

Da wwwisis 3.0 a wxis (wwwisis 5.0):
una nuova API per ExtGAS

Giuseppe A. Romano
(giuseppe.romano@isti.cnr.it)

Istituto di Scienza e Tecnologie della Informazione

giugno 2008

Indice

Introduzione	p.	5
Le funzionalità di ExtGAS/IsisGAS	p.	5
Le API wwwisis e wxis	p.	7
Da wwwisis a wxisi	p.	9
La funzione search	p.	9
La funzione get	p.	18
La funzione browse	p.	23
Il gateway wwwisis > wxis	p.	27
lo sviluppo del gateway wwwisis > wxis	p.	34
la distribuzione e l'installazione gateway	p.	35
Bibliografia	p.	36

Introduzione

Le funzionalità di ExtGAS/IsisGAS

Da un punto di vista generale ExtGAS accede a cataloghi collocati su rete IP sia che essi siano omogenei, cioè che siano mappati su database CDS/ISIS, sia che siano memorizzati su altri Database System: database eterogenei.

In questo contesto siamo interessati ai database omogenei e quindi a quella parte di ExtGAS denominata IsisGAS.

Il sistema ExtGAS/IsisGAS utilizza, per l'accesso ai database CDS/ISIS la API wwwisis/wwwi32.exe nella versione 3.0 distribuita free dall'UNESCO.

La API wwwisis versione 3.0, sviluppato da www.bireme.br, contiene un errore (e, non è detto che sia un errore reale quanto un limite implicito!).

Al momento della generazione del database tramite l'applicativo mx, con l'opzione **[<dbn>]/mstxl=1/4** inserita in un **cipar file** conosciuto come variabile di ambiente **cipar**, (mx program con export cipar=<ciparfilename>) è possibile utilizzare/caricare database CDS/ISIS che si espandano fino a un massimo di 1,2,4 gigabytes. Il seguente è un esempio:

```
date
export cipar=cipar.file
/export/home/metaopac/mpisa/BiremePrograms/bin/mx iso=upi01.iso create=unipi -all now tell=100000
date
```

dove cipar.file contiene:

mstxl=4

che sta a indicare un massimo di 4 gigabytes.

A questo punto è abbastanza evidente che il numero di record che possono essere memorizzati sul database può superare il 1000000. La API wwwisis riesce a gestire per il search un massimo di 999999 record (mfn=999999). Le rimanenti operazioni riescono a elaborare mfn>999999.

Naturalmente il limite di 999.999 record è un limite teorico. E' sufficiente trasformare il database con numero di record maggiore di 999999 in due o più database componenti. Volendo, però, ovviare a questo limite è possibile utilizzare per il search la API wwwisis release 4/5, noto anche come wxis, soprattutto perchè ne è stato annunciato il rilascio free entro la fine dell'anno corrente (2008).

Il programma wxis non è una vera e propria API come wwwisis ma una interfaccia XML like verso un database CDS/ISIS.

Per non modificare la logica di costruzione della parte IsisGAS di ExtGAS è stato necessario creare una interfaccia software tra wwwisis3.0 e wxis in modo che i comandi noti alla API wwwisis3.0 potessero essere tradotti da un insieme di comandi wxis tali da

ottenere degli script (IsisScript) da eseguire con il comando:

wxis IsisScript=IsisScriptFileName

in modo da utilizzare sempre il meccanismo della API wwwisis3.0 per l'accesso al database ma tramite il programma wxis ottenere gli stessi risultati che IsisGAS si aspetta dall'esecuzione della API wwwisis3.0.

Si tenga presente che i formati di restituzione per l'ambiente **search** di IsisGAS hanno, nel nostro contesto, alcuni requirements particolari dovuti alla gestione dei comandi **back** e **more** per navigare nei sottoinsiemi dei documenti da presentare. Parte di tale formato è il seguente:

```
mhl,if val(v1001) = val(v1031) then '<NumberOfSelectedRecords>',/,F((val(v1032)-
val(v1031)+1),3,0),',',v1002,/, '</NumberOfSelectedRecords>',/,
.....
.....fi,
.....
```

dove, dato **h1** e **h2** i numeri di documenti selezionati (**h1**=start, **h2**=end) come nell'esempio:

```
bool=( romano/(70,90,110) )
h1=5001
h2=6000
pft=@unipiTiLns.pft
```

dove vogliamo ottenere la lista dei documenti, selezionati per (romano/(70,90,110)), compresi tra il 5001esimo e il 6000esimo.

Si tenga inoltre presente che wwwisis3.0 rende disponibile durante la fase di search/retrieve una serie di virtual tag Isis che contengono i più disparati valori. Nel nostro contesto siamo interessati ai seguenti virtual tag e relativi valori:

- v1001** è il numero di documento attivo, che soddisfa la query, nella specifica operazione di search
- v1002** è il numero totale di docs selezionati nella specifica operazione di search
- v1021** contiene la CDS/ISIS query per la specifica operazione di search
- v1031** è il numero iniziale di docs che soddisfa la query nella specifica operazione di search
- v1032** è il numero totale di docs che soddisfa la query
- v1071** contiene il Print Format utilizzato per la stampa dei risultati del search
- v1091** contiene l'exit code dell'operazione

Per cui nel caso della richiesta (romano/(70,90,110)) supposto esistano 5115 docs che all'inizio soddisfano completamente la query, nel nostro caso (limitato ai documenti compresi tra 5001 e 600) si hanno i seguenti valori dei tag:

```
v1001 vale 5001
v1002 vale 115
v1031 vale 5001
```

v1032 vale 5115

che producono le seguenti linee di stampa:

```
<NumberOfSelectedRecords>  
115:115  
</NumberOfSelectedRecords>
```

I due valori sono utilizzati dal servlet ExtGAS per pilotare e costruire i bottoni associati a comandi **more** e **back**.

L'exit code della operazione (virtual tag v1091) viene testato mediante l'epilog.pft che viene eseguito da **wwwisis30** al termine di tutte le operazioni. Il seguente è un esempio di epilog format:

```
mhl,if val(v1091) = 7 then,  
'<error-msg>',/, '0 record retrieved',/, '</error-msg>',/,  
else,  
if val(v1091) = 3 then,  
'<error-msg>',/, '0 record retrieved: search syntax error',/, '</error-msg>',/,  
else,  
if val(v1091) = 4 then,  
'<error-msg>',/, '0 record retrieved: search execution error',/, '</error-msg>',/,  
else,  
if val(v1091) = 5 then,  
'<error-msg>',/, '0 record printed: format error',/, '</error-msg>',/,  
else '</Start>',/,  
fi,  
fi,  
fi,  
fi,/,
```

Invece **wxis**, tramite il suo IsisScript Language, mette a disposizione i seguenti parametri (evidenziati in grassetto) che possono essere caricati a scelta nei tag preferiti (nel caso particolare vengono mappati gli stessi tag di wwwisis3.0) come:

```
<!-- index of current <loop> execution -->  
<field action=define tag=1001>Isis_Current</field>  
  
<!-- total the times possible for the <loop> -->  
<field action=define tag=1002>Isis_Total</field>  
  
<!-- parameter from for the <loop> -->  
<field action=define tag=1031>Isis_From</field>  
  
<!-- storage of task execution status -->  
<field action=define tag=1091>Isis_Status</field>  
  
<!-- error pointer in search -->  
<field action=define tag=1092>Isis_ErrorInfo</field>  
  
<!-- total list items -->  
<field action=define tag=1103>Isis_Items</field>
```

```
<!-- list item -->
<field action=define tag=1100>Isis_Item</field>
```

e tramite la funzione:

```
<field action=replace tag=nnnn><pft>pft_specs_string</pft></field>
```

definire un virtual field e assegnarne un il valore,

(si riferisca a “IsisScript Language Reference”, versione 1.1, San Paolo (Brasile) – 2006, pp. 16-17)

Sarà, quindi, necessario procedere all'implementazione dei tag descritti per **wwwisis 3.0** e procedere al loro caricamento.

