



adlib

WWWOPAC
referentiegids

Adlib Information Systems

Copyright © 1992-2011 Adlib Information Systems B.V.® Alle rechten voorbehouden. Adlib® is een product van Adlib Information Systems B.V.®

De informatie in dit document kan zonder enige voorafgaande waarschuwing worden gewijzigd en houdt geen verplichting in voor Adlib Information Systems. Adlib aanvaardt geen aansprakelijkheid voor de volstrekte juistheid en volledigheid van de hierin opgenomen teksten. De software, zoals deze in dit document staat beschreven, wordt geleverd onder de voorwaarden van een gebruiksrechtovereenkomst. De bedoelde software mag uitsluitend volgens de voorwaarden van deze overeenkomst worden gebruikt of gekopieerd.

Daar onze producten voortdurend verbeterd worden, kunnen latere versies verschillen met de producten die hierin beschreven staan. Dit document houdt geen enkele contractuele verplichting in om software te leveren, en mag niet als definitieve productbeschrijving worden beschouwd.

Inhoud

Inleiding	1
1 WWWOPAC configureren	3
1.1 Webconfiguratiebestanden	3
1.2 WWWOPAC-opdrachten	7
1.2.1 Redundante opdrachten	14
1.2.2 Selectief wwwopac-functionaliiteit uitschakelen	19
1.2.3 type="richText" in <field> tags	21
1.2.4 Adlib foutmeldingen	22
2 Een zoekvraag opgeven	23
2.1 Opdrachten en parameters in de CGI-string,	23
2.1.1 Drie verschillende syntaxes	35
2.1.2 SEARCH en FLD/VAL-paren in één zoekvraag	36
2.1.3 Pointerfiles openen	37
2.1.4 isutf8=1	37
2.2 Afbeeldingen ophalen	38
2.2.1 Dynamisch thumbnails maken	38
2.2.2 Afbeeldingen ophalen in hoge resolutie	41
2.2.3 Standaard miniatuurafbeelding	42
2.2.4 Getoonde afbeeldingen automatisch watermerken	43
2.3 Imagemetadata	44
3 OAI (Open Archives Initiative)	47
3.1 Instellingen voor het adlibweb.xml-bestand	48
3.1.1 AdlibXML	51
3.2 Gebruik	52
3.3 Protocol requests	54
3.4 OAI Sets in Adlib	55
3.5 ResumptionTokens	56
3.6 Voorbeelden	58
3.7 OAI repositories en zoekmachines	58
3.8 Bepaalde records uitsluiten van OAI-zoekresultaten	60
4 SRW/SRU	61
4.1 Hoe Adlib SRU implementeert	62
4.1.1 Uw eigen SRU Profile maken	63
4.1.2 De Adlib SRU-server (wwwopac) configureren	64
4.2 Enkele kenmerken van SRW/SRU requests	71
4.3 Uw SRU-server testen	72

5 Vanaf het web data schrijven	73
5.1 CGI escapen	73
5.2 Records schrijven, bijwerken of verwijderen	74
5.3 Na het schrijven van een record	75
6 Aanverwante zaken	77
6.1 Status	77
6.2 Veldnamen	79
6.3 Meertalige velden	79
6.4 Afhankelijkheden	80
7 Bijlage 1: een eigen recordformaat	81
8 Bijlage 2: het Adlib Base Profile	85
9 Index	95

Inleiding

De Adlib Internet Servermodule stelt u in staat om uw catalogus voor het publiek beschikbaar te stellen op het internet en/of intranet. Met behulp van zoekformulieren en andere webpagina's kan de bezoeker van uw website rechtstreeks in de catalogus zoeken. De Internet Servermodule bestaat voornamelijk uit het programma `wwwopac.exe` en een webapplicatie genaamd Adlib Internet Server.

`Wwwopac.exe` is een executable zonder grafische gebruikersinterface die in samenwerking met een webserver fungeert als Common Gateway Interface-programma. Het verzorgt de interactie tussen de webapplicatie en de database. Dat komt neer op het vertalen van http zoekopdrachten samengesteld door de webapplicatie, naar een Adlib-zoekopdracht, het vervolgens daadwerkelijk uitvoeren van die zoekopdracht in de database, en het dan teruggeven van het zoekresultaat als een lijst van records of termen in XML-formaat.

De Internet Server webapplicatie transformeert dit XML-zoekresultaat met behulp van XSLT stylesheets naar een stuk HTML, en voegt dit samen met andere stukjes HTML die worden geproduceerd door de `aspx`-pagina en de daarin aanwezige zogenaamde (`.ascx`) usercontrols die alle een afzonderlijk stukje grafische interface (een knop, een tab, e.d.) van het gehele HTML-formulier programmeren. Het HTML-document gebruikt verder CSS-stijlen om opmaak (kleuren, lettertypes, e.d.) aan de verschillende grafische elementen toe te kennen. Het volledige document wordt op de server zo samengesteld, en dan pas verzonden naar de browser op de client die alleen nog maar die HTML-pagina hoeft weer te geven.

De logica van de Internet Server webapplicatie, die nodig is voor de werking van de usercontrols en het samenstellen van de http zoekopdracht, is geprogrammeerd in .NET, en zit gecompileerd opgeslagen in verschillende dll's die u in de `\bin` of `\executables`-map vindt. Deze code kunt u daarom niet wijzigen.

Maar doordat het zichtbare deel van de Internet Server webapplicatie, de HTML-pagina's en formulieren die in de browser worden weergegeven, helemaal wordt opgebouwd via usercontrols (die zelfstandige stukjes HTML representeren), kunt u de lay-out van de hele webapplicatie wél aanpassen. U kunt usercontrols simpelweg verplaatsen om de structuur van een webpagina te veranderen, of de usercontrols zelf wijzigen om de webapplicatie nog meer in de eigen huisstijl vorm te geven.

De `wwwopac.exe` wordt geïnstalleerd als aanvullend programma bij een bestaande webserver (IIS 5.0 of hoger voor Windows 2000, XP, of

hoger), en moet via eigen instellingen in het bestand *adlibweb.xml* worden voorbereid op het gebruik binnen uw netwerk of systeem. Zie voor de installatieprocedure van de Internet Servermodule en systeemeisen de *Installatiegids Adlib Internet Server* (op onze website).


(Een webserver is een programma dat verzoeken om webpagina's van webbrowsers opvangt en uitvoert. Daarnaast zorgt de webserver voor het starten van aanvullende programma's, zoals wwwopac, en voor de beveiliging van documenten.)

Deze referentiegids beschrijft in het algemeen alle functionaliteit die wwwopac.exe (meest recente versie) biedt, en verwijst waar van toepassing kort naar de implementatie ervan in versie 3 (ook van toepassing op 3.2) van de Internet Server modelwebapplicatie.

U kunt voor wwwopac in principe helemaal zelf een webapplicatie bouwen, maar de Internet Server modelwebapplicatie is heel flexibel en u kunt hem voor een belangrijk deel aan uw eigen wensen aanpassen, of u kunt dat door AIS laten doen; dus het gebruik van de modelwebapplicatie is wel zo efficiënt en kostenbesparend.

U kunt de Internet Server modelwebapplicatie bekijken op:

<http://demo.adlibsoft.com/internetserver3> Klik in de geopende

website op  voor een Help-tekst over de functionaliteit op het momenteel getoonde tabblad.

■ Benodigde kennis

Voor de basis van het leren aansturen van alle functionaliteit van wwwopac hebt u voldoende aan deze handleiding. Voor meer informatie over de toepassing van externe technologieën en standaarden wordt u ter plekke doorverwezen.

Voor het kunnen aanpassen van de Internet Server modelwebapplicatie is kennis van HTML, XML, XSLT, CSS en ASPX noodzakelijk. Het zoekresultaat dat wwwopac oplevert is namelijk in XML-formaat, en de webapplicatie gebruikt XSLT en CSS stylesheets om die data van een grafische lay-out te voorzien. De HTML webpagina's als geheel (waar het zoekresultaat een onderdeel van kan zijn) worden via ASPX samengesteld. Maar dit onderwerp maakt geen deel uit van deze gids. Voor het leren maken van substantiële aanpassingen aan de webapplicatie (die natuurlijk verder gaan dan de noodzakelijke instellingen tijdens installatie) biedt AIS een cursus.

1 WWWOPAC configureren

1.1 Webconfiguratiebestanden

In een webconfiguratiebestand initialiseert u het gebruik van `wwwopac`. Er zijn twee varianten: *adlibweb.xml* (één XML-bestand), en `www`-bestanden (één of meer tekstbestanden met de extensie `.www`, die voor Internet Server 3 en hoger niet worden gebruikt). *Adlibweb.xml* werd met `wwwopac 4.7` geïntroduceerd en is bedoeld als vervanger voor het nu verouderde `www`-bestand. In plaats van de platte tekst in het `www`-bestand wordt XML gebruikt en dat heeft als voordeel dat u met configuratiegroepen kunt werken. Door `wwwopac 5.0` en hoger wordt het `www`-formaat wel nog ondersteund maar de voorkeur gaat uit naar het gebruik van *adlibweb.xml*; voor de toepassing van OAI (zie verderop in deze gids) is *adlibweb.xml* zelfs noodzakelijk.

In een webconfiguratiebestand moet minimaal een databasepad (`DATABASEPATH`) opgegeven worden, en een databasenaam (`DATABASE`), eventueel gevolgd door de naam van een dataset gescheiden door een groter-dan teken (als die niet al in de zoekvraag wordt genoemd).

Het is in principe ook mogelijk om te zoeken zonder een webconfiguratiebestand. Alle voor de zoekvraag benodigde gegevens, die u normaal in het webconfiguratiebestand opgeeft, moet u dan wel in de CGI-request opgeven. Er blijft echter een aantal `wwwopac`-opdrachten dat u wel alleen maar in een webconfiguratiebestand kunt gebruiken (zie de lijst met `wwwopac`-opdrachten).

Het wel gebruiken van een webconfiguratiebestand (zoals in Internet Server 3) is echter aan te raden als de veiligheid van bepaalde gegevens gegarandeerd moet zijn. Alleen de databases die u in het webconfiguratiebestand opgeeft, zijn namelijk toegankelijk voor de webapplicatie en command-line zoekvragen die direct in het invoervak *Adres* van een browser worden ingevoerd. Zonder een webconfiguratiebestand kan een gebruiker zich via command-line zoekvragen toegang verschaffen tot alle Adlib databases in de data-map op de webserver.

Als u zonder webconfiguratiebestand een bepaalde database wilt aanspreken dan moet u in de CGI-request met de variabele `DATABASE` dus naar een database verwijzen zoals u dat normaal in het webconfiguratiebestand zou doen.

Wwwopac controleert dan ook eerst of deze variabele in een CGI-request naar een www-bestand met die naam verwijst, en zo niet, of er naar een database wordt verwezen. De variabele `DATABASEPATH` komt hiermee ook beschikbaar voor gebruik in CGI-requests, want anders zou wwwopac zonder webconfiguratiebestand niet weten waar de database moet worden gezocht.

Bij aanwezigheid van een willekeurig www-bestand of *adlibweb.xml*-bestand kunnen in een CGI-request geen andere databases worden aangesproken dan degene die in dat/die webconfiguratiebestand(en) zijn gespecificeerd.

Als wwwopac geen webconfiguratiebestand kan vinden (of als er geen is opgegeven), dan zal het programma op zoek gaan naar het bestand *default.www*. Als deze ook niet gevonden wordt, krijgt u een foutmelding te zien in uw browser: *WWWOPAC ERROR: unable to find a default.www file*.

In een webconfiguratiebestand kunt/moet u een aantal zaken instellen. Hiervoor wordt gebruik gemaakt van zogenaamde wwwopac-opdrachten. Dit zijn opdrachten die soms zowel in het HTML-formulier gezet kunnen worden (en daarmee ook command-line in het URL-invoervak van de browser) als in het webconfiguratiebestand. Deze wwwopac-opdrachten worden door wwwopac gebruikt om de locaties van de noodzakelijke bestanden te vinden, en om enkele parameters voor zoeken en het ophalen van data in te stellen.

Uit veiligheidsoverwegingen kan ervoor worden gekozen om zoveel mogelijk instellingen in het webconfiguratiebestand te zetten: dit bestand kan namelijk niet door de webgebruiker worden ingezien, in tegenstelling tot het HTML-formulier. In het webconfiguratiebestand kunnen zowel absolute paden als relatieve paden opgenomen worden. Gebruik bij voorkeur maar één absoluut pad.

