world-wide web server for ISIS-databases Version 3.0, URL <http://leonardo.isti.cnr.it/metaopac/wwwisis.pdf>
17. Giuseppe A. Romano - ExtGAS manuale di installazione e d'uso - ISTI-B4-16,2005 - URL: <http://leonardo.isti.cnr.it/metaopac/ExtGasInstall.pdf>
18. Giuseppe A. Romano - ExtGas/IsisGAS revisione 2.0: manuale di installazione e d'uso - ISTI-B4-16, 2006 URL: <http://leonardo.isti.cnr.it/metaopac/ExtGASOmoDbInstall.pdf>