Per esempio nel caso di search senza ordinamento sarà necessario definire e caricare:

```
<field action=replace tag=1032><pft>if val(v1002) > val(v1031)+(h2-h1+1)-1 then 'h2'
else F(val(v1002),1,0),fi</pft></field>
```

poiché per IsisGAS è h1=start e h2=end (nel contesto wwwisis3.0), mentre per wxis è from=h1 e count=h2-h1+1.

e

```
<field action=replace tag=1002><pft>F((val(v1032)-val(v1031)+1),1,0)</pft></field>
```

in modo da poter eseguire la frazione di formato:

```
mhl,if val(v1001) = val(v1031) then '<NumberOfSelectedRecords>',/,F((val(v1032)-
val(v1031)+1),3,0),',',v1002,/,',</NumberOfSelectedRecords>',/,
```

Mentre nel caso di search con ordinamento è:

```
<field action=replace tag=1032><pft>if val(v1002) > val(v1031)+(h2-h1+1)-1 then 'h2'
else F(val(v1002),1,0),fi</pft></field>
```

essendo come sopra...e.....

```
<!-- total list items -->
<field action=define tag=1103>Isis_Items</field>
```

in quanto il search con ordinamento deve essere costruito utilizzando l'IsisScript language come: search + load sorted list + print sorted list.

A queste note si aggiunga che la gestione dell'epilog.pft, caratteristica di wwwisis3.0, è completamente mutata in ambiente wxis, per cui non è più possibile utilizzare l'epilog.pft derivabile da wwwisis3.0. Le funzionalità dell'epilog andranno di volta in volta definite nel contesto dell'IsisScript cui si riferiscono.

Inoltre a seconda delle implementazioni di ExtGAS/IsisGAS è utile rendere sempre disponibili due virtual tag:

- I. Che mantenga nel virtual tag v1021 la booleana query utilizzata.
- II. Che mantenga in un generico virtual il nome del Print Format File che è utilizzato per la restituzione. Nel caso particolare viene utilizzato il virtual tag v3333.

Le operazioni principali eseguite da ExtGAS/IsisGAS sono relative a:.

1. **search** per la ricerca e la selezione dei documenti che soddisfano le richieste di utente.
2. **get**: per il retrieval dei records selezionati.
3. **browse** per il retrieval di insiemi di record a partire da una particolare posizione, e l'accesso e il retrieval delle chiavi di ricerca costruite e memorizzate sull'inverted file.

Da wwwisis a wxis

Di seguito verranno analizzate dal punto di vista dei due applicativi le funzioni relative all'accesso all'informazione mediante le funzionalità di:

1. search
2. get
3. browse

La funzione search

Esistono 3 condizioni particolari:

Il search per ottenere il numero massimo di record selezionati

```
db=<&dbname>
cipar=<&ciparfilename>
bool=<&booleanSearch>
h1=<&fromHits=1>
h2=<&toHits=1>
pft=@<&pft='mhl,v1002,/'>
prolog=@<&prolog>
epilog=@<&epilog>
out=<&outfile>
```

che dovrà generare un ISIS_Script come nell'esempio:

```
<IisScript name=ISIS_script_total>
<!--
Define external files and their locations
-->
<parm name=cipar><pft>
    'unipi.*=/home/metaopac/mpisa/IsisDbdir.beppe/unipi.*'/
    'prolog.unipi=/home/metaopac/mpisa/upi01Conf/prolog.unipi/'
</pft></parm>
<!--
Execute search
-->
<do task=search>
<!--
Search result output file
-->
<file action=create type=output>/tmp/WXIS_num.file</file>
<!--
Assign constant values to virtual tags
-->
<field action=define tag=1002>Iis_Total</field>
<field action=define tag=1091>Iis_Status</field>
<field action=define tag=1092>Iis_ErrorInfo</field>
```

```
<!--
```

Parameters

```
-->
```

```
<parm name=expression>c$</parm>
```

```
<parm name=from>1</parm>
```

```
<parm name=count>1</parm>
```

```
<parm name=db>unipi</parm>
```

```
<!--
```

Prolog original

```
-->
```

```
<display><pft>@prolog.unipi</pft></display>
```

```
<loop>
```

```
<!--
```

Print results

```
-->
```

```
<display><pft>v1002,</pft></display>
```

```
</loop>
```

```
<!--
```

Epilog – new epilog

```
-->
```

```
<display><pft>
```

```
    if val(v1091)>0 then '<error-msg>',/, 'Error ',v1091,' ,syntax error: ',v1092,/, '</error-  
msg>',/,
```

```
    else if val(v1002)=0 then '<error-msg>',/, 'No records found',/, '</error-msg>',/, else  
'EpilogMessage' fi,fi
```

```
</pft></display>
```

```
</do>
```

```
</IsisScript>
```

che produce risultati di questo tipo:

```
<Prolog>
```

```
Monografie Universit&agrave; di Pisa
```

```
</Prolog>
```

```
1539426
```

- Il search per ottenere i relativi record selezionati senza ordinamento

```
db=<&dbname>
gizmo=<&gizmo>
cipar=<&ciparfilename>
bool=<&booleanSearch>
h1=<&fromHits=5001>
h2=<&toHits=6000>
pft=@<&pft=unipiTiLns.pft>
prolog=@<&prolog>
epilog=@<&epilog>
out=<&outfile>
```