Hieronder ziet u een voorbeeld van een eenvoudig *adlibweb.xml*-bestand:

```
<?xml version="1.0" encoding="UTF-8" ?>
<webConfiguration>
<!-- Global settings -->
<globalConfiguration>
  <databasepath>D:\demo.adlibsoft.com\IS3\data</databasepath>
  <xmltype>structured</xmltype>
  <adlib_smarthost />
  <highlight>on</highlight>
  <maxlimit>400</maxlimit>
  <logfile>default.log</logfile>
</globalConfiguration>
```



```
<!-- Default group for library -->
<groupConfiguration group="defaultLibrary">
  <brieffields>
    <field>lead_word</field>
    <field>title</field>
    <field>author.name</field>
    <field>corporate_author</field>
    <field>year_of_publication</field>
    <field>digital_reference</field>
    <field>copy.number</field>
    <field>shelf_mark</field>
  </brieffields>
  <detailfields>
    <field>lead_word</field>
    <field>title</field>
    <field>author.name</field>
    <field>corporate_author</field>
    <field>illustrator.name</field>
    <field>statement_of_responsibility</field>
    <field>edition</field>
    <field>place_of_publication</field>
    <field>publisher</field>
    <field>year_of_publication</field>
    <field>print.place</field>
    <field>print.name</field>
    <field>pagination</field>
    <field>illustrations</field>
    <field>dimensions</field>
    <field>material_type</field>
    <field>accompanying_material</field>
    <field>series.title</field>
    <field>series.number</field>
    <field>subseries.title</field>
    <field>subseries.number</field>
    <field>subseries.issn</field>
    <field>isbn</field>
    <field>binding_method</field>
    <field>parent</field>
    <field>child</field>
    <field>object.object_number</field>
    <field>keyword.contents</field>
    <field>geographical_keyword</field>
    <field>person.keyword.name</field>
    <field>copy.number</field>
    <field>shelf_mark</field>
    <field>site</field>
    <field>loan_category</field>
    <field>digital_reference</field>
    <field type="richText">abstract</field>
    <field type="richText">notes</field>
    <field>source.title.article</field>
    <field>source.title</field>
    <field>source.day</field>
    <field>source.month</field>
    <field>source.volume</field>
  </detailfields>
</groupConfiguration>
```

```

    <field>source.issue</field>
  </detailfields>
</groupConfiguration>

<!-- Database Full Catalogue -->
<databaseConfiguration database="ChoiceFullCatalogue" groups=
  "defaultLibrary">
  <database>document>fullcatalogue</database>
</databaseConfiguration>
<!-- note: a databaseConfiguration tag may have only one groups
attribute -->

<!-- Database Books -->
<databaseConfiguration database="ChoiceBooks" groups=
  "defaultLibrary">
  <database>document>book</database>
</databaseConfiguration>

<!-- Database Audio Visuals -->
<databaseConfiguration database="ChoiceAudioVisuals" groups=
  "defaultLibrary">
  <database>document>avm</database>
</databaseConfiguration>

<!-- Database Articles -->
<databaseConfiguration database="ChoiceArticles" groups=
  "defaultLibrary">
  <database>document>article</database>
</databaseConfiguration>

<!-- Database Serials -->
<databaseConfiguration database="ChoiceSerials" groups=
  "defaultLibrary">
  <database>document>serials</database>
</databaseConfiguration>

<!-- Database Dublin Core -->
<databaseConfiguration database="ChoiceDublinCore" groups=
  "defaultLibrary">
  <database>document>resource</database>
</databaseConfiguration>

</webConfiguration>

```

■ Virtuele directory's

Om te voorkomen dat webgebruikers toegang hebben tot de gehele harde schijf van een computer, wordt er bij webapplicaties gewerkt met 'virtuele' directory's die door forward slashes van elkaar worden gescheiden. Een virtuele directory verwijst naar de echte directory. U maakt een virtueel directory bij het beheer van IIS 5; ken aan een virtueel directory enkel de hoogst noodzakelijke toegangsrechten toe.

Alleen de directory met *wwwopac.exe* heeft uitvoerrechten nodig. Voor de andere directory's zijn alleen leesrechten voldoende. Zo kunt u bijvoorbeeld de virtuele directory *'/adlib'* maken voor de echte directory: *'c:\bibl\wwwopac\wwwroot'* en die alleen leesrechten geven.

Deze virtuele directory's kunt u in plaats van de echte directory's opnemen in een webconfiguratiebestand (gebruik dan dus forward slashes).

1.2 WWWOPAC-opdrachten

In een webconfiguratiebestand hebt u de onderstaande opdrachten tot uw beschikking. Deze opdrachten zijn niet hoofdlettergevoelig.

```
<!-- commentaar -->
```

In *adlibweb.xml* plaats u zo commentaar.

```
ADLIB_PTR=<padnaam naar map met pointerfiles>
```

Als u met *wwwopac.exe* pointerfiles wilt openen die zich niet bevinden op de standaard plek in een submap van de *data-*map van uw applicatie, dan geeft u met deze variabele de padnaam op van de map waarin de pointerfiles zich wel bevinden.

```
ALLOW_ALL_DBS=true
```

`<allow_all_dbs>true</allow_all_dbs>` zorgt ervoor dat alle beschikbare databases doorzocht kunnen worden.

Normaal gesproken zou u een webconfiguratiebestand gebruiken waarin elke database die doorzocht mag worden, moet worden geïnitieerd. Dit is een goede beveiliging voor databases die niet toegankelijk mogen zijn vanaf het web. Maar als dat niet nodig is, dan is `ALLOW_ALL_DBS` een handige manier om die beveiliging te omzeilen. `ALLOW_ALL_DBS` kan alleen in *default.www* of *adlibweb.xml* gebruikt worden.

Als u dus wilt toestaan dat er in alle databases gezocht kan worden, dan specificeert u dit ofwel in *default.www* of *adlibweb.xml* met `ALLOW_ALL_DBS`, of door helemaal geen *default.www* of *adlibweb.xml* te definiëren.

BRIEFFIELDS=<taglijst of veldenlijst>

Met de variabele `brieffields` geeft u de velden op die u in de lijstweergave van het zoekresultaat wilt zien, bijvoorbeeld:

```
<brieffields>
  <field>lead_word</field>
  <field>title</field>
  <field>author.name</field>
  <field>corporate_author</field>
  <field>year_of_publication</field>
  <field>digital_reference</field>
  <field>copy.number</field>
  <field>shelf_mark</field>
</brieffields>
```

U kunt tags en veldnamen door elkaar gebruiken. Tags worden, voor zover bekend in Adlib, omgezet naar veldnamen in de XML-uitvoer.

Als u alle velden uit de datadictionary in de verkorte weergave van records wilt opnemen, dan hoeft u niet alle velden op te sommen, maar gebruikt u:

```
<brieffields>
  <allDataDictionaryFields/>
</brieffields>
```

om dat in één keer te doen. Wilt u daarnaast ook nog velden tonen die niet in de datadictionary gedefinieerd zijn, voeg de tags daarvan dan als een normale opsomming voor deze variabele toe, bijvoorbeeld:

```
<brieffields>
  <allDataDictionaryFields/>
  <field>au</field>
  <field>ti</field>
</brieffields>
```

Als er geen `brieffields` worden opgegeven, dan worden bij SCAN de volgende tags gebruikt: %0, te, uf, us, rt, bt, nt en et. Bij een search (zoeken zonder SCAN) worden dan alle velden gebruikt (niet de tags die niet als veld bekend zijn).

Voor optimale prestaties van Adlib is het beslist aan te raden alleen de `brieffields` op te geven die u daadwerkelijk wilt gebruiken. Elk te verwerken veld (zeker een gekoppeld veld) kost namelijk processortijd, en bij het ophalen van meerdere records voor een lijstweergave telt dat zeker op.

DATABASE=<FACS-naam>

Gebruik deze variabele in het webconfiguratiebestand om naar de databasenaam te verwijzen. Een variabele met dezelfde naam kunt u gebruiken in CGI-strings, maar die verwijst dan in eerste instantie naar een www-bestand, en als dat niet gevonden wordt, naar een database met die naam. Voorbeeld:
 <database>document>fullcatalogue</database>

DATABASEPATH=<schijfstation:>\<Adlib-map>\data

Dit mag in het webconfiguratiebestand voorkomen en in CGI-strings. Deze instelling verwijst naar een *fysieke* locatie, gezien vanuit de webserver. Voor de beste prestaties van de webserver moet de data op een lokale schijf van de webserver staan. (Het is overigens verstandig om in een webconfiguratiebestand maar één absoluut pad op te nemen.)
 Voorbeeld:

<databasepath>C:\onsmuseum.nl\IS3\data</databasepath>

DETAILFIELDS=<taglijst of veldenlijst>

Met de variabele `detailfields` geeft u de velden op die u in de detailweergave van het zoekresultaat wilt zien. U kunt tags en veldnamen door elkaar gebruiken. Tags worden, voor zover bekend, omgezet naar veldnamen in de XML-uitvoer.

Als er geen `detailfields` worden opgegeven, dan worden alle velden gebruikt (niet de tags die niet als veld bekend zijn). `detailfields` zijn niet van toepassing op SCAN.

Bijvoorbeeld:

```
<detailfields>
  <field>lead_word</field>
  <field>title</field>
  <field>author.name</field>
  <field>corporate_author</field>
  <field>illustrator.name</field>
</detailfields>
```

Voor optimale prestaties is het beslist aan te raden alleen de `detailfields` op te geven die u daadwerkelijk wilt gebruiken.

HIGHLIGHT=<1, on, 0, off, of gewenste XML-tag>

Het is vaak wenselijk om de (deel)term waarop een gebruiker in de webapplicatie zoekt, in het zoekresultaat te markeren om het te laten opvallen. Dit gebeurt door die (deel)term bijvoorbeeld in een andere kleur te laten weergeven, het zogenoemde highlighten.

Vóór wwwopac versie 5.0.2 was het een tijdrovende bezigheid

om deze functionaliteit in een webapplicatie te beheren, want in het Internet Server ASP-script stond speciale code om de zoekterm in de inhoud van de opgehaalde (applicatie-afhankelijke) tags op te zoeken en via XSL anders op te laten maken.

Vanaf 5.0.2 is de implementatie ervan daarom vereenvoudigd. Highlighting is daartoe ingebouwd in `wwwopac`. In een zoekvraag of in het configuratiebestand kunt u deze functie uitschakelen via `HIGHLIGHT=0` en respectievelijk `<highlight>off</highlight>` (standaard staat highlighting aan). De default instelling zorgt ervoor dat de gezochte (deel)term in het XML-resultaat tussen `<highlight>` en `</highlight>` komt te staan. Om de Internet Server webapplicatie iets met deze highlight tags te laten doen, moet `HIGHLIGHT=1` óók in het *globalsettings.xml*-bestand staan ingesteld.

In Internet Server 3 wordt de stijl van highlighting in het bestand *adlib.css* gedefinieerd, in de `highlight` class:

```
.highlight
{
  font-weight: bold;
  color: #cc1010;
}
```

U kunt dit eenvoudig aanpassen om de stijl te veranderen.

Overigens, als u de XML-tag `<highlight>` om wat voor reden dan ook, niet kunt gebruiken voor dit doel, dan kunt u bij het inschakelen van highlighting opgeven dat deze XML-tag een andere naam moet krijgen. U doet dit via de opdracht `<highlight>gewenste_XML-tag</highlight>`. U schakelt hiermee het highlighten in én u definieert tegelijk de XML-tag, bijvoorbeeld: `<highlight>markeren</highlight>`. In het XML-resultaat wordt een gezochte (deel)term nu omsloten door `<markeren>` en `</markeren>`. Hierop moet de stylesheet *highlight.xsl* dan even worden aangepast.

Voor getrunceerd zoeken is er verder een verschil tussen het automatisch highlighten van veldinhoud uit term- en woordindexen. Na normaal (rechts)getrunceerd zoeken in term-indexen wordt alleen de deelterm aan het begin van de term gehighlight (dus waarop de term werd gevonden) en niet eventueel verderop nogmaals voorkomende deeltermen. Als u

dus bijvoorbeeld zoekt op *a** en een gevonden term is *adrenaline*, dan wordt alleen de eerste *a* gehighlight. Voor links trunceren geldt een analoge redenering. Wanneer uw indexen geschikt zijn gemaakt voor links trunceren, en u zoekt bijvoorbeeld op **e*, en een gevonden term is *adrenaline*, dan wordt alleen de *e* op het eind gehighlight. In het zoekresultaat uit een woordindex wordt wel elk voorkomen van de gezochte (deel)term gehighlight.

Opmerkingen:

- Met een OR-combinatie zoeken in de woordindex van één veld, highlight alleen de eerste (deel)term waarop u zoekt, niet de tweede. Voor een uitgebreide zoekvraag waarbij op meerdere velden wordt gezocht, geldt dat per veld alleen de eerste (deel)term waarop wordt gezocht wordt onthouden voor het highlighten.
- Highlighting werkt niet voor numerieke, logische, en datum-indexen.
- Highlighting werkt alleen bij is-gelijk (=) of contains-zoekvragen.

LOGFILE=<pad naar loggingbestand>

Schakel logging aan door met deze variabele een loggingbestand op te geven, bijvoorbeeld:

```
LOGFILE=../data/webstat.csv
```

Het aangegeven bestand wordt gecreëerd en groeit met elke search die *wwwopac.exe* verwerkt. Het formaat van het bestand is comma-separated (csv) en kan overal worden gecreëerd en bewaard. (Omdat het een door komma's gescheiden bestand is, kan het door een verscheidenheid aan programma's worden geopend, inclusief Microsoft Excel.)