che dovrà generare un ISIS_Script come nell'esempio:

```
<IsisScript name=ISIS_script_docs_unsorted>
<!--
Define external file and their locations
-->
<parm name=cipar><pft>
    'unipi.*=/home/metaopac/mpisa/IsisDbdir.beppe/unipi.*'/
    'html.*=/home/metaopac/mpisa/upi01Conf/lat2html.*'/
    'unipiTiLns.pft=/home/metaopac/mpisa/upi01Conf/unipiTiLns.pft'/
    'prolog.unipi=/home/metaopac/mpisa/upi01Conf/prolog.unipi'/
</pft></parm>
<!--
Execute search
-->
<do task=search>
<!--
Search result output file
-->
<file action=create type=output>/tmp/WXIS_docs.file</file>
<!--
Assign constant values to virtual tags
-->
<field action=define tag=1001>Isis_Current</field>
<field action=define tag=1002>Isis_Total</field>
<field action=define tag=1031>Isis_From</field>
<field action=define tag=1091>Isis_Status</field>
<field action=define tag=1092>Isis_ErrorInfo</field>
<!--
Parameters
-->
<parm name=expression>( romano/(70,90,110) )</parm>
<parm name=from>5001</parm>
<parm name=count>1000</parm>
<parm name=gizmo>html</parm>
<parm name=db>unipi</parm>
<!--
```



```

<$OurConf$>&Obj=@unipiSh.pft&Opt=get&Type=Doc&Id=02026584","Doc");>Un secolo di associazionismo
nel territorio pisano, 1850-1950 : itinerario storico attraverso le pubblicazioni della Biblioteca universitaria di
Pisa / catalogo a cura di Michela Pazzagli e Giovanni Perna ; introduzione di Romano Paolo Coppini</a>
</Record>
<Record>
&nbsp;</b><input type=checkbox name="Checker" value=2028022><a
href=javascript:top.WindowOpen("<$OurServer$>?
<$OurConf$>&Obj=@unipiSh.pft&Opt=get&Type=Doc&Id=02028022","Doc");>Un' idea dell'Europa /
Romano Prodi</a>
</Record>
<Record>
.....
.....
.....
</Record>
<Record>
&nbsp;</b><input type=checkbox name="Checker" value=2136498><a
href=javascript:top.WindowOpen("<$OurServer$>?
<$OurConf$>&Obj=@unipiSh.pft&Opt=get&Type=Doc&Id=02136498","Doc");>&#233;ka- o Eka- nel
Rigveda? / candidata Tiziana Soressi ; relatore Chiar.mo Prof. Romano Lazzeroni</a>
</Record>
</Start>
<p><input type=submit value=inizia><input type=reset value=ripristina><input onclick=doCheckAll()
type=button value="seleziona tutto"></FORM>

```

- Il search per ottenere i relativi record selezionati con ordinamento

```
db=<&dbname>  
gizmo=<&gizmo>  
cipar=<&ciparfilename>  
bool=<&booleanSearch>  
h1=<&fromHits=3001>  
h2=<&toHits=4000>  
pft=@<&pft=unipiTiL.pft>  
freq=@<&freq=Sti.pft>  
prolog=@<&prolog>  
epilog=@<&epilog>  
out=<&outfile>
```

Si tenga presente che nella search con ordinamento **wwwisis3.0**:

1. esegue la ricerca,
2. carica la chiave di ordinamento generandola dal pft definito nel parametro freq,
3. esegue l'ordinamento e
4. stampa i risultati utilizzando il pft specificato nel parametro pft

Mentre per **wxis** è necessario

1. attivare la ricerca,
2. caricare una lista contenente le chiavi di sort generate tramite il parametro <list action=load type=sort...> utilizzando il formato specifico per il sort,
3. attivare un task list per scaricare la lista ordinata e
4. stampare i record secondo le specifiche contenute nel parametro specificato da pft.

Il tutto dovrà generare un ISIS_Script come nell'esempio:

```
<IisisScript name=ISIS_script_docs_sorted>  
<!--  
Define external file and their locations  
-->  
<parm name=cipar><pft>  
    'unipi.*=/home/metaopac/mpisa/IsisDbdir.beppe/unipi.*/'  
    'html.*=/home/metaopac/mpisa/upi01Conf/lat2html.*/'  
    'unipiTiL.pft=/home/metaopac/mpisa/upi01Conf/unipiTiL.pft/'  
    'Sti.pft=/home/metaopac/mpisa/upi01Conf/Sti.pft/'  
    'prolog.unipi=/home/metaopac/mpisa/upi01Conf/prolog.unipi/'  
</pft></parm>  
<!--  
Define search result output file  
-->  
<file action=create type=output>/tmp/WXIS_docs_sorted.file</file>  
<!--  
Start serach operation  
-->  
<do task=search>  
<!--  
Assign constant values to virtual tags
```

```

-->
<field action=define tag=1001>Isis_Current</field>
<field action=define tag=1002>Isis_Total</field>
<field action=define tag=1091>Isis_Status</field>
<field action=define tag=1092>Isis_ErrorInfo</field>
<!--
Define parameters
-->
<parm name=expression>( romano/(70,90,110) )</parm>
<parm name=from>3001</parm>
<parm name=count>1000</parm>
<parm name=gizmo>html</parm>
<parm name=db>unipi</parm>
<!--
Prolog original
-->
<display><pft>@prolog.unipi</pft></display>
<loop>
<!--
Load and sort
-->
<list action=load type=sort><pft>@Sti.pft,</pft></list>
</loop>
<!--
Epilog – new epilog
-->
<display><pft>
    if val(v1091)>0 then '<error-msg>',/,'Error ',v1091,' ,syntax error: ',v1092,/,'</error-
msg>',/,
    else if val(v1002)=0 then '<error-msg>',/,'No records found',/,'</error-msg>',/, fi,fi
</pft></display>
<!--
And now print sorted results
-->
</do>
<!--
Start task list
-->
<do task=list>
<!-- commento:
- from è sempre 1
- count è sempre come sopra: h2-h1+1
- nello statement successivo: val(v1031)+1000-1 then '1000'
vale esattamente: val(v1031)+(h2-h1+1)-1 then 'h2-h1+1'
virtual tag v1002 viene caricato come numero di item nella lista
    <field action=define tag=1002>Isis_Items</field>
-->
<!--
Define parameters
-->

```


La funzione get

La funzione **get** permette l'accesso al record tramite il suo MasterFileRecordNumber che rappresenta l'ordine di caricamento del record all'interno del file che lo contiene. Essa si articola in tre sottosezioni:

A - Il reperimento e il display di un ben definito record

```
db=<&dbname>
cipar=<&ciparfilename>
gizmo=<&gizmo>
from=<&fromMFN=26605>
count=<&count=1>
pft=@<&pft=>
prolog=@<&prolog>
epilog=@<&epilog>
out=<&outfile>
```

Il problema maggiore sta nella gestione dell'errore in quanto l'accesso a un particolare record potrebbe essere fatto da chiamata esterna e/o interna con numero di MFN minore o uguale a zero.

Il tutto dovrà generare un ISIS_Script come nell'esempio:

```
<IsisScript name=ISIS_script_get_docs>
<!--
Define external file and their locations
-->
<parm name=cipar><pft>
'unipi.*=/home/metaopac/mpisa/IsisDbdir.beppe/unipi.*'/
'prolog.unipi=/home/metaopac/mpisa/upi01Conf/prolog.unipi'/
'unipiSh.pft=/export/home/metaopac/mpisa/upi01Conf/unipiSh.pft'/
'html.*=/export/home/metaopac/mpisa/upi01Conf/lat2html.*'/
</pft></parm>
<!--
Execute retrieval on a mfn range
-->
<do task=mfnrange>
<!--
Get result output file
-->
<file action=create type=output>/tmp/WXIS_get_docs.file</file>
<!--
Load from mfn value into virtual tag 2000
-->
<field action=replace tag=2000><pft>'26605'</pft></field>
<!--
if tag 2000 value is > 0 continue to GOOD operation to do
-->
<flow action=jump><pft>if val(2000)>0 then 'GOOD' fi</pft></flow>
```

```

<!--
Else display Prolog original and error message... and than
-->
<display><pft>@prolog.unipi</pft></display>
<display><pft>'<error-msg>',/, 'No records found',/, '</error-msg>',/, </pft></display>
<!--
close output file and exit
-->
<file action=close type=output>/tmp/WXIS_get_docs.file</file>
<flow action=exit>0</flow>
<!--
Label GOOD
-->
<label>GOOD</label>
<!--
Assign constant values to virtual tags
-->
<field action=define tag=1001>Isis_Current</field>
<field action=define tag=1002>Isis_Total</field>
<field action=define tag=1031>Isis_From</field>
<field action=define tag=1091>Isis_Status</field>
<field action=define tag=1092>Isis_ErrorInfo</field>
<!--
Define parameters
-->
<parm name=db>unipi</parm>
<parm name=from><pft>v2000</pft></parm>
<parm name=count>1</parm>
<!-- commento:
- from è sempre calcolato ..... qui è 26605
- count è sempre > 0 ..... qui è 1
- nello statement: if val(v1002)>val(v1031)+1-1 then '26605'
  vale esattamente: val(v1031)+count-1 then 'from'
  virtual tag v1002 viene caricato come numero di item totali
-->
<!--
Prolog original
-->
<display><pft>@prolog.unipi</pft></display>
<loop>
<field action=replace tag=1032><pft>if val(v1002)>val(v1031)+1-1 then '26605' else
F(val(v1002), 1,0),fi</pft></field>
<display><pft>@unipiSh.pft</pft></display>
</loop>
<!--
Epilog new epilog
-->
<display><pft>
  if val(v1091)>0 then '<error-msg>',/, 'Error ',v1091,' ,syntax error: ',v1092',/, '</error-
msg>',/,

```

```

else if val(v1001)=0 OR val(1031)=0 then '<error-msg>',/, 'No records
found',/, '</error-msg>',/, else 'EpilogMessage' fi,fi
</pft></display>
</do>
</IisScript>

```

Che produce risultati di questo tipo

```

<Prolog>
Monografie Universit&agrave; di Pisa
</Prolog>
<DRecord>
<font face=TimesNewRoman size=2><strong>Autore/i: </strong><a
href=javascript:top.AutoWindowOpen("<$OurServer$?>
<$OurConf$&Obj=@unipiTiL0.pft,SortedBy:@Sti.pft&Opt=search&Field0=&Field1=&Field2=Concina,
Ennio", "Doc");>Concina, Ennio</a>;
</font><br>
<font face=TimesNewRoman size=2><strong>Luogo, editore, data di pubblicazione: </strong>Torino :
Einaudi, 1990</font><br>
<font face=TimesNewRoman size=2><strong>ISBN/ISSN: </strong>88-06-11837-4</font><br>
.....
.....
<font face=TimesNewRoman size=2><strong>Collocazione: </strong><ol><li>Centro bibl.: Storia delle arti,
Fondo: Storia delle arti, collocaz.: F.e. 94 bis<li>Centro bibl.: Storia delle arti, Fondo: Storia delle arti,
collocaz.: F.e. 94</ol></font>
</Drecord>

```

B - Il reperimento e il display di un insieme ordinato di record (lista di record) a partire da un ben definito MFN (from) per un numero altrettanto definito di record (count).

```

db=<&dbname>
cipar=<&ciparfilename>
gizmo=<&gizmo>
from=<&fromMFN>
count=<&count=n>
pft=@<&pft=>
prolog=@<&prolog>
epilog=@<&epilog>
out=<&outfile>

```

Che corrisponde esattamente al precedente esempio di reperimento di un unico definito record.

C - Il reperimento e il display di un insieme ordinato di record (lista di record) che corrisponde a un insieme ordinato di chiamate alla precedente sottosezione

Nel caso particolare si avrebbe una lista di record da presentare; dove la lista è costituita da un insieme di MFN separati da una virgola:

```

get=10,2,34,.....123456

```

L'operazione è abbastanza complessa perché richiede:

- di stampare una intestazione (prolog) utilizzando un print format che non produca nulla:

```
db=unipi
prolog=@unipiRTFbox.prolog.unipi
gizmo=html
from=1
count=1
pft=@unipiRTFbox.prolog.pft
cipar=/export/home/metaopac/mpisa/upi01Conf/unipi.txt
outapp=/tmp/Isis.1203520372132.0.out
```

- di stampare, uno per uno, tanti record quanto sono i richiesti; e quindi effettuare tante chiamate IsisScript name=ISIS_script_get_docs come descritto sopra:

```
db=unipi
gizmo=html
from=433
count=1
pft=@unipiRTFbox.pft
cipar=/export/home/metaopac/mpisa/upi01Conf/unipi.txt
outapp=/tmp/Isis.1203520372132.0.out
```

e

```
db=unipi
gizmo=html
from=3124
count=1
pft=@unipiRTFbox.pft
cipar=/export/home/metaopac/mpisa/upi01Conf/unipi.txt
outapp=/tmp/Isis.1203520372132.0.out
```

- di stampare un trailer di lista (epilog) utilizzando ,anche qui, un print format che non produca nulla:

```
db=unipi
epilog=@unipiRTFbox.epilog.unipi
gizmo=html
from=1
count=1
pft=@unipiRTFbox.epilog.pft
cipar=/export/home/metaopac/mpisa/upi01Conf/unipi.txt
outapp=/tmp/Isis.1203520372132.0.out
```

I file di formato sono caricati come segue:

unipiRTFbox.prolog.unipi

```
'<Prolog>',/,
'Monografie Universit&agrave; di Pisa',/,
'</Prolog>',/, '<Start>',/,
```

unipiRTFbox.prolog.pft

```
mhl,if a(v1) then v1 fi,
```

che afferma che in assenza del campo tag1 se ne stampi il contenuto!