De informatie die wordt opgeslagen betreft de datum en het tijdstip van de zoekactie, het IP-adres* vanaf waar de zoekactie kwam, welke database werd doorzocht, en het aantal records dat werd opgehaald.

* In de oude twee-lagen structuur had *wwwopac* direct toegang tot het IP-adres en logde dat in het derde veld van het logbestand.

In een drie-lagen webapplicatie zoals die van Internet Server 3 heeft *wwwopac* echter niet direct toegang tot het IP-adres van de bezoeker, en wordt in het derde veld in het logbestand steeds het IP-adres gelogd van

de computer waarop deze ASP-applicatie draait, omdat dat de machine is die wwopac aanroept. Daarom moet u de ASP-webapplicatie als volgt aanpassen om toch het IP-adres van de bezoeker te loggen: aan elke zoekvraag die naar wwopac wordt gestuurd moet door die webapplicatie `&callerIP=<IP-adres of DNS-naam van bezoeker>` worden toegevoegd. Deze zogenaamde *callerIP* is het laatste veld in het logbestand van wwopac. (Wwwopac blijft overigens nog steeds ook het derde veld vullen, al is dat in de drie-lagen webapplicatie niet bruikbaar.) Als de *callerIP* niet meegestuurd wordt, dan zal een kopie van het derde veld gelogd worden, om compatibiliteit met twee-lagen webapplicaties zeker te stellen.

Er is ook een handige manier om elke maand (of elke dag of elk jaar) automatisch een nieuw logbestand aan te laten maken, zodat deze bestanden niet te groot worden. U doet dat door letterlijk bijvoorbeeld `LOGFILE=log%m%Y.txt` in het webconfiguratiebestand op te nemen. Wanneer wwopac een logbestand creëert in april 2005, dan wordt de naam van dat bestand dus `log042005.txt`. Steeds wanneer er wordt geschreven in dat logbestand wordt de huidige datum gecontroleerd, en zodra een nieuwe maand ingaat, wordt een nieuw logbestand aangemaakt voor die maand. U kunt ook `%y` (met kleine letter) gebruiken om alleen de laatste twee cijfers van een jaar op te nemen. En `%d` kunt u optioneel nog toepassen om elke dag een nieuw bestand aan te laten maken. (Dit `txt`-bestand wordt trouwens gevuld als een `csv`-bestand.)

RUNADAPL=<mijnadapl>

In het webconfiguratiebestand of in de CGI-string kunt u `RUNADAPL=<mijnadapl>` opgeven om direct een bepaalde adapt uit te voeren, bijvoorbeeld om naar de database te schrijven.

TIMEOUT=<seconden>

Met `TIMEOUT` stelt u de tijd in waarna de zoekopdracht wordt beëindigd, ongeacht of er nog wordt gezocht. Deze parameter voorkomt dat de computer per ongeluk blijft hangen als er problemen met de zoekopdracht zijn. Maar ook worden te langdurige zoekopdrachten afgebroken. De default is 300 seconden.

XMLADAPL=<pad naar pré-XML-generatie adapl>

Voorafgaande aan XML-generatie van het zoekresultaat door wwwopac kunt u per record nog een adapl laten uitvoeren. Deze adapl geldt dan voor alle typen zoekopdrachten: SEARCH, FLD... enz., SCAN én OAI-requests. Laat deze adapl geen XML of HTML genereren.

Voor gebruik bij OAI-requests is een dergelijke adapl handig om waarden uit velden die niet echt een één-op-éénrelatie met Dublin Core-elementen hebben, te bewerken of samen te voegen vóór het afbeelden ervan op Dublin Core.

XMLTYPE=<UNSTRUCTURED, STRUCTURED, DIAGNOSTIC of RAW>

De Adlib wwwopac genereert automatisch XML-uitvoer, waarmee adapls om een zoekresultaat om te zetten in HTML of XML overbodig worden.

Afhankelijk van de zoekmethode wordt een index dan wel recordlijst in XML gegenereerd: SCAN levert een index op (een lijst met sleutels) terwijl een search een lijst met records oplevert. (Met "search" worden hier alle zoekopdrachten bedoeld die geen gebruik maken van SCAN.)

XML kan in verschillende types worden opgeleverd, afhankelijk van het soort gegevens dat ermee wordt bewaard. In het webconfiguratiebestand, maar eventueel ook alleen in een CGI-string, kunt u dit type specificeren:

- UNSTRUCTURED: zoals adlwin.exe standaard-XML uitvoert;
- STRUCTURED: een soort thesaurusboom. Groeperingen, en internal en external links worden in een boomstructuur geordend;
- RAW: het complete record wordt getoond, eventueel gespecificeerde `brieffields` of `detailfields` worden dan genegeerd.
- DIAGNOSTIC: dit levert alleen het aantal hits op, maar dus geen records.
Het kan extra en meer efficiënte functionaliteit voor webapplicaties bieden, zoals de mogelijkheid voor verscheidene databases eerst het aantal hits aan de gebruiker te tonen voordat deze een keuze maakt om de resulterende records van een van die databases op te halen en te bekijken.

SCAN: Het resultaat van een SCAN, een index, wordt in het gelijknamige XMLTYPE opgeleverd in een boomstructuur met alleen interne links. Daarin worden de tags %0, te, bt, nt, et, us (use), uf (used for), sf (semantic factor), fv (semantic factor for) en rt (related term) weergegeven.

Deze XML-typen kunnen met één en dezelfde stylesheet worden gecombineerd.

1.2.1 Redundante opdrachten

Met de voortschrijdende ontwikkeling van de Internet Server webapplicatie worden sommige bestaande wwwopac-functies redundant. Deze nu verouderde wwwopac-opdrachten kunt u wel nog steeds gebruiken, maar in de Internet Server 3 webapplicatie hebben ze geen toepassing meer. Het gaat op de volgende opdrachten:

* of

In een www-bestand (niet in *adlibweb.xml*) kan commentaar worden geplaatst, om bijvoorbeeld de gekozen instellingen te documenteren in het bestand zelf. Begin een commentaarregel met een * of een #; beide tekens mogen nog vooraf worden gegaan door spaties, deze worden genegeerd. Commentaar moet op een eigen regel staan. U kunt dus niet áchter een instelling commentaar geven; doe dat ervoor of erna.

<spaties>

U kunt naar wens spaties in een www-bestand invoegen (geldt niet voor een XML-bestand), om de code duidelijker te maken. Tijdens de verwerking van het bestand worden alle spaties genegeerd. (Regelovergangen zijn wel belangrijk.)

ADLIB_DIRi=<taalspecifiek foutmeldingendirectory>

U kunt hiermee een map aangeven waarin foutmeldingsbestanden staan in één taal, in www-bestandsyntaxis bijvoorbeeld:

ADLIB_DIR1=F:\adlib\dutch\errors. Een aantal bestanden met standaard naamgeving die u in de betreffende directory kunt plaatsen, zijn *norecs.html*, *syntax.html*, *nofield.html*. Met de variabelen NORECS, SYNTAXERROR en INVALIDFIELD respectievelijk kunt u naar andere dan de default foutmeldingsbestanden verwijzen als u die een andere naam dan de standaard hebt gegeven.

ADLIB_OUTPUT=<serverschijfstation:>\temp

Hiermee kon een pad opgegeven worden waarnaar tijdelijke bestanden worden weggeschreven, en was alleen van belang voor sorteren. Het sorteren van een zoekresultaat is vanaf wwwopac 4.7 echter sneller geworden, want er is nu geen tijdelijk bestand meer nodig. Dit betekent dat u geen `ADLIB_OUTPUT`-directory meer hoeft op te geven in het webconfiguratiebestand.

BRIEFADAPL=<pad naar lijst-ADAPL>

Dit verwijst naar een `adapl` waarin de presentatie van een recordlijst is vastgelegd. Geldig vanuit het zoekformulier en in het webconfiguratiebestand; wordt gebruikt voor bijna alle selecties, ook na `SCAN`.

Wwwopac genereert automatisch XML tenzij u een `briefadapl` opgeeft. Gebruik dus alleen een `briefadapl` als u via `ADAPL HTML` wilt genereren, en dat niet in een ASPX-omgeving wilt doen.

Voor een eventuele dataconversie van zoekresultaten voordat die automatisch naar XML worden omgezet, kunt u een `XMLADAPL` opgeven.

Wanneer u in het webconfiguratiebestand een `BRIEFADAPL` specificeert dan worden de zoekresultaten voor lijstweergave dus niet in XML opgeleverd, maar door de opgegeven `adapl` bijvoorbeeld naar HTML vertaald. Deze `BRIEFADAPL` wordt alleen niet gebruikt voor een zoekvraag waarmee u een specifiek record opvraagt, bijvoorbeeld: `%250=5`. In plaats daarvan wordt de opgegeven `DETAILADAPL` gebruikt. In dit voorbeeld wordt record nr. 5 opgevraagd en het resultaat wordt dus met de `adapl` voor detailweergave verwerkt. (`%0` is de tag voor recordnummer, en de hexadecimale waarde van het `%`-teken is 25, in de CGI-string voorafgegaan door een `%`-teken.)

BRIEFSTYLE (zie `STYLE`)

CONTENT-TYPE=<text/xml of text/html>

Specificeert het type content. Alleen bij het gebruik van `adapls` die XML schrijven is het noodzakelijk `CONTENT-TYPE=text/xml` te specificeren. De default als er `adapls` gespecificeerd zijn is `text/html`. Er wordt dan van uitgegaan dat de opgegeven `adapls` HTML schrijven. Als er geen `adapls` worden gebruikt, is

de default `text/xml` want `wwwopac` genereert zonder `adapls` automatisch XML.

DETAILADAPL=<pad naar detail-ADAPL>

Geldig vanuit zoekformulier en webconfiguratiebestand. Verwijst naar een `adapl` waarin de presentatie van een enkel record is vastgelegd.

(Als er een applicatie wordt gemaakt waarbij het resultaat altijd 1 record is, dan kan de `adapl` voor het lijstschermb scherm overgeslagen worden door bij `'BRIEFADAPL'` de naam van de detail-`adapl` op te geven.)

`Wwwopac` genereert automatisch XML tenzij u een detail-`adapl` opgeeft. Gebruik dus alleen een detail-`adapl` als u via `ADAPL HTML` wilt genereren, en dat niet in een `ASPX`-omgeving wilt doen.

Voor een eventuele dataconversie van zoekresultaten voordat die automatisch naar XML worden omgezet, kunt u een `XMLADAPL` opgeven.

DETAILSTYLE (zie `STYLE`)

INVALIDFIELD=<HTML of XML-bestand>

Duidt het HTML of XML-bestand aan dat u de bezoeker wilt tonen als de opgegeven zoekvraag een ongeldige tag of veldnaam bevat.

LANGUAGE=<taalnummer>

De `LANGUAGE`-parameter is niet redundant geworden maar werd in het `www`-bestand van oudere webapplicaties als volgt toegepast: met `LANGUAGE` wordt de standaard gebruikte taal aangegeven en wordt indirect bepaald in welk `ADLIB_DIRi` directory standaard moet worden gezocht naar foutmeldingsbestanden (`i=taalnummer`); de default taal is Engels (`taalnummer=0`). In webpagina's kunt u de gebruiker echter een keuze bieden uit verschillende talen en met die keuze eventuele foutmeldingen dan uit het bijbehorende `ADLIB_DIRi` directory halen door het betreffende taalnummer via de zoekvraag aan `wwwopac` door te geven.

Na `language=` mag ook een tekst komen in plaats van een getal. De eerste vier taalcodes in Adlib kunt u vervangen door de volgende termen:

0	en of eng
1	nl, nld of dut
2	fr, fra of fre
3	de, deu of ger

Bij elke andere term zal in het webconfiguratiebestand worden gezocht naar de bijbehorende taalcode. (Als die definitie er niet is, wordt taal 0 gebruikt.) De genoemde definitie maakt u door bijvoorbeeld de volgende toekenningen in te voegen:

```
es=4
```

```
spa=4
```

Wanneer u nu `language=es` of `language=spa` gebruikt, weet Adlib dat u taal 4 bedoelt.

Veelgebruikte afkortingen voor talen zijn vastgelegd in ISO-standaard 639-2 en 639-1 (zie <http://lcweb.loc.gov/standards/iso639-2/englangn.html>). Voor definities van taalcodes groter dan 3 wordt u aangeraden de standaard taalafkortingen te gebruiken, maar dat is niet verplicht.

LISTADAPL

LISTADAPL is synoniem voor BRIEFADAPL.

NOEXPIRES=TRUE

NOEXPIRES=TRUE specificeert u in het www-bestand van uw webapplicatie om te voorkomen dat de melding "Expires: 0" door de Internet Server wordt gegenereerd voor niet goed werkende POST-requests vanuit NetScape 4.x browsers.

Stel deze optie alleen in als veel van uw internetbezoekers een NetScape browser gebruiken en klagen over problemen met het bladeren naar volgende of vorige pagina's in zoekresultaten, terwijl uw webapplicatie POST-requests gebruikt.