unipiRTFbox.epilog.unipi

```
mhl,'</Start>',/,
```

unipiRTFbox.epilog.pft

```
mhl,if a(v1) then v1 fi,
```

Si noti che le quattro operazioni producono il loro out sempre sullo stesso file che è definito con l'opzione **outapp** (append). Il risultato finale è il seguente:

```
<Prolog>
Monografie Universit&agrave; di Pisa
</Prolog>
<Start>
<Record>
<font face=TimesNewRoman size=2><strong>Titolo: </strong>1 Alcesti, Medea, Ippolito, Gli eraclidi, Ecuba,
Andromaca, Le supplici, Eracle, Le troiane, Elettra, Elena / Euripide ; a cura di Filippo Maria
Pontani</font><br>
<font face=TimesNewRoman size=2><strong>Opera: </strong><a
href=javascript:top.AutoWindowOpen("<$OurServer$?
<$OurConf$&Obj=@unipiTiL0.pft,SortedBy:@Sti.pft&Opt=search&Field0=id=000005436","Doc");>Tutte le
tragedie : edizioni integrali</a></font><br>
<font face=TimesNewRoman size=2><strong>Luogo, editore, data di pubblicazione: </strong>Roma :
Newton, 1994</font><br>
<font face=TimesNewRoman size=2><strong>Edizione: </strong>2. ed.</font><br>
.....
.....
</Record>
<Record>
<font face=TimesNewRoman size=2><strong>Titolo: </strong>1 Database management</font><br><font
face=TimesNewRoman size=2><strong>Opera: </strong><a
href=javascript:top.AutoWindowOpen("<$OurServer$?
<$OurConf$&Obj=@unipiTiL0.pft,SortedBy:@Sti.pft&Opt=search&Field0=id=000136914","Doc");>Principle
s of database and knowledge-base systems</a></font><br><font face=TimesNewRoman
size=2><strong>Luogo, editore, data di pubblicazione: </strong>Rockville : Computer Science Press, 1988</
font><br><font face=TimesNewRoman size=2><strong>Descrizione fisica: </strong>xi, 631 p., 24
cm</font><br><font face=TimesNewRoman size=2><strong>ISBN/ISSN: </strong>0-7167-8158-1</font>
.....
.....
<font face=TimesNewRoman size=2><strong>Collocazione: </strong><ol><li>Centro bibl.: Matematica
informatica e fisica, Fondo: MIFIN, collocaz.: H.2.0 u41/2 I 1INF<li>Centro bibl.: Matematica informatica e
fisica, Fondo: MIFIN, collocaz.: H.2.0 u41/2 I 2INF</ol></font>
</Record>
</Start>
```

La funzione browse

La funzione browse permette l'accesso al record tramite la selezione di una sua chiave di identificazione caricata nell'inverted file, o tramite la sua identificazione oggettiva MasterFileRecordNumber (MFN). Naturalmente, in maniera riduttiva, è possibile navigare nell'inverted file e nel file che contiene i documenti (master file), in avanti e indietro, e quindi accedere, se l'inverted file è costruito con criteri di classificazione, a classi di chiavi. Nella stessa maniera è possibile costruire database di supporto estraendo dai record particolari valori significativi (per esempio i **titoli**, gli **autori**, ecc..). Fondamentalmente la funzione browse si risolve in due operazioni successive:

1. Selezione del primo record tramite una sua chiave di identificazione
2. Accesso al record e ai suoi successivi

Di seguito verrà descritto il punto 1, il punto 2 non richiede commenti essendo una chiamata o sequenza di chiamate di `get:docs` utilizzando lo script `ISIS_script_get_docs`

Selezione del record tramite una sua chiave di identificazione

La selezione può avvenire a partire da:

- Un particolare record, per continuare fino alla fine del database:

```
db=unipi
prolog=@prolog.unipi
epilog=@epilog.unipi
gizmo=html
from=1222
count=100
pft=@unipiShproof.pft
cipar=/export/home/metaopac/mpisa/upi01Conf/unipi.txt
out=/tmp/Isis.1211359456227.0.out
```

e in questo caso si verifica esattamente la condizione della **funzione get** utilizzando lo script `ISIS_script_get_docs` con **count** uguale a quanto richiesto dall'utente. E quindi si rimanda al capitolo precedente.

- A partire dall'inverted file del database stesso

```
db=unipi
prolog=@prolog.unipi
epilog=@epilog.unipi
gizmo=html
k1=a
k2=z
count=100
pft=@Inv.pft
cipar=/export/home/metaopac/mpisa/upi01Conf/unipi.txt
```

out=/tmp/Isis.1211359470046.0.out

e in questo caso viene utilizzato un particolare print format, Inv.pft, che è così definito:

```
mhl,if val(v1001) = 1 then '<SubProlog>',/,'Inverted file',/,'</SubProlog>',/,'<Start>',/,'fi,
if a(v1) then
'<NumberOfSelectedIndexRecords>',/,'v1031',':',v1032,':',v1002,'/,'</NumberOfSelectedIndexRecords
>',/,'else,,
mhl,'<Record>',/,'
mhl,'&nbsp;</b><a href=javascript:top.WindowOpen("<$OurServer$>?<$OurConf$>
&Obj=@unipiTiL.pft&Opt=search&Field0=!,V1,!","Doc");>',V1,'(',V2,')</a>',/,'
'</Record>',/,'
fi,
```

e utilizza il virtual tag v1 che contiene la stringa selezionata e il virtual tag v2 che contiene il numero dei posting della stringa stessa nel database.

- A partire dall'inverted file di database esterno utilizzato come database di appoggio per la costruzione di query complesse:
 - Con una prima richiesta per posizionarsi sull'inverted file

```
db=aixau
gizmo=html
k1=me
posting
count=1
pft=@unipiAix.pft
cipar=/export/home/metaopac/mpisa/isti_piConf/cnr.txt
out=/tmp/Isis.1211357337294.0.out.aix.out
```

dove unipiAix.pft è utilizzato per posizionarsi sul record di inverted: v3^m/, ; dove il sottocampo ^m del virtual tag3 contiene il Master File Record Number del record.

- E una seconda per leggere il record e eventualmente i successivi:

```
db=aixau
prolog=@prolog.cnr-pub
epilog=@epilog.cnr-pub
from=2433
count=50
pft=@unipiAixAu.pft
cipar=/export/home/metaopac/mpisa/isti_piConf/cnr.txt
out=/tmp/Isis.1211357337294.0.out
```

eseguendo un loop di get_docs utilizzando lo script ISIS_script_get_docs con **count** uguale a quanto richiesto dall'utente.

In conclusione per il posizionamento è definito il seguente prototipo di IsisScript


```

<IsisScript name=ISIS_script_get_posting>
<!--
Define external files and their locations
-->
<parm name=cipar><pft>
'unipi.*=/home/metaopac/mpisa/IsisDbdir.beppe/unipi.*'/
'prolog.unipi=/home/metaopac/mpisa/upi01Conf/prolog.unipi/'
'Inv.pft=/export/home/metaopac/mpisa/upi01Conf/Inv.pft/'
</pft></parm>
<!--
Locate key range
-->
<do task=keyrange>
<!--
Define output file and its location
-->
<file action=create type=output>/tmp/WXIS_inverted.file</file>
<!--
Define and load virtual tags
-->
<field action=define tag=1001>Isis_Current</field>
<field action=define tag=1002>Isis_Total</field>
<field action=define tag=1031>Isis_From</field>
<field action=define tag=1091>Isis_Status</field>
<field action=define tag=1092>Isis_ErrorInfo</field>
<field action=define tag=1>Isis_Key</field>
<field action=define tag=2>Isis_Postings</field>
<field action=define tag=3>Isis_Posting</field>
<!--
Load keyrange parm location
-->
<parm name=db>unipi</parm>
<parm name=from>a</parm>
<parm name=to>z</parm>
<parm name=count>10</parm>
<parm name="posting">1</pft></parm>
<!--
Standard prolog
-->
<display><pft>@prolog.unipi</pft></display>
<!--
Loop until count
-->
<loop>
<!--
Display data using Inv.pft
-->
<display><pft>@Inv.pft</pft></display>
</loop>
|<!--

```

Epilog: write error messages if required

-->

```
<display><pft> if val(v1001)=0 then '<error-msg>',/, '0 record retrieved',/, '</error-msg>' else  
'<NumberOfSelectedIndexRecords>',/, v1031, ' :', v1, ' :', v1001,/, '</NumberOfSelectedIndexR  
ecords>' fi,/, </pft></display>  
</do>  
</IsisScript>
```

Naturalmente al variare del tipo di accesso, o sull'INVERTED FILE del database stesso o sull'INVERTED FILE del database di supporto come indice alternativo, varia la struttura del formato di stampa e, di conseguenza, dell'epilog.

Il gateway wwwisis > wxis

Il gateway **wwwisis->wxis** va costruito a partire dalla classe AccessIsis.java di ExtGAS.

Nel metodo IsisGet di AccessIsis.java:

1. verificare che venga richiesta l'attivazione del gateway (inserire in IsisSysTable.java il boolean wxisSearch (true/FALSE)).
2. attivare il metodo IsisGet della nuova classe WXISgateway con gli stessi parametri del metodo IsisGet di AccessIsis.java.

WXISgateway(IsisGet)

1. Legge il file FileQuery e imposta i parametri:
 1. &dbname = db
 2. &h1 = h1
 3. &h2 = h2
 4. &k1 = k1
 5. &k2 = k2
 6. &count = count
 7. &from = from
 8. &bool = bool
 9. &gizmo = gizmo
 10. &out = out
 11. &outapp = outapp
 12. &cipar = cipar
 13. &prolog = prolog
 14. &epilog = epilog
 15. &pft = pft
 16. &freq = freq
 17. &posting = boolean
2. Legge il file &cipar e lo carica in un array di stringhe.
3. if (&h1==1 && &h1==&h2 && &freq==null) build ISIS_Script_total
4. else if (&freq != null) build ISIS_Script_docs_sorted
5. else if (&h1>0) build ISIS_script_docs_unsorted
6. else if (&from!=null && &count!=null && &bool==null) build ISIS_script_get_docs
7. else if (&posting==true && &k1!=null && &k2==null) build ISIS_script_get_posting
8. else if (&posting==true && &k1!=null && &k2!=null) build ISIS_script_get_posting
9. Get IsisWXISpgmName from IsisSysTable
10. Execute WXIS program with IsisScript=ISIS_Script_filename as parms
11. Return to caller

WXISgateway(ISIS_Script_total)

1. ISIS_ScriptFilename=FileQuery+".wxis"
2. open ISIS_ScriptFilename and write
 1. "<IsisScript name=ISIS_script_total>"
 2. "<parm name=cipar><pft>"
 3. "GetFromCiparArray(&dbname+*.*)'"
 4. "GetFromCiparArray(&gizmo+*.*)'"
 5. if &prolog.startsWith(@) "GetFromCiparArray(&prolog.substring(1))'"
 6. if &pft.startsWith(@) "GetFromCiparArray(&pft.substring(1))'"
 7. "</pft></parm>"
 8. "<do task=search>"
 9. "<file action=create type=output>+ &out + "</file>"
 10. "<field action=define tag=1002>Isis_Total</field>"
 11. "<field action=define tag=1091>Isis_Status</field>"
 12. "<field action=define tag=1092>Isis_ErrorInfo</field>"
 13. "<parm name=expression>+ &bool + "</parm>"
 14. "<parm name=from>+ &h1 + "</parm>"
 15. "<parm name=count>+ (&h2-&h1+1) + "</parm>"
 16. "<parm name=db>+ &dbname + "</parm>"
 17. "<display><pft>+ &prolog + "</pft></display>"
 18. "<loop>"
 19. "<display><pft>+ &pft + "</pft></display>"
 20. "</loop>"
 21. "<field action=replace tag=1021><pft>"+ &bool + ""</pft></field>"
 22. "<display><pft>"
 23. "if val(v1091)>0 then '<error-msg>',/, 'Error ',v1091,' ,syntax error:
' ,v1092,/, '</error-msg>',/, "
 24. "else if val(v1002)=0 then '<error-msg>',/, 'No records found',/, '</error-msg>',/,
else EpilogMessage fi,fi"
 25. "</pft></display>"
 26. "</do>"
 27. "</IsisScript>"
3. close ISIS_ScriptFilename
4. return

WXISgateway(ISIS_Script_docs_unsorted)

1. ISIS_ScriptFilename=FileQuery+".wxis"
2. open ISIS_ScriptFilename and write
 1. "<IsisScript name=ISIS_script_docs_unsorted>"
 2. "<parm name=cipar><pft>"
 3. "'GetFromCiparArray(&dbname+*.*)'/'"
 4. "if &gizmo!=null 'GetFromCiparArray(&gizmo+*.*)'/'"
 5. "if &prolog.startsWith(@) 'GetFromCiparArray(&prolog.substring(1))'/'"
 6. "if &pft.startsWith(@) 'GetFromCiparArray(&pft.substring(1))'/'"
 7. "</pft></parm>"
 8. "<do task=search>"
 9. "<file action=create type=output>"+ &out + "</file>"
 10. "<field action=define tag=1001>Isis_Current</field>"
 11. "<field action=define tag=1002>Isis_Total</field>"
 12. "<field action=define tag=1091>Isis_Status</field>"
 13. "<field action=define tag=1031>Isis_From</field>"
 14. "<field action=define tag=1092>Isis_ErrorInfo</field>"
 15. "<parm name=expression>"+ &bool + "</parm>"
 16. "<parm name=from>"+ &h1 + "</parm>"
 17. "<parm name=count>"+ (&h2-&h1+1) + "</parm>"
 18. "<parm name=db>"+ &dbname + "</parm>"
 19. "<parm name=gizmo>"+ &gizmo + "</parm>"
 20. "<display><pft>"+ &prolog + "</pft></display>"
 21. "<loop>"
 22. "<field action=replace tag=1032><pft>if val(v1002) > val(v1031)+"+ (&h2-&h1+1) + "-1 then "+ &h2 + " else F(val(v1002),1,0),fi </pft></field>"
 23. "<display><pft>"+ &pft + "</pft></display>"
 24. "</loop>"
 25. "<field action=replace tag=3333><pft>"+ &pft.substring(1) + "</pft></field>"
 26. "<field action=replace tag=1021><pft>"+ &bool + "</pft></field>"
 27. "<display><pft>"
 28. "if val(v1091)>0 then '<error-msg>',/, 'Error ',v1091, ' ,syntax error: ',v1092,/, '</error-msg>',/,,"
 29. "else if val(v1002)=0 then '<error-msg>',/, 'No records found',/, '</error-msg>',/, else EpilogMessage fi,fi"
 30. "</pft></display>"
 31. "</do>"
 32. "</IsisScript>"
3. close ISIS_ScriptFilename
4. return

WXISgateway(ISIS_Script_docs_sorted)