Stel deze optie niet standaard in, omdat het voornamelijk een work-around is voor dit specifieke NetScape-probleem. Als alternatief voor deze optie zou u GET-requests in uw webapplicatie kunnen gebruiken, in plaats van POST's. Om te voorkomen dat complete request-URL's in de browser verschijnen, zou u die requests vanuit een pagina in een framespagina moeten maken.

NORECS=<XML of HTML-bestand>

Dit duidt het HTML of XML-bestand aan dat u de bezoeker wilt tonen als de opgegeven zoekopdracht geen records oplevert. Als u geen NORECS-bestand opgeeft of wanneer het opgegeven bestand niet gevonden, genereert wwwopac een standaard foutmeldingsbestand voor dit doel. Het standaard gegenereerde *norecs.xml* bijvoorbeeld, ziet er als volgt uit:

```
<adlibXML>
<diagnostic>
  <hits>0</hits>
  <error>
    <code>0</code>
    <message>no records found</message>
  </error>
</diagnostic>
</adlibXML>
```

Een mogelijk aanwezige UTF-8 header in een XML foutmeldingsbestand wordt herkend en overgeslagen. (wwwopac herkent in UTF-8 unicode gecodeerde tekstbestanden. Dat houdt in dat in bijvoorbeeld *norecs.xml* diakritische tekens e.d. mogen voorkomen.)

STYLE=<mijnlayout.xslt>

Met de opdracht `STYLE=<mijnlayout.xslt>` werd in oudere webapplicaties de locatie van een XSLT-bestand opgegeven. De Internet Server webapplicatie gebruikte deze stylesheet dan om het XML zoekresultaat naar HTML te transformeren, in www-bestandsyntaxis bijvoorbeeld:

```
STYLE=layout.xslt
```

Gebruik in plaats van `STYLE`, `BRIEFSTYLE` en `DETAILSTYLE` als u voor de lijst- en detailweergave verschillende stylesheets wilt opgeven.

Deze drie parameters mogen zowel in het webconfiguratiebestand als in de CGI-string voorkomen.

Bij gebruik van een van deze drie parameters moet de stylesheet zelf speciale uitvoer genereren als de zoekvraag geen records oplevert. (Het *norecs.html*-bestand wordt dus niet gebruikt in deze situatie.)

SYNTAXERROR=<HTML of XML-bestand>

Dit duidt het bestand aan dat u de bezoeker wilt tonen als de opgegeven zoekvraag een syntaxisfout bevat.

TRANSFORMED=1

TRANSFORMED=1 (in *www*-bestandsyntaxis) betekende in oudere webapplicaties: transformeer het XML zoekresultaat op de server. Als u deze instelling niet maakte, dan werd de waarde achter *...STYLE*= rechtstreeks aan de webclient doorgegeven, en moest dan dus een volledige URL zijn (want anders kon de client het pad niet vinden). De webclient is dan verantwoordelijk voor de transformatie, iets dat niet alle clients kunnen. In Internet Server 3 is **TRANSFORMED**=1, overbodig geworden: alle transformaties vinden automatisch op de server plaats.

Deze parameter mag zowel in het webconfiguratiebestand als in de CGI-string voorkomen.

Als u bij gebruik van *transformed=1* HTML wilt maken (wat meestal zo zal zijn), dan moet u ervoor zorgen dat in het webconfiguratiebestand de instelling *content-type=text/html* staat, en dat de uitvoer van de XSLT begint met *<html* of *<!doctype*. Als dat niet zo wordt gedaan, dan wordt de uitvoer door de browser geïnterpreteerd als XML.

De combinatie *TRANSFORMED=1&DEBUG=1* wordt overigens niet ondersteund.

1.2.2 Selectief *wwwopac*-functionaliteit uitschakelen

Vanaf *wwwopac* versie 6.1.0 kunt u via instellingen in uw webconfiguratiebestand selectief *wwwopac*-functionaliteit uitschakelen. Dit geeft u meer controle over welke informatie via deze software opgevraagd kan worden en op welke manier, en versterkt dus uw veiligheidsbeleid.

U kunt de volgende twee opdrachten voor dit doel zo vaak gebruiken als nodig: *allow* en *disallow*.

U kunt per database (*databaseConfiguration*-sectie in het webconfiguratiebestand) verschillende permissies instellen. Maak de default-instellingen in de *globalConfiguration*-sectie.

Per *databaseConfiguration*-sectie kunt u dus meerdere *allow*- en *disallow*definities geven. Dit is noodzakelijk omdat u namelijk een kleine twintig verschillende waarden aan deze twee opdrachten kunt toekennen, en omdat de lijst met permissies in volgorde van opsomming cumulatief wordt uitgevoerd. Zo hebt u dus veel controle

over wat wel en niet mag. Neem bijvoorbeeld de volgende definitielijst:

```
<disallow>all</disallow>  
<allow>search</allow>  
<allow>scan</allow>
```

Deze opsomming staat gebruikers c.q. de webapplicatie alleen toe normaal te zoeken en een indexscan uit te voeren, verder niets.

De functionaliteitwaarden die u aan de opdrachten kunt toekennen zijn de volgende:

- **Status** - server-informatie opvragen (via `?status=1` in de CGI-aanroep).
- **Scan** - een index doorzoeken, wat resulteert in een lijst met zoek sleutels.
- **Search** - normaal zoeken, zonder `scan`. Dit resulteert in een lijst met records.
- **OAI** - OAI 2.0-functionaliteit.
- **SRU** - het CQL-SRU 1.1-protocol.
- **SDI** - SDI-functionaliteit (alleen voor sommige webapplicaties relevant).
- **CheckPerms** - voor ADW. Alleen wanneer u het ADW-protocol gebruikt om een Adlib-database te benaderen, is het nodig dat u alle "voor ADW"-functionaliteit toestaat via `allow`, anders werkt het niet. De betekenis van de hier opgesomde ADW-functionaliteitwaarden zal verder niet worden besproken.
- **LoadFile** - voor ADW.
- **Locking** - voor ADW.
- **CheckIndex** - voor ADW.
- **PFile** - voor ADW.
- **FreeList** - voor ADW.
- **Sync** - voor ADW en voor het schrijven van records in webapplicaties waarin dat mogelijk is.
- **RunADAPL** - een `adapl` kunnen starten (alleen voor sommige webapplicaties relevant).
- **Thumbnail** - miniatuurweergaven kunnen weergeven (voor webapplicaties).

- **ImageMetadata** - afbeeldingsmetadata opvragen (voor webapplicaties).
- **Delete** - records verwijderen (voor sommige webapplicaties relevant).
- **Test** - gebruik voor interne testdoeleinden door Adlib.
- **All** - alle bovenstaande functionaliteit tezamen.

Als u geen allow of disallow instelt, dan geldt de default-permissie, die impliciet gelijk is aan:

```
<allow>all</allow>
<disallow>test</disallow>
<disallow>sync</disallow>
<disallow>delete</disallow>
```

Dit is redelijk veilig, ofschoon gebruikers zo wel ruim informatie kunnen opvragen via bijvoorbeeld *Status of Checkperms*.

Merk nog het volgende op:

- Deze instellingen zijn niet hoofdlettergevoelig.
- Wanneer een functie uitgeschakeld is, en de gebruiker probeert hem toch toe te passen, dan produceert wwwopac een foutmelding 19.
- Als u deze permissies gebruikt in uw webconfiguratiebestand, dan kunt u de daarin eventueel voorkomende instellingen `WRITE_ALLOWED`, `SYNC_ALLOWED`, en `DELETE_ALLOWED` eruit verwijderen, aangezien ze overbodig zijn geworden. Deze drie opties worden overigens nog wel ondersteund, dus u kunt ze in het webconfiguratiebestand laten staan, als u geen gebruik van `allow` of `disallow` wilt maken.
- Staat veiligheid voorop bij uw implementatie van Adlib Internet Server, dan wordt u geadviseerd de volgende instellingen te gebruiken:

```
<disallow>all</disallow>
<allow>scan</allow>
<allow>search</allow>
<allow>thumbnail</allow>
<allow>imagemetadata</allow>
```

1.2.3 type="richText" in <field> tags

Elk opgesomd veld in `BRIEFFIELDS`, `DETAILFIELDS` of `OAIFIELDS` (zie hoofdstuk 6 voor informatie over OAI) staat in *adlibweb.xml* bij Internet Server 3 apart tussen XML `<field>`-tags. Dit heeft als

voordeel dat u voor specifieke velden de XML tag-attribuut `type="richText"` kunt invoegen om regelovergangen in de inhoud (`\n`) van die velden als XML-regelovergangen over te nemen (`
` tags); zo wordt alle tekst uit die inhoud niet op dezelfde regel achter elkaar geplaatst zoals gewoonlijk het geval is. Merk op dat u dit attribuut normaal gesproken alleen voor niet-RTF-velden wilt gebruiken: als u het namelijk voor een *Rich-text*veld gebruikt dan worden RTF-codes uit de veldinhoud in het `wwwopac.exe` zoekresultaat opgeleverd, maar die codes worden door de Internet Server standaard niet als opmaakkenmerken herkend en dus gewoon tussen de tekst weergegeven.

Een voorbeeld van een aangepaste veldlijst:

```
<OAIFIELDS>
  <field>be</field>
  <field>ob</field>
  <field type="richText">av</field>
</OAIFIELDS>
```

Om in zo'n XML-structuur ineens alle velden uit de datadictionary op te nemen en daar ook `type="richText"` aan toe te kennen, gebruikt u de tag `<allDataDictionaryFields/>`, bijvoorbeeld:

```
<OAIFIELDS>
  <field>be</field>
  <allDataDictionaryFields type="richText"/>
</OAIFIELDS>
```

Hierin is `be` dan een tag die niet in de datadictionary gedefinieerd is.

Merk op dat `
` tags in het XML-zoekresultaat moeten worden afgevangen in een stylesheet om in de opmaak daadwerkelijk een regelovergang te creëren. Daartoe kunt u bijvoorbeeld de volgende template aan een stylesheet toevoegen:

```
<xsl:template match="br">
  <br/>
</xsl:template>
```

1.2.4 Adlib foutmeldingen

Eventuele foutmeldingen van `wwwopac` die normaliter op het scherm van de server worden getoond en een bevestiging verwachten middels klikken op *OK*, worden automatisch afgevangen en gerapporteerd in het XML-zoekresultaat in de `/adlibXML/diagnostic/error/info` node, zodat de webapplicatie op het scherm van een gebruiker op het internet niet schijnbaar kan bevriezen doordat die foutmeldingen daar niet op het scherm worden getoond.

2 Een zoekvraag opgeven

2.1 Opdrachten en parameters in de CGI-string,

De Adlib wwwopac-executable is een server-side CGI-programma. Dat houdt in dat CGI-strings met daarin opgenomen de zoekvraag (aan de client-side) via het mediatype (het MIME-type) 'application/x-www-form-urlencoded' vanuit de browser naar de server worden gestuurd; dit is ook de standaard codering die door HTML-formulieren wordt gegenereerd.

CGI staat voor Common Gateway Interface. Deze interface maakt het mogelijk informatie tussen een webserver en een CGI-programma te transporteren. Een CGI-programma kan ieder programma zijn dat ontworpen is om data, die voldoet aan CGI-eisen, te accepteren en te verwerken.

Een CGI-string wordt in principe (onzichtbaar) door de Internet Server webapplicatie samengesteld, aan de hand van in een HTML-formulier ingevulde zoektermen of aan de hand van klikken op een (afbeeldings)hyperlink naar al bekende records. (Dit gebeurt in gecompileerde code waaraan u niets kunt wijzigen.)

Ook kan een CGI-string direct getypt worden in het URL-invoervak van de browser. (Bepaalde tekens, zoals spaties of aanhalingstekens, moeten daarbij dan wel door een escape-code worden vervangen; zie het onderste deel van de volgende lijst.) U kunt daarbij bijvoorbeeld dieplinken naar de Internet Server 3 webapplicatie zelf, via de syntaxis:

```
http://<webdomein>/<virtuele_directory>/dispatcher.aspx?action=search&database=<databasenaam>&search=<zoekvraag>
```

Bijvoorbeeld:

```
http://demo.adlibsoft.com/InternetServer3/dispatcher.aspx?action=search&database=ChoiceFullCatalogue&search=prirref=28
```

Maar u kunt uw zoekvraag ook direct aan wwwopac voorleggen, en daarmee buiten de Internet Server webapplicatie om werken. Het XML-zoekresultaat wordt dan niet getransformeerd/bewerkt tot een mooie webpagina. De syntaxis is dan:

```
http://<webdomein>/<virtuele_directory>/wwwopac-map/wwwopac.exe?database=<databasenaam>&search=<zoekvraag>
```

Bijvoorbeeld:

`http://demo.adlibsoft.com/InternetServer3/wwwopac/wwwopac.exe?database=ChoiceFullCatalogue&search=ti=ko*`

of

`http://demo.adlibsoft.com/InternetServer3/wwwopac/wwwopac.exe?database=ChoiceFullCatalogue&FLD1=ti&VAL1=ko&TRC1=on`

De "CGI"-string bestaat dus standaard uit een URL die wijst naar een aspx-bestand van de Internet Server webapplicatie, of direct naar *wwwopac.exe* (CGI), en een zoekvraag die ofwel volgens de syntaxis van de Adlib selectietaal* is opgebouwd, of via een alternatieve syntaxis met (veld)naam/waarde-paren (die in Internet Server 3 overigens niet gebruikt wordt). (Zie paragraaf 2.1.1 en de beschrijving van "=" hieronder voor meer informatie over de verschillende syntaxes.)

* Bijzondere selectietaalsyntaxis, zoals de komma, + of - als verkorte schrijfwijze voor OR, AND of AND NOT-combinaties, wordt wel door *wwwopac.exe* maar niet door *dispatcher.aspx* geaccepteerd. Voor zulke bijzondere syntaxis bestaat echter altijd een eenvoudiger equivalent dat door *dispatcher.aspx* wel correct wordt afgehandeld. Een goede manier om zo'n eenvoudig equivalent te verkrijgen is om uw zoekvraag in de selectietaal van een Adlib-applicatie daadwerkelijk uit te voeren: Adlib verandert een bijzondere zoekvraag dan automatisch in een meer uitgeschreven zoekvraag in een eenvoudigere syntaxis. Gebruik dan die zoekvraag als het alternatief voor gebruik met *dispatcher.aspx*.

In het algemeen hebt u de volgende mogelijkheden:

Gereserveerde woorden, tekens, variabelen en opdrachten	
=	<p>Hiermee kent u een waarde toe aan een variabele, bijvoorbeeld: <code>database=ChoiceFullCatalogue</code> of <code>FLD1=au</code> of <code>SEARCH=ti=computer*</code>.</p> <p>Een naam/waarde-paar, zoals in the tweede voorbeeld, bestaat uit een gereserveerd woord of variabele, gevolgd door een logisch te kiezen indexnummer (i), een =-teken, en dan de parameter of waarde, bijvoorbeeld: <code>FLD1=keyword</code> (de veldnaam <code>keyword</code> wordt toegekend aan de variabele <code>FLD1</code>), of <code>VAL1=fiets</code>. Het indexnummer geeft aan welke paren bij elkaar horen; de volgorde waarin de paren in de CGI-string voorkomen, maakt daarom niet uit. Deze syntaxis wordt in Internet</p>

	<p>Server 3 niet gebruikt.</p> <p>De syntaxis die in de Internet Server 3 webapplicatie wordt gebruikt, is: <code>tag=waarde</code>. Tags zijn hoofdlettergevoelig. Veldnamen kunt u in deze syntaxis niet gebruiken. In deze syntaxis moet de zoekvraag beginnen met <code>SEARCH=</code>. Verder gelden in dit geval dezelfde regels voor het samenstellen van uitgebreide zoekvragen als in de selectietaal van Adlib; er kunnen dus ronde haakjes gebruikt worden en Booleaanse operatoren zoals daar normaal is (zonder indexnummer).</p>
?	Scheid de URL (naar het server-side <code>.aspx</code> -bestand, de <code>wwwopac.exe</code> of <code>adlibweb.dll</code>) van de zoekvraag door een vraagteken.
&	Scheid naam/waarde-paren via een ampersand.
BOOL	Een zoekvraag in de naam/waarde-paren syntaxis kan worden uitgebreid via <code>BOOLi</code> . Ken (via <code>=</code>) als parameter een van de booleaanse functies toe: <code>OR</code> , <code>AND</code> , <code>WHEN</code> of <code>NAND</code> . (<code>NAND</code> staat voor 'and not') <code>AND</code> is de standaard operator als er geen wordt opgegeven.
DATABASE	<p>Verwijs hiermee in de CGI-string naar de naam van de database waarin u wilt zoeken. Deze naam moet ook zijn ingesteld in het <code>adlibweb.xml</code>-bestand.</p> <p>Gebruik deze variabele direct na het vraagteken in de CGI-string.</p> <p>(Dezelfde variabele kan in oudere webapplicaties ook aangewend worden om het te gebruiken <code>www</code>-bestand, de voorloper van <code>adlibweb.xml</code>, aan te geven. Laat in de aanroep van het <code>www</code>-bestand de extensie <code>.www</code> dan weg.)</p>
DEBUG	Schakel debug-modus in (<code>DEBUG=1</code>) of uit (<code>DEBUG=0</code>); bij applicatieontwikkeling erg handig. Standaard staat <code>DEBUG</code> uit.

DOM	<p>Met de domeinvariabele <code>DOMi</code> kunt u het domein specificeren waarbinnen in de huidige index gezocht moet worden, bijvoorbeeld: <code>FLD1=name&VAL1=simmons&DOM1=author</code></p> <p>U kunt ook zonder <code>DOM</code> op domeinen zoeken. Neem dan achter <code>VAL</code> het domein in de zoekstring op, voorafgaand aan de gezochte term, gescheiden door twee dubbelepunten, bijvoorbeeld: <code>FLD1=name&VAL1=author::simmons</code></p> <p>Als truncatie aan staat kunt u met <code>DOM</code> ook een domein opgeven, zonder een <code>VAL</code> te verstrekken; alle termen in dat domein van de index worden dan opgehaald.</p>
FLD	<p>Gebruik <code>FLDi</code> om de (database)veldnaam te specificeren waarop gezocht moet worden met waarde <code>VALi</code>. Adlib database-tags (twee-letterige veldcodes) zijn ook toegestaan.</p> <p>Het is mogelijk om op meerdere velden tegelijkertijd te zoeken zonder daar gecombineerde indexen voor aan te maken. Dit kunt u doen door meerdere velden of tags, gescheiden door komma's, in uw CGI-string op te geven.</p> <p>De opdracht <code>FLD1=title,author,keyword</code> zoekt bijvoorbeeld tegelijkertijd in de titel-, auteur- en trefwoordvelden. In feite worden deze meervoudige zoekacties door <code>wwwopac</code> omgezet in een Adlib-zoekopdracht met 'or'-statements:</p> <p><code>http://myhost/myapp/wwwopac.exe?FLD1=title,author,keyword&VAL1=schipper</code></p> <p>resulteert in de volgende Adlib zoekvraag: (<i>title = schipper or author = schipper or keyword = schipper</i>)</p> <p>Hoewel de zoektijd hierdoor toeneemt, kunt u op deze wijze alle soorten velden doorzoeken, geïndexeerd of niet-geïndexeerd, termindexen en woordindexen.</p> <p>Deze naam/waarde-paren syntaxis wordt in de Internet Server 3 webapplicatie overigens niet gebruikt.</p>

LIMIT	Bepaal hoeveel records er per keer van een zoekresultaat opgehaald en getoond worden, bijvoorbeeld: <code>LIMIT=10</code> .
MODE	Met <code>SEARCHMODE</code> specificeert u de default manier waarop meerdere termen in één invoervak worden gecombineerd. Als u echter voor een of enkele van de invoervakken een afwijkende zoekmodus wilt opgeven, gebruikt u daarvoor de variabele <code>MODEi</code> in de CGI-request om per <code>FLDi/VALi</code> -paar een zoekmodus op te geven. Voor <code>MODEi</code> gelden dezelfde mogelijke waarden als voor <code>SEARCHMODE</code> .
OCC	Geef aan op welke occurrences moet worden gesorteerd met <code>OCCi</code> : mogelijke waarden zijn <code>All</code> (er wordt gesorteerd op alle occurrences: voor elke occurrence komt het record dus een keer voor in het zoekresultaat) en <code>First</code> (er wordt alleen op de 1e occurrence van een veld gesorteerd).
OPR	Specificeer met <code>OPRi</code> een operator, anders dan een boolean: <code>contains</code> (bevat), <code>generic</code> , <code>narrower</code> , <code>%3E</code> of <code>></code> , <code>%3C</code> of <code><</code> , <code>%3E%3D</code> of <code>>=</code> , en <code>%3C%3D</code> of <code><=</code> . Let er op de hexadecimale waarden te gebruiken (of eventueel <code>&lt;</code> en <code>&gt;</code> voor resp. <code><</code> en <code>></code>) bij het hard coderen van een zoekvraag in HTML. (Dit omdat <code><</code> en <code>></code> al gereserveerde tekens zijn in HTML.) <code>contains</code> werkt alleen in het zoekformulier.
SCAN	Genereer een index (lijst met sleutels) via <code>SCAN=veldnaam_of_tag</code> . Dit mogen ook integer en datumvelden zijn. Enkele voorgedefinieerde tags en gereserveerde variabelen* met betrekking tot de gevonden records komen tijdelijk beschikbaar voor gebruik in adapls. In de standaard XML-uitvoer worden de waarden uit die variabelen geplaatst tussen XML-tags.
SCANDOM	Gebruik <code>SCANDOM</code> als u <code>SCAN</code> gebruikt; <code>SCANDOM=domeinnaam</code> beperkt zoeken via <code>SCAN</code> tot het opgegeven domein.
SCANEND	U kunt <code>SCANEND</code> gebruiken om aan te geven tot en met welke waarde u wilt zoeken (met <code>SCAN</code>), bijvoorbeeld: <code>SCANEND=waarde</code> .

SCANUNIQUE	SCANUNIQUE=1: standaard, dus zonder dat u deze instelling opgeeft, filtert <code>SCAN</code> alle dubbele termen uit een index voordat die opgeleverd wordt. Alhoewel de naamgeving van <code>SCANUNIQUE</code> wat verwarrend kan zijn, stelt u hiermee juist in dat het filteren van dubbele termen <i>niet</i> gebeurt; alle termen worden dus uit de index opgehaald. (Deze optie wordt voornamelijk alleen in ADW-toepassingen gebruikt.)
SCANVAL	Specificeer de waarde waarmee de lijst met termen zal starten, via <code>SCANVAL=waarde</code> (automatisch getrunceerd). Als u niets opgeeft, dan start de lijst met de eerste waarde in de index. Enkele voorgedefinieerde tags en gereserveerde variabelen* met betrekking tot de gevonden records komen tijdelijk beschikbaar voor gebruik in adapts. In de standaard XML-uitvoer worden de waarden uit die variabelen geplaatst tussen XML-tags. U kunt <code>SCAN</code> overigens ook zonder <code>SCANVAL</code> gebruiken. <code>SCAN</code> en <code>SCANVAL</code> werken niet in combinatie met <code>SEARCH</code> .
SCANXMLFORCE	SCANXMLFORCE=1: wanneer u een <code>SCAN</code> uitvoert, hebt u normaal gesproken geen keuze voor wat betreft het <code>XMLTYPE</code> : de XML wordt in een 'scan-xmltype' opgeleverd waarin eigenlijk vooral metadata staat. Voor sommige toepassingen (momenteel alleen ADW) is het handig als er met de <code>SCAN</code> meer informatie uit de records wordt opgehaald om bijvoorbeeld een hiërarchie te kunnen weergeven en om niet opnieuw records te hoeven ophalen als u alle records voor alle sleutels wilt weergeven. Daarvoor is het handig als u voor een <code>SCAN</code> toch een al ingesteld <code>XMLTYPE</code> kunt forceren in het resultaat: u doet dat met <code>SCANXMLFORCE=1</code> .

SEARCH	<p>In Internet Server 3 worden zoekvragen opgegeven in de vorm (syntaxis) waarin ze in de Adlib selectietaal worden opgebouwd; laat de betreffende zoeklogica (in het <i>Adres</i>-vak in de browser) voorafgaan door <code>SEARCH=</code>. Gebruik wel de hexadecimale weergave van leestekens en gereserveerde tekens.</p> <p>Dat u zoekvragen structureert zoals in de Adlib selectietaal betekent overigens ook dat u grotendeels dezelfde mogelijkheden hebt.</p> <p>Bij het coderen van een CGI-string via naam/waarde-paren wordt de zoekvraag impliciet al samengesteld als een Adlib search en hier mag u de opdracht <code>SEARCH</code> dus niet gebruiken.</p>				
SEARCHMODE	<p>In een search met naam/waarde-paren kunt u sinds <code>wwwopac 4.5.3</code> in elke <code>VAL</code> ook meer dan één term opgeven, met daartussen spaties, en/of plus- of mintekens (zie verderop voor uitleg). Het doel van iets dergelijk is bezoekers van uw web-applicatie (ook) via een enkelvoudig invoervak te laten zoeken, zoals zij dat misschien ook in zoekmachines op het internet doen. In die zoekmachines kunnen in één invoervak meerdere termen opgegeven worden, gescheiden door spaties en/of plus- of mintekens.</p> <p>Vóór <code>wwwopac 4.5.3</code> gebruikte u in plaats van <code>VAL</code>, <code>WQY</code>.</p> <p>Vanaf <code>wwwopac 4.7</code> doet u dit echter anders. De <code>WQY</code>-variabele is verwijderd om plaats te maken voor de (optionele) variabele <code>SEARCHMODE</code>. U gebruikt deze als volgt:</p> <table border="1" data-bbox="400 1107 1037 1426"> <thead> <tr> <th data-bbox="400 1107 692 1155">Syntaxis</th> <th data-bbox="692 1107 1037 1155">Betekenis</th> </tr> </thead> <tbody> <tr> <td data-bbox="400 1155 692 1426"><code>SEARCHMODE=weband</code></td> <td data-bbox="692 1155 1037 1426">Alle opgegeven waarden moeten in één of (verspreid over) meerdere occurrences van het gezochte veld van een record voorkomen. U kunt geen plustekens gebruiken, maar wel mintekens.</td> </tr> </tbody> </table>	Syntaxis	Betekenis	<code>SEARCHMODE=weband</code>	Alle opgegeven waarden moeten in één of (verspreid over) meerdere occurrences van het gezochte veld van een record voorkomen. U kunt geen plustekens gebruiken, maar wel mintekens.
Syntaxis	Betekenis				
<code>SEARCHMODE=weband</code>	Alle opgegeven waarden moeten in één of (verspreid over) meerdere occurrences van het gezochte veld van een record voorkomen. U kunt geen plustekens gebruiken, maar wel mintekens.				

	SEARCHMODE=webor	Ten minste één van de opgegeven waarden moet in het gezochte veld van een record voorkomen. Gebruik eventueel plus- en/of mintekens om nauwkeuriger te zoeken (zie het overzicht hieronder).
	SEARCHMODE=	Als u een lege SEARCHMODE specificeert dan wordt de gehele invoer als één lange string beschouwd, en wordt daarop gezocht. (Dit is de 'klassieke' manier van zoeken met behulp van VAL-termen voor wwwopac.)
Tekens in invoervak Betekenis		
	+	De waarden voor en na een + moeten tegelijk in het zoekresultaat voorkomen. Die waarden kunnen uit één of meer occurrences afkomstig zijn.
	-	De waarde na een - mag niet in het zoekresultaat voorkomen.
	<spatie>	Gebruik een spatie om verschillende waarden van elkaar te scheiden.
	` ,	Gebruik enkele of dubbele aanhalingstekens rond een waarde om een letterlijke tekst te zoeken waarin spaties voorkomen – een term die uit meerdere woorden bestaat bijv.: "OAI repository".

	<p>U kunt met <code>WEBAND</code> en <code>WEBOR</code>, net als met 'klassiek' zoeken, zowel getrunceerd als ongetrunceerd zoeken.</p> <p>Als u voor één of enkele van de invoervakken een afwijkende zoekmodus wilt opgeven, gebruikt u daarvoor de variabele <code>MODEi</code>.</p> <p>Op uw website kunt u bezoekers door middel van <code>SEARCHMODE</code> de mogelijkheid geven zelf een zoekmethode met de syntaxis van een webzoekmachine te kiezen via een knop of het aankruisen van een keuzerondje. <code>SEARCHMODE</code> geldt dus voor alle naam/waarde-paren.</p> <p>(Bij het zoeken op meerdere waarden in één veld moet u in een CGI-string die u in het invoervak <i>Adres</i> van uw browser invoert, de escape-codering van +, -, spaties en aanhalingstekens zelf toepassen.)</p>
SEQ	<p>De standaard rangschikking van records is oplopend (<i>ascending</i>). Kies in een search met naam/waarde-paren voor het alternatief via <code>SEQi=descending</code>.</p> <p>Het indexeringsnummer hier staat los van <code>FLD</code> en <code>VAL</code> indexnummers. <code>SEQ1</code> hoort dus niet bij <code>FLD1</code>, bijvoorbeeld. Wel groepeert u met dit nummer <code>SEQ</code>, <code>SRT</code> en <code>TYP</code>. Met <code>SRT</code> kunt u bijvoorbeeld aangeven het zoekresultaat op auteur te willen sorteren en met <code>SEQ</code> dan de rangschikking bepalen.</p> <p><code>SEQ</code> kan niet gebruikt worden zonder ook een <code>SRT</code> op te geven.</p>
SRT	<p>Sorteer de zoekresultaten op een veldnaam of tag via <code>SRTi</code>, in een search met naam/waarde-paren. Sorteren vertraagt een zoekopdracht overigens aanzienlijk!</p> <p>Het indexeringsnummer hier staat los van <code>FLD</code> en <code>VAL</code> indexnummers. <code>SRT1</code> hoort dus niet bij <code>FLD1</code>, bijvoorbeeld. Wel groepeert u met dit nummer <code>SEQ</code>, <code>SRT</code> en <code>TYP</code>, alhoewel <code>SRT</code> ook alléén kan worden opgegeven. Met <code>SRT</code> kunt u bijvoorbeeld aangeven het zoekresultaat op auteur te willen sorteren en met <code>SEQ</code> dan de rangschikking bepalen.</p>

	<p>Bijvoorbeeld:</p> <pre>.../wwwopac.exe?database=ChoiceFullCatalogue&FLD1=ti &VAL1=ko&TRC1=on&SRT0=author.name&SEQ0=descending</pre> <p>In de <diagnostic> van het XML zoekresultaat vindt u deze zoekvraag als volgt terug:</p> <pre><search>'ti' = "ko*"</search> <sort>sort 'author.name' descending</sort></pre> <p>Als u op meer dan één waarde wilt sorteren, dan krijgt de eerste SRT-variabele bijvoorbeeld indexnummer '1', de tweede het nummer '2', enz.</p>
STARTFROM	<p>In de webapplicatie wordt gewoonlijk maar een deel van een index weergegeven. Om het volgende of vorige deel van de lijst op halen, moet u STARTFROM gebruiken.</p> <p>Specificeer de numerieke waarde vanaf waar moet worden gestart met zoeken, waarin '1' de eerste sleutel ná zoeken op SCANVAL is. U kunt voorwaarts zoeken (STARTFROM = positieve waarde) of achterwaarts (STARTFROM = negatieve waarde).</p> <p>Gebruik de LIMIT-waarde om de STARTFROM-waarde voor de nieuwe pagina te berekenen.</p>
TRC	<p>Rechtstruncatie staat standaard uit. Schakel het aan voor het naam/waarde-paar i via TRCi=on. U kunt een waarde echter ook trunceren door een * achter het (eerste deel van een) woord te zetten.</p> <p>Met de opdracht TRCi=all in combinatie met een search op een lege VALi worden álle records uit de database opgehaald. TRCi=all werkt alleen voor velden met termindexen. Gebruik hiervoor dus bijvoorbeeld de volgende zoekvraag:</p> <pre>FLD1=auteur&VAL1=&TRC1=all</pre> <p>De volgende zoekvraag zal overigens geen enkel record ophalen:</p> <pre>FLD1=auteur&VAL1=&TRC1=on</pre> <p>U kunt met TRCi=all of TRCi=on dus bepalen of gebruikers met een lege zoek sleutel al dan niet alle records uit de database kunnen opvragen.</p>

TYP	<p>Te gebruiken naast <code>SRT</code>, om het type van de sortering aan te geven. Sorteert het zoekresultaat alfabetisch (default), numeriek, of op datum (voor een datumveld) via <code>TYPi=t</code>, <code>TYPi=n</code>, of <code>TYPi=d</code> respectievelijk.</p> <p>Het indexnummer moet overeenkomen met die van de bedoelde <code>SRT</code>. Dit is niet geassocieerd met de indexnummers voor <code>FLD</code> en <code>VAL</code>.</p>
VAL	<p>Met <code>VALi</code> specificeert u de waarde waarop gezocht moet worden, in het veld <code>FLDi</code>. Zie de uitleg bij <code>FLD</code> en <code>TRC</code> voor meer informatie.</p>
<p>Schrijf gereserveerde woorden in de zoekvraag altijd in hoofdletters, tenzij anders aangegeven. Verder zijn alleen de veldnamen en tags hoofdlettergevoelig; schrijf deze precies zoals ze in de database voorkomen. Voor de andere parameters en waarden maakt het gebruik van boven- en onderkast niet uit.</p>	
<code><%HH></code>	<p>Gereserveerde tekens en leestekens (alle niet-alfanumerieke tekens) in namen en waarden die aan de CGI-string worden meegegeven moeten vervangen worden (escaped) door een procentteken met twee hexadecimale cijfers; zie de voorbeelden hieronder:</p>
<code><spatie></code>	<p>Spaties in formulier- en/of veldnamen en waarden moeten worden vervangen door een <code>+</code>. Maar <code>%20</code> in plaats van <code>+</code> werkt ook. Gebruik bij voorkeur niet letterlijk spaties.</p>
<code>\n</code>	<p>Vervang in waarden met meerdere regels tekst <code>\n</code> (CR LF: harde return) door <code>%0D%0A</code>.</p>
<code>&</code>	<p>Het <code>&</code>-teken (ampersand) in namen en waarden moet worden vervangen door <code>%26</code></p>
<code>,</code>	<p>Komma's in namen en waarden moeten worden vervangen door <code>%2C</code>.</p>
<code>%</code>	<p>Een <code>%</code>-teken in tekst wordt gecodeerd als <code>%25</code>. Dus de Adlib tag voor recordnummer (<code>%0</code>) moet worden gecodeerd als <code>%250</code>: <code>FLDi=%250</code>. U kunt ook de veldnaam gebruiken: <code>preref</code>.</p>
<code>+</code>	<p>Een plusteken (bijvoorbeeld voor gebruik in <code>SEARCHMODE</code>-invoer) in een CGI-string moet worden</p>

	gecodeerd als %2B.
-	Een minteken (bijvoorbeeld voor gebruik in SEARCHMODE -invoer) in een CGI-string moet worden gecodeerd als %2D.
'	Een enkel aanhalingsteken wordt gecodeerd als %27.
"	Een dubbel aanhalingsteken wordt gecodeerd als %22.

* SCAN-resultaten

De voorgedefinieerde tags en gereserveerde variabelen in het tijdelijke bestand dat de scan-functie oplevert, zijn (voor elk veld):

- &B[1]:** het aantal records in de selectie (dus na een eventueel toegepaste limiet).
In XML-uitvoer staat deze waarde tussen de tags `<hits_on_display>` en `</hits_on_display>`.
- &B[2]:** geeft normaal het totaal aantal hits, maar omdat wwwopac stopt met zoeken na de opgegeven limiet, zal deze variabele altijd gelijk aan de limiet zijn.
In XML-uitvoer staat deze waarde tussen de tags `<hits>` en `</hits>`.
- &B[3]:** bevat de waarde uit STARTFROM.
In XML-uitvoer staat deze waarde tussen de tags `<first_item>` en `</first_item>`.
- &B[4]:** beantwoordt de vraag "zijn er nog waarden in de lijst die aan de eerste waarde in deze selectie voorafgaan?" met 0 (nee) of 1 (ja).
In XML-uitvoer staat deze waarde tussen de tags `<backward>` en `</backward>`.
- &B[5]:** beantwoordt de vraag "zijn er nog waarden in de lijst die volgen op de laatste waarde in deze selectie?" met 0 (nee) of 1 (ja).
In XML-uitvoer staat deze waarde tussen de tags `<forward>` en `</forward>`.
- &I:** (&hoofdletter i) Voor elk gevonden record in de selectie (na de limiet) krijgt &I een volgnummer. Voor het eerste gevonden record na zoeken met

SCAN wordt &I=1, de volgende &I=2, enzovoort. Met het ophalen van een nieuwe selectie (via bijvoorbeeld knoppen *Volgende* of *Vorige*) wordt aan &I voor de nieuwe records op dezelfde wijze volgnummers toegekend: voor het eerste gevonden record uit de nieuwe selectie wordt &I dus weer 1. In de XML-uitvoer van wwwopac 6.2 en hoger komt dit volgnummer niet meer voor.

- <TE>**: TE bevat steeds een term of naam die tijdens het uitvoeren van SCAN gevonden is.
- <NT>**: nauwere term, kan meerdere occurrences bevatten, die benaderbaar zijn via NT[i], waarin i het nummer van de occurrence is.
- <BT>**: ruimere term, kan meerdere occurrences bevatten, die benaderbaar zijn via BT[i], waarin i het nummer van de occurrence is.
- <US>**: voorkeursterm, kan meerdere occurrences bevatten, die benaderbaar zijn via US[i], waarin i het nummer van de occurrence is.
- <UF>**: niet-voorkeursterm, kan meerdere occurrences bevatten, die benaderbaar zijn via UF[i], waarin i het nummer van de occurrence is.
- <RT>**: gerelateerde term, kan meerdere occurrences bevatten, die benaderbaar zijn via RT[i], waarin i het nummer van de occurrence is.
- <priref>**: bevat de priref, het nummer van het term- of naamrecord in de gekoppelde database, waarin de gevonden term zich bevindt.
-

2.1.1 Drie verschillende syntaxes

Of de zoekvraag nu wordt ingevoerd in het vak *Adres* van een browser, of dat de vraag wordt samengesteld aan de hand van gebruikersinvoer, of dat deze hard wordt gecodeerd in HTML, maakt voor de codering van de complete URL niets uit. U kunt echter kiezen uit drie verschillende syntaxes:

1. Zoals eerder getoond, kunt u werken met variabele-paren
FLDi=naam&VALi=waarde&...
Bij het gebruik van deze syntaxis (en de tweede hieronder) worden booleaanse operatoren van voren naar achteren

uitgevoerd; u kunt geen haakjes gebruiken om een andere volgorde aan te geven.

2. U kunt waarden ook direct aan tags (niet aan de veldnaam) toekennen: `tag=waarde&...` Tags zijn hoofdlettergevoelig. Syntaxis 1 en 2 kunnen overigens door elkaar heen gebruikt worden.
3. Als u liever zoekvragen opgeeft in de vorm waarin ze in de Adlib selectietaal worden opgebouwd dan kan dat ook: laat de betreffende zoeklogica dan voorafgaan door `SEARCH=`.
Bijvoorbeeld: `.../wwwopac.exe?DATABASE=mijndb&SEARCH=all`
of bijvoorbeeld
`.../wwwopac.exe?DATABASE=mijndb&SEARCH=(author.name%20%3D%20jansen*)`

U kunt ook haakjes gebruiken, pointerfiles openen en booleaanse vergelijkingen nesten en in de gewenste volgorde laten uitvoeren. Vervang leestekens en gereserveerde tekens wel door hun hexadecimale waarden met een %-teken ervoor.

Met het `SEARCH=` statement kunt u dus alles doen wat u in een Adlib-toepassing ook kunt, zoals:

- gecombineerde selecties maken;
- selecties met haakjes;
- selecties met de `$today`-variabele;
- afdrukken (op de server);
- pointerfiles openen.

Merk nogmaals op dat sommige bijzondere selectietaalsyntaxis wel door `wwwopac.exe` wordt geaccepteerd maar niet door `dispatcher.aspx`. Zie de volledige uitleg aan het begin van dit hoofdstuk.

2.1.2 SEARCH en FLD/VAL-paren in één zoekvraag

De drie mogelijke syntaxis van een zoekvraag kunnen door elkaar gebruikt worden. U kunt dus `FLD/VAL`-paren gebruiken en in dezelfde zoekvraag ook een `SEARCH=` opnemen, normaal gescheiden door &-tekens.

Om beide delen te combineren, kan de operator `SEARCHBOOL` worden gebruikt. Hier kunt u de waarden `OR` of `AND NOT` aan toekennen; `AND` is de default als beide syntaxis voorkomen, dus dan hoeft u `SEARCHBOOL` niet op te geven.

Omdat bij het interpreteren van een zoekvraag door Internet Server niet kan worden uitgemaakt of het `SEARCH=` gedeelte links of rechts van de `FLD/VAL`-paren staat, als ze gecombineerd moeten worden met `AND`

NOT, wordt standaard verondersteld dat het SEARCH-gedeelte links in de vergelijking staat. Een FLD/VAL-zoekvraag AND NOT een SEARCH-zoekvraag is dus niet mogelijk.

2.1.3 Pointerfiles openen

Via wwwopac kunt u ook pointerfiles openen. Pointerfiles zijn handig om de bezoeker van uw website een selectie van records voor te schotelen, in de vorm van een galerij bijvoorbeeld. In een pointerfile kunt u regelmatig een nieuwe selectie opslaan (vanuit uw eigen Adlib-toepassing) en als u het bestand onder hetzelfde pointerfilenummer bewaart, dan hoeft de hard gecodeerde zoekvraag in de HTML of ASPX-pagina niet eens te worden aangepast om wel de nieuwe galerij te tonen.

Zie voor het maken van pointerfiles de Adlib Gebruikersgids.

Gebruik in de zoekvraag voor wwwopac eenvoudig:

```
SEARCH=pointer%20#
```

Vul voor # het nummer van het gewenste pointerfile in (de naam ervan opgeven is niet mogelijk). %20 vervangt de spatie in CGI-strings. Wwwopac zoekt de pointerfile in een voor pointerfiles gereserveerde submap (met de naam van de database) in de \data-map van uw applicatie. U kunt de voor wwwopac bestemde pointerfiles ook ergens anders opslaan, maar dan moet u wwwopac wel vertellen waar de pointerfiles gevonden kunnen worden. U doet dat via de variabele ADLIB_PTR. Ken aan deze variabele de padnaam naar de pointerfiles toe, in de zoekvraag of in het webconfiguratiebestand.

Als u bij het openen van een pointerfile stuit op foutcode 135 dan kan de pointerfile niet geopend worden. Dit betekent waarschijnlijk dat de pointerfile op dat moment in gebruik is, of dat hij niet bestaat.

2.1.4 isutf8=1

Als de toekenning isutf8=1 als argument aan de CGI-string wordt meegegeven, dan wordt alle tekst in die string als UTF-8 gecodeerd; zonder die toekenning wordt alle tekst als WinLatin1 geïnterpreteerd. Voor wwwopac.exe/adlibweb.dll is het nodig dat de CGI-string in één bepaalde codering wordt aangeboden (of dat nu UTF-8 is of ASCII). Maar MS Internet Explorer is met het coderen van CGI-strings voor het maken van een GET of POST-request niet helemaal consequent als er tekens uit een andere tekenset (bijv. Japans) worden opgegeven: de hele string wordt in UTF-8 gecodeerd, behalve accenttekens uit de huidige karakterset – in West-Europa is dat bijvoorbeeld WinLatin1

(ISO-8859-1). (En technisch gezien kunnen de accenttekens in WinLatin1 helaas niet van de UTF-8 gecodeerde tekens worden onderscheiden.)

Als u denkt dat gebruikers alleen maar westerse tekens zullen invoeren, dan kunt u `isutf8=1` achterwege laten, maar de praktijk kan natuurlijk anders blijken en dan problemen geven.

Als u wilt dat alle mogelijke tekens in de zoekstring worden geaccepteerd, gebruik dan `isutf8=1` om de CGI-string als UTF-8 te interpreteren. In de Internet Server 3 modelwebapplicatie wordt deze instelling standaard meegegeven, en dat kunt u ook niet aanpassen.

Om deze omzetting in een oudere Internet Server webapplicatie te gebruiken, moet u de volgende aanpassingen maken:

Zet in de HTML-pagina na de tag `<html>` de volgende code:

```
<meta http-equiv="Content-Type" content="text/html; charset=us-ascii">
```

en zet in het zoekformulier:

```
<input type="hidden" name="isutf8" value="1">
```

(Als `us-ascii` problemen oplevert, kunt u dat eventueel vervangen door `ISO-8859-1`.)

Voor webapplicaties die in C# zijn geschreven, móet u `isutf8=1` meesturen, omdat C#-functies altijd UTF8-codering toepassen.

2.2 Afbeeldingen ophalen

2.2.1 Dynamisch thumbnails maken

Afbeeldingen van items in uw collectie wilt u in de lijstweergave van een zoekresultaat vaak via thumbnails (miniaturafbeeldingen) in die lijst tonen. U hoeft hiervoor niet apart miniaturen van elke afbeelding in uw database te bewaren. De `wwwopac` kan de normale afbeeldingen in uw database namelijk dynamisch verkleinen.

Binnen een HTML image-tag kunt u daartoe aan de `src`-attribuut ervan een opdracht aan `wwwopac` meegeven. De naam van de afbeelding die moet worden opgehaald, kan de Internet Server webapplicatie aan het momentele zoekresultaat onttrekken. (In statische HTML-pagina's staat achter de `src`-attribuut meestal de URL naar een afbeelding.)

U kunt in diezelfde opdracht ook de afmetingen van de te maken thumbnail bepalen, en het uitvoertype van de miniatuur (hoeft niet

hetzelfde te zijn als het type van de oorspronkelijke afbeelding). Het voordeel is dat de verkleinde afbeelding naar de client wordt gestuurd: doordat de kleinere afbeelding minder geheugen inneemt, hoeven er minder gegevens over het internet te worden gestuurd, waardoor webpagina's sneller worden geladen.

Een HTML-opdracht voor het ophalen van een afbeelding waarvan de naam en locatie bekend zijn, ziet er bijvoorbeeld als volgt uit:

```
<Img src="http://localhost/mywebapp/wwwopac.exe?
thumbnail=../../webimages/test.gif&outputtype=
image/jpeg&xsize=10&ysize=20&dontkeepaspectratio=1">
```

De parameters die u kunt gebruiken bij een thumbnail-query zijn:

XSIZE	XSIZE specificeert de horizontale afmeting van de resulterende afbeelding. YSIZE wordt automatisch bepaald, waarbij de x/y-verhouding (aspect ratio) behouden blijft.
YSIZE	YSIZE specificeert de verticale afmeting van de resulterende afbeelding. XSIZE wordt automatisch bepaald, waarbij de x/y-verhouding (aspect ratio) behouden blijft.
OUTPUTTYPE	Geef het mime-type van de resulterende afbeelding op. Deze hoeft niet hetzelfde te zijn als het origineel. Geldige types zijn onder andere: <ul style="list-style-type: none"> • image/jpeg (standaard instelling) • image/gif • image/bmp
DONTKEEPASPECTRATIO	Als u deze parameter gebruikt en op 1 instelt, dan wordt de originele x/y-verhouding niet behouden. Gebruik deze parameter alleen als u zowel XSIZE als YSIZE opgeeft. De resulterende miniatuurweergave wordt altijd YSIZE hoog en XSIZE breed, in één van beide dimensies waarschijnlijk uitgerekte. Als u toch maar één afmeting opgeeft, wordt van de andere dimensie de originele afmeting genomen.

Maximumgrootte voor thumbnails

Vanaf wwwopac 6.1.0 is er de mogelijkheid om zowel de maximum breedte als de maximum hoogte voor deze miniatuurweergaven te specificeren en tegelijk de oorspronkelijke hoogte/breedte-verhouding van een afbeelding te behouden. U geeft eenvoudig zowel een `XSIZE` als `YSIZE`-waarde op voor de op te halen `thumbnail`, en specificeert geen `DONTKEEPPASPECTRATIO`, of u stelt `DONTKEEPPASPECTRATIO` in op nul. De hoogte/breedte-verhouding blijft dan intact en de miniatuurafbeelding zal altijd worden begrensd door ofwel de opgegeven maximale breedte of de opgegeven maximale hoogte, of door beide als `YSIZE/XSIZE` overeenkomt met de hoogte/breedte-verhouding van de originele afbeelding.

Een achtergrondkleur instellen - Als u voor miniatuurweergaven een maximumgrootte instelt, en de originele afbeelding heeft een andere hoogte/breedte-verhouding, dan zal naast of boven en onder de miniatuurweergave lege ruimte zichtbaar worden. Dit is de achtergrond voor de miniatuurweergave, en die is standaard zwart.

Bij het dynamisch genereren van een miniatuurweergave kunt u echter een andere achtergrondkleur meegeven. U doet dat via de `THUMBNAILBACKCOLOR` parameter. Erachter geeft u een kleur op als een hexadecimale kleurcode (zoals in HTML gebruikelijk is), in het formaat `RRGGBB` (R=rood, G=groen, B=blauw). In dit kleurmodel kan elke kleurcomponent decimaal een waarde hebben van 0 tot een met 255 (er zijn hier dus 256 tinten rood, 256 tinten groen, etc.) De combinatie van drie tinten in deze kleuren geeft de resulterende kleur. De decimale getallen moeten per stuk naar hexadecimaal worden omgezet en achter elkaar worden geplaatst. Decimaal 00 blijft zo 00 in hex, en 255 wordt FF. Voorbeelden: `FF0000` is fel rood, `00FF00` is fel groen, `888888` is grijs. (Gebruik eventueel de Windows Rekenmachine om decimale waarden naar hexadecimaal te converteren.)

In een Internet Server-zoekvraag wordt dit dus bijvoorbeeld:

```
...&thumbnailbackcolor=222222...
```

In de Internet Server 3 webapplicatie worden te genereren thumbnails gespecificeerd in de XSLT stylesheets voor de verschillende lijst- en detailweergaven van museumobjecten, bijvoorbeeld:

```
<xsl:template match="Reproduction/reproduction.identifier_URL">
  <xsl:if test="position() = 1">
    <!-- Only show the first image -->
    
  </xsl:if>
</xsl:template>
```

In de `thumbnail`-parameters kunt u naar wens wijzigingen aanbrengen, bijvoorbeeld om grotere of juist kleinere thumbnails te laten genereren.

2.2.2 Afbeeldingen ophalen in hoge resolutie

Afbeeldingen van items in uw collectie kunt u (verkleind) van de server naar de client sturen via de `thumbnail`-opdracht. Grotere afbeeldingen worden met die opdracht soms echter in een niet-optimale resolutie opgehaald. Daarom zijn er extra parameters beschikbaar om voor het ophalen van grotere afbeeldingen een goede resolutie (lees: kwaliteit) te waarborgen. Met deze parameters kunt u een afbeelding eventueel bijsnijden en dan verkleinen of vergroten, voordat het resultaat naar de client wordt gestuurd. De parameters worden dus gebruikt met `thumbnail=` en zijn de volgende:

FULLIMAGE	<p>FULLIMAGE=1 gebruikt de volledige afbeelding voor het bijsnijden en verkleinen of vergroten en geeft de beste resolutie.</p> <p>FULLIMAGE=0 (default) gebruikt ofwel de volledige afbeelding of automatisch door Windows aangemaakte miniatuurweergaven die bijvoorbeeld in de Verkenner worden gebruikt. Gebruik deze instelling wanneer u miniatuurweergaven wilt genereren.</p>
CROPX, CROPY	<p>Dit zijn de coördinaten, vanaf de linker bovenhoek van de afbeelding gemeten (x is horizontaal), die de linker bovenhoek van de afbeelding zullen vormen als u die wilt bijsnijden. Het getal dat u als coördinaat opgeeft mag in pixels zijn of een percentage van de betreffende dimensie van de volledige afbeelding.</p>

CROPWIDTH, CROPHEIGHT	Dit zijn de afmetingen die de bij te snijden afbeelding moet krijgen, vóór verkleinen of vergroten via XSIZE en YSIZE. Het getal dat u als breedte of hoogte opgeeft mag in pixels zijn of een percentage van de betreffende dimensie van de volledige afbeelding.
CROPABSOLUTE	Met CROPABSOLUTE=1 geeft u aan dat de getallen die u bij de andere CROP-parameters hebt opgegeven absoluut zijn (in pixels dus). CROPABSOLUTE=0 (default) geeft aan dat de betreffende getallen percentages zijn.
XSIZE, YSIZE	De horizontale en verticale afmetingen (in pixels) van de resulterende afbeelding. Als u maar één van beide parameters opgeeft, wordt de andere automatisch berekend voor een gelijkblijvende x/y-verhouding.

Een HTML-opdracht voor het ophalen van een afbeelding waarvan de naam en locatie bekend zijn, ziet er bijvoorbeeld als volgt uit:

```
<Img src="http://localhost/mywebapp/wwwopac.exe?
thumbnail=../../webimages/myimage.gif&outputtype=
image/jpeg&xsize=200&dontkeepaspectratio=0&fullimage=1&crop
x=10&cropy=10&cropwidth=380&cropheight=280&cropabsolute=1">
```

Stel dat *myimage.gif* in dit voorbeeld 400x300 pixels groot is. Het gaat om een foto van een schilderij waar ook de lijst op staat. Wat deze query doet is een rand van 10 pixels van de afbeelding "afsnijden", totdat deze de afmetingen 380x280 heeft (en de lijst niet meer te zien is); vervolgens wordt de afbeelding verkleind naar 200x147 pixels (de y-coördinaat wordt automatisch uitgerekend want de x/y-verhouding moet in dit voorbeeld gelijk blijven). Dit resultaat wordt naar de client gestuurd. Het originele afbeeldingsbestand *myimage.gif* blijft uiteraard onveranderd.

2.2.3 Standaard miniatuurafbeelding

Als bij het tonen van een zoekresultaat een op te halen afbeelding niet gevonden kan worden, wordt deze fout normaliter met een standaard Windows pictogram aangegeven op de plaats waar de afbeelding had moet worden weergegeven. Maar vanaf *wwwopac* 5.0.3 kunt u in uw *adlibweb.xml*-webconfiguratiebestand specificeren welke standaard afbeelding moet worden getoond, als de gezochte afbeelding niet wordt gevonden. Zo kunt u zelf een afbeelding maken die minder de

indruk van een fout geeft, of beter bij de vormgeving van uw site past.

In *adlibweb.xml* definieert u een standaard afbeelding via de instelling: `DEFAULTTHUMBNAIL=<afbeelding.ext>`, dus bijvoorbeeld: `<DEFAULTTHUMBNAIL>museumlogo.jpg</DEFAULTTHUMBNAIL>`.

2.2.4 Getoonde afbeeldingen automatisch watermerken

Als afbeeldingen in de (detail)weergave van records zó groot worden weergegeven dat het voor bezoekers van uw website verleidelijk kan worden de afbeeldingen te kopiëren en mogelijk te reproduceren zonder daarbij de auteursrechten in ogenschouw te nemen, dan kan het wenselijk zijn al die afbeeldingen bij de weergave ervan van een watermerk of copyrightvermelding te voorzien. Aangezien u waarschijnlijk niet het beeldmateriaal zelf wilt gaan aanpassen, is er vanaf *wwwopac 5.0.3* de mogelijkheid uw afbeeldingen standaard te laten watermerken voor weergave in de webbrowser, door een afbeelding van een watermerk of copyrightvermelding steeds over een afbeelding uit een record heen af te beelden, al dan niet met een zekere mate van transparantie en met een door u ingestelde grootte en positie ten opzichte van een onderliggende afbeelding uit het record. De hieronder beschreven instellingen geven u veel mogelijkheden. (De afbeelding van het watermerk of de copyrightsvermelding wordt hier een "overlay" genoemd.) Merk op dat u die instellingen alleen in *adlibweb.xml* kunt gebruiken en niet in CGI-strings (om te voorkomen dat het watermerken vanaf de opdrachtregel in de webbrowser kan worden uitgeschakeld).

IMAGEOVERLAYFILE=<bestandsnaam>

Geef (het pad en) de bestandsnaam op van de afbeelding die u als overlay wilt gebruiken. Als u een relatief pad wilt gebruiken, neem dan het relatieve pad ten opzichte van uw *\data-map*. Voorbeeld:

```
<imageoverlayfile>..\imgs\copyright.jpg</imageoverlayfile>
```

IMAGEOVERLAYBLEND=<1-100>

Geef hier de mate van ondoorzichtigheid van de overlay op: met een percentage van 100 is de overlay volkomen ondoorzichtig, terwijl een laag percentage de overlay bijna geheel transparant maakt.

IMAGEOVERLAYPOSITION=<0-4>

Met de vijf mogelijk waarden bepaalt u de positie van de overlay ten opzichte van de onderliggende afbeelding (want de overlay kan kleiner zijn): 0 = linksboven, 1 = rechtsboven, 2 = rechtsonder, 3 = linksonder, en 4 = gecentreerd.

IMAGEOVERLAYPERCENTAGE=0 of <1-100>

Hiermee bepaalt u de relatieve grootte van de overlay ten opzicht van de onderliggende afbeelding. Kies 0 (de default-waarde) als u de overlay-grootte nooit wilt aanpassen aan de grootte van onderliggende afbeelding. Kies een waarde (percentage) tussen 1 en 100 om de overlay te laten vergroten of verkleinen ten opzichte van de grootte van de weer te geven afbeelding eronder. Met 100 wordt de overlay even groot gemaakt als de afbeelding; met elke lagere waarde wordt de overlay dus percentueel kleiner dan die afbeelding.

IMAGEOVERLAYMINPIXELSIZE=<aantal pixels>

Als u de overlay alleen wilt gebruiken voor grote afbeeldingen, en niet voor kleine afbeeldingen of miniatuurweergaven, dan kunt u via deze instelling de minimale grootte van een weer te geven onderliggende afbeelding in het totale aantal pixels opgeven, waarop een overlay moet worden toegepast. Het gaat hierbij niet om de originele afmetingen van de op te halen afbeelding, maar om de afmetingen waarin die wordt weergegeven in de detail- of lijstweergave. Geef via een geheel getal het totale aantal pixels op. Als u 0 opgeeft (default-waarde) dan wordt de overlay altijd toegepast.

Stel dat u uw overlay alleen wilt gebruiken bij afbeeldingen die groter zijn dan de miniatures ervan in de lijstweergaven. Als uw miniatures 10x10 pixels zijn, dan moet u dus een waarde van 101 opgeven voor de hier besproken instelling.

Weergegeven afbeeldingen van bijvoorbeeld 5x75 of 250x200 pixels worden dan dus van de overlay voorzien, maar afbeeldingen van bijvoorbeeld 8x9 of 10x10 pixels worden zonder overlay getoond.

2.3 Imagemetadata

Met de opdracht `imagemetadata=<bestandsnaam_afbeelding>` kunt u metadata van afbeeldingen opvragen. Dit betreft gegevens zoals de datum waarop de afbeelding werd opgeslagen en de bestandsgrootte e.d. Voor de bestandsnaam kunt u het best een relatieve padnaam opgeven (relatief t.o.v. de hoofdmap van de Internet Server). Het pad hoeft niet tussen aanhalingstekens, en u kunt zowel / als \ gebruiken in het pad, bijvoorbeeld:

```
...wwwopac.exe?imagemetadata=images/test.gif
```

of voor meerdere afbeeldingen tegelijk:

...wwwopac.exe?imagemetadata=images/*.gif

Het resultaat voor één gevonden afbeelding ziet er dan bijvoorbeeld als volgt uit:

```
<?xml version="1.0" encoding="UTF-8" ?>
<imageMetaDataSet>
  <imageMetaData>
    <FILE>test.gif</FILE>
    <SIZE>2150</SIZE>
    <CREATIONDATE>27/03/2005</CREATIONDATE>
    <CREATIONTIME>16:11:39</CREATIONTIME>
    <WRITEDATE>24/01/2006</WRITEDATE>
    <WRITETIME>11:45:38</WRITETIME>
    <ACCESSDATE>07/06/2006</ACCESSDATE>
    <ACCESSTIME>15:44:10</ACCESSTIME>
    <PATH>D:\demo.adlibsoft.com\IS3\images\test.gif</PATH>
  </imageMetaData>
</imageMetaDataSet>
```

Voor meer informatie over image metadata, zie de Adlib Designer Help.

3 OAI (Open Archives Initiative)

Het OAI-protocol, gebaseerd op HTTP en XML, is een metadata harvesting protocol, bedoeld om de voor zoekmachines normaal gesproken onzichtbare inhoud van internetdatabases voor die zoekmachines te openen en doorzoekbaar te maken via een opgegeven term.

De Adlib *wwwopac.exe* en *oaiserver.exe* ondersteunen OAI-protocol 2.0. Via een OAI-zoekvraag (Eng.: OAI protocol request) aan een zogenaamd repository, een server waarop bijvoorbeeld *wwwopac* geconfigureerd is voor het OAI-protocol, kan (meta)data uit databases onttrokken worden (ook Adlib-databases), en beschikbaar worden gesteld op het internet waar het door service-providers zoals zoekmachines geïndexeerd kan worden.

Een volledige beschrijving van het OAI-protocol vindt u op <http://www.openarchives.org>

Het verschil tussen data en metadata in een Adlib-database is een kwestie van afspraak. We beschouwen de gegevens in de database normaal gesproken als data, maar de beschrijving van een object of boek, kan evengoed als metadata worden beschouwd aangezien bijvoorbeeld niet de inhoud van een boek zelf in de database te vinden is. In Adlib-databases wordt informatie óver andere gegevens c.q. voorwerpen bewaard: metadata dus.

De metadata die een OAI-zoekvraag aan een Adlib-database oplevert, kan voldoen aan een aantal gespecificeerde standaarden. De Dublin Core metadata-standaard is de standaard die wordt meegeleverd in onze implementatie van OAI. Zie <http://www.openarchives.org/OAI/openarchivesprotocol.htm#dublincore> voor het XML-schema hiervan, en <http://dublincore.org/documents/1999/07/02/dces/> voor een meer leesbare specificatie van Dublin Core.

Die standaard bestaat uit 15 zogenoemde elementen (vergelijkbaar met Adlib velden) waarin de metadata wordt doorgegeven. Aangezien Adlib-databases veel meer velden bevatten dan Dublin Core elementen heeft, wordt er dus een beperkte hoeveelheid informatie uit een opgehaald Adlib-record geselecteerd. U mag echter ook meer dan 15 elementen definiëren. Dublin Core is eigenlijk een smalle basis die iedereen kan ondersteunen; dit maakt uitwisseling tussen verschillend gestructureerde data gemakkelijker.

In principe geeft *wwwopac* of *oaiserver* alle velden uit een Adlib-record terug (in Adlib XML-formaat). Speciaal voor OAI gebruikt

wwwopac/oaiserver echter een XSLT-stylesheet (met daarin een Adlib-veld naar Dublin Core-element mapping) die in het webconfiguratiebestand is ingesteld om dat zoekresultaat te transformeren naar het toepasselijke metadataformaat (ook in XML), alvorens het naar de harvester te sturen. Het metadata zoekresultaat wordt aldus getransformeerd tot een zogenaamd OAI-record (per Adlib-record één OAI-record). Zo'n record bestaat voornamelijk uit een *header* en de *metadata*. De header bestaat uit een unieke identifier voor het opgehaalde record, en uit een datumstempel die aangeeft wanneer het record voor het laatst is gewijzigd. De metadata is uiteraard de opgehaalde data na transformatie.

Het outputformaat van een OAI-zoekresultaat wordt dus bepaald door een XSLT-stylesheet op de server. Onze standaard Dublin Core stylesheet voor dit doel is: *oai_dc.xsl*. Eventuele andere uitvoerformaten kunt u op deze stylesheet baseren.

3.1 Instellingen voor het adlibweb.xml-bestand

In *adlibweb.xml* moet het gebruik