1. ISIS_ScriptFilename=FileQuery+".wxis"
2. open ISIS_ScriptFilename and write
 1. "<IsisScript name=ISIS_script_docs_sorted>"
 2. "<parm name=cipar><pft>"
 3. "'GetFromCiparArray(&dbname+*.*)'/'"
 4. if **&gizmo**!=null "'GetFromCiparArray(&gizmo+*.*)'/'"
 5. if **&prolog**.startsWith(@) "'GetFromCiparArray(&prolog.substring(1))'/'"
 6. if **&pft**.startsWith(@) "'GetFromCiparArray(&pft.substring(1))'/'"
 7. if **&freq**.startsWith(@) "'GetFromCiparArray(&freq.substring(1))'/'"
 8. "</pft></parm>"
 9. "<file action=create type=output>" + **&out** + "</file>"
 10. "<do task=search>"
 11. "<field action=define tag=1001>Isis_Current</field>"
 12. "<field action=define tag=1002>Isis_Total</field>"
 13. "<field action=define tag=1091>Isis_Status</field>"
 14. "<field action=define tag=1092>Isis_ErrorInfo</field>"
 15. "<parm name=expression>" + **&bool** + "</parm>"
 16. "<parm name=from>" + **&h1** + "</parm>"
 17. "<parm name=count>" + (**&h2-&h1+1**) + "</parm>"
 18. "<parm name=db>" + **&dbname** + "</parm>"
 19. "<parm name=gizmo>" + **&gizmo** + "</parm>"
 20. "<display><pft>" + **&prolog** + "</pft></display>"
 21. "<loop>"
 22. "<list action=load type=sort><pft>" + **&freq** + ",/</pft></list>"
 23. "</loop>"
 24. "<display><pft>"
 25. "if val(v1091)>0 then '<error-msg>',/, 'Error ',v1091, ' ,syntax error:
' ,v1092,/, '</error-msg>',/,,"
 26. "else if val(v1002)=0 then '<error-msg>',/, 'No records found',/, '</error-msg>',/,
fi,"
 27. "</pft></display>"
 28. "</do>"
 29. "<do task=list>"
 30. "<parm name=from>" + **"1"** + "</parm>"
 31. "<parm name=count>" + (**&h2-&h1+1**) + "</parm>"
 32. "<parm name=db>" + **&dbname** + "</parm>"
 33. "<field action=define tag=1001>Isis_Current</field>"
 34. "<field action=define tag=1002>Isis_Items</field>"
 35. "<field action=define tag=1031>Isis_From</field>"
 36. "<field action=define tag=1>Isis_Item</field>"
 37. "<loop>"
 38. "<field action=replace tag=1021><pft>" + **&bool** + ""</pft></field>"
 39. "<field action=replace tag=1032><pft>if val(v1002) > val(v1031)+$(\&h2-\&h1+1)$ +$(\&h2-\&h1+1)$ +-1 then "" + (**&h2-&h1+1**) + "" else F(val(v1002),1,0),fi</pft></field>"
 40. "<display><pft>" + **&pft** + "</pft></display>"
 41. "<field action=replace tag=3333><pft>" + **&pft.substring(1)** + ""</pft></field>"
 42. "<display><pft>" + **EpilogMessage** + "</pft></display>"
 43. "</loop>"

44. "</do>"
45. "</IsisScript>"
3. close ISIS_ScriptFilename
4. return

WXISgateway(ISIS_Script_get_docs)

1. ISIS_ScriptFilename=FileQuery+".wxis"
2. open ISIS_ScriptFilename and write
 1. "<IsisScript name=ISIS_script_get_docs>"
 2. "<parm name=cipar><pft>"
 3. "GetFromCiparArray(&dbname+*.*)'"
 4. if &gizmo!=null "GetFromCiparArray(&gizmo+*.*)'"
 5. if &prolog.startsWith(@) "GetFromCiparArray(&prolog.substring(1))'"
 6. if &pft.startsWith(@) "GetFromCiparArray(&pft.substring(1))'"
 7. "</pft></parm>"
 8. "<do task=mfnrange>"
 9. if !&outapp "<file action=create type=output>"+ &out + "</file>"
 10. else "<file action=create type=append>"+ &outapp + "</file>"
 11. <field action=replace tag=2000><pft>"+ &from + ""</pft></field>"
 12. "<flow action=jump><pft>if val(v2000)>0 then 'GOOD' fi</pft></flow>"
 13. "<display><pft>"+ &prolog + ""</pft></display>"
 14. "<display><pft>'<error-msg>',/, '0 records retrieved',/, '</error-msg>',/, '</pft></display>"
 15. if !&outapp "<file action=close type=output>"+ &out + "</file>"
 16. else "<file action=close type=output>"+ &outapp + "</file>"
 17. "<flow action=exit>0</flow>"
 18. "<label>GOOD</label>"
 19. "<field action=define tag=1001>Isis_Current</field>"
 20. "<field action=define tag=1002>Isis_Total</field>"
 21. "<field action=define tag=1091>Isis_Status</field>"
 22. "<field action=define tag=1031>Isis_From</field>"
 23. "<field action=define tag=1092>Isis_ErrorInfo</field>"
 24. "<parm name=expression>"+ &bool + ""</parm>"
 25. "<parm name=from><pft> v2000 </pft></parm>"
 26. "<parm name=count>"+ &count + ""</parm>"
 27. "<parm name=db>"+ &dbname + ""</parm>"
 28. "<parm name=gizmo>"+ &gizmo + ""</parm>"
 29. "<display><pft>"+ &prolog + ""</pft></display>"
 30. "<loop>"
 31. "<field action=replace tag=1032><pft>if val(v1002) > val(v1031)+&count + "-1 then "+ &count+&from + "-1 "+ " else F(val(v1002),1,0),fi </pft></field>"
 32. "<display><pft>"+ &pft + ""</pft></display>"
 33. "</loop>"
 34. "<display><pft>"
 35. "if val(v1091)>0 then '<error-msg>',/, 'Error ',v1091,' ,syntax error: ',v1092,/, '</error-msg>',/, "
 36. "else if val(v1001)=0 or val(v1031)=0 then '<error-msg>',/, 'No records found',/, '</error-msg>',/, ' fi,fi"
 37. "</pft></display>"
 38. "</do>"
 39. "</IsisScript>"
3. close ISIS_ScriptFilename
4. return

WXISgateway(ISIS_Script_get_posting)

1. ISIS_ScriptFilename=FileQuery+".wxis"
2. open ISIS_ScriptFilename and write
 1. "<IsisScript name=ISIS_script_get_posting>"
 2. "<parm name=cipar><pft>"
 3. "'GetFromCiparArray(&dbname+*.*)'/'"
 4. if **&gizmo**!=null "'GetFromCiparArray(&gizmo+*.*)'/'"
 5. if **&prolog**.startsWith(@) "'GetFromCiparArray(&prolog.substring(1))'/'"
 6. if **&pft**.startsWith(@) "'GetFromCiparArray(&pft.substring(1))'/'"
 7. "</pft></parm>"
 8. "<do task=keyrange>"
 9. if **!&outapp**"<file action=create type=output>" + **&out** + "</file>"
 10. else "<file action=create type=append>" + **&outapp** + "</file>"
 11. "<field action=define tag=1001>Isis_Current</field>"
 12. "<field action=define tag=1002>Isis_Total</field>"
 13. "<field action=define tag=1091>Isis_Status</field>"
 14. "<field action=define tag=1031>Isis_From</field>"
 15. "<field action=define tag=1092>Isis_ErrorInfo</field>"
 16. "<field action=define tag=1>Isis_Key</field>"
 17. "<field action=define tag=2>Isis_Postings</field>"
 18. "<field action=define tag=3>Isis_Posting</field>"
 19. "<parm name=from><pft>" + **&k1** + "</pft></parm>"
 20. "<parm name=to><pft>" + **&k2** + "</pft></parm>"
 21. "<parm name=count>" + **&count** + "</parm>"
 22. "<parm name=db>" + **&dbname** + "</parm>"
 23. "<parm name=gizmo>" + **&gizmo** + "</parm>"
 24. "<parm name=posting> 1 </parm>"
 25. "<display><pft>" + **&prolog** + "</pft></display>"
 26. "<loop>"
 27. "<display><pft>" + **&pft** + "</pft></display>"
 28. "</loop>"
 29. "<field action=replace tag=3333><pft>" + **&pft.substring(1)** + "</pft></field>"
 30. if **&k2!=null** "<display><pft>if val(v1001)=0 then '<error-msg>',/, '0 record retrieved',/, '</error-msg>' else '<NumberOfSelectedIndexRecords>',/, v1031, ' :', v1, ' :', v1001,/, '</NumberOfSelectedIndexRecords>' fi, 'EpilogMessage'</pft></display>"
 31. else "<display><pft>if val(v1001)=0 then '<error-msg>',/, '0 record retrieved',/, '</error-msg>',/, fi, 'EpilogMessage'</pft></display>"
 32. "</do>"
 33. "</IsisScript>"
3. close ISIS_ScriptFilename
4. return

Lo sviluppo del gateway wwwisis > wxis

Il gateway è stato sviluppato in java ed è stato denominato AccessIsisViaWxis.java.

Il suo sviluppo ha richiesto la definizione di cinque parametri di sistema che sono stati definiti e implementati nella classe IsisSysTable:

1. EnabledwxisProgram(**false/true**) autorizza il sistema ad accedere ai database CDS/ISIS utilizzando la API wxis attivata dal programma wxisProgramName
2. wxisProgramName(**C:\wxis.exe**) è il programma, distribuito dall'UNESCO, che implementa la API wxis
3. wxisSyntaxErrorMessage('(<error-msg>','/','Error ',**v1091**',' , **syntax error: ',v1092**,','/','</error-msg>') che definisce la modalità di presentazione degli eventuali "Syntax Error" che dovessero verificarsi.

Si notino i virtual tag v1091 e v1092 che contengono rispettivamente il codice di errore e l'errore medesimo. Il messaggio può essere personalizzato in qualunque modo si ritenga opportuno.

Nel caso particolare del sistema **MetaPub** è stato definito, per le Pubblicazioni dell'ISTI, come:

```
'(<error-msg>','/','Pubblicazioni ISTI: Error ',v1091',' , syntax error: ',v1092,'/','</error-msg>'
```

4. wxisEpilogMessage(**if v3333:'Shproof.pft' OR v3333:'Aix.pft' OR v3333:'v1002' then else '</Start>' fi**) definisce il messaggio finale da inviare al sistema per concludere le operazioni di stampa. Il default tiene conto del virtual tag v3333 che viene caricato dal sistema con il nome del Print Format utilizzato e a seconda del relativo nome invia o meno la chiusura della lista dei record prodotti (</Start>)

Anche questo messaggio può essere personalizzato a proprio piacimento.

Nel caso particolare del sistema **MetaPub** è stato definito, per le Pubblicazioni dell'ISTI, come:

```
if v3333:'Shproof.pft' OR v3333:'Aix.pft' OR v3333:'v1002' then else '</Start>' fi
```

si noti che la costante letterale 'v1002' rappresenta il contenuto di un PrintFormat particolare utilizzato per stampare il massimo numero di documenti/chiavi selezionate

5. wxisZeroRecordMessage('(<error-msg>','/','**0 record retrieved**','/','</error-msg>') rappresenta il messaggio di errore da inviare al client nel caso di nessun risultato selezionato.

Anche esso può essere personalizzato così come nel file di configurazione dell'OPAC dell'Area di Ricerca del CNR di Pisa relativo al metaopac MAI/AZALAI:

'<error-msg>',/,,'Your request ',v1021,' : 0 record retrieved',/,,'</error-msg>'

Si noti il virtual tag v1021 che contiene la query eseguita.

La distribuzione e l'installazione del gateway

Il gateway wwwisis->wxis (la classe java AccessIsisViaWxis) è distribuita con il software ExtGAS/IsisGAS alle seguenti URL:

1. http://leonardo.isti.cnr.it/metaopac/ExtGAS.src.new_3.tgz che contiene le classi java del sistema ExtGAS/IsisGAS.
2. <http://leonardo.isti.cnr.it/metaopac/ExtGasSystem.tgz> che contiene il sistema ExtGAS/IsisGAS con esempi, utility CDS/ISIS, API CDS/ISIS e documentazione

Con il comando:

```
tar -xzf ExtGAS.src.new_3
```

o

```
tar -xzf ExtGasSystem.tgz
```

fare l'untar della distribuzione.

Dalla directory ExtGAS.src.new_3 copiare le classi nella relativa webapps del Jserver che si utilizza.

Per attivare il gateway è sufficiente definire nel relativo Database System File i seguenti parametri:

1. EnabledwxisProgram true
2. wxisProgramName /<dirname>/wxis

Se necessario, successivamente, modificare i rimanenti tre parametri descritti nel paragrafo precedente:

1. wxisSyntaxErrorMessage
2. wxisEpilogMessage
3. wxisZeroRecordMessage

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 multiplatforma 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>
19. Giuseppe A. Romano - ExtGASoai - OAI-PMH gateway per ExtGAS – ISTI-B4-20 , 2007 URL: http://leonardo.isti.cnr.it/metaopac/ExtGAS_OAI.pdf

20. Giuseppe A. Romano - ExtGASoai - OAI-PMH gateway per ExtGAS: organizzazione logica del software - ISTI-B4-13 , 2007 URL: http://leonardo.isti.cnr.it/metaopac/ExtGAS_OAI_logic.pdf
21. Giuseppe A. Romano - ExtGASoai - OAI-PMH gateway per ExtGAS: manuale di installazione e d'uso (Draft version) - ISTI-B4-12 , 2007 URL: http://leonardo.isti.cnr.it/metaopac/ExtGAS_OAI_install.pdf
22. Andrew Buxton – The WWWISIS Handbook (for version 4 and 5), Institute of Development Studies at University of Sussex - March 2002
23. BIREME / PAHO / WHO :Latin America and Caribbean Center on Health Sciences Information – IsisScript Language Reference – Version 1.1, Sao Paulo 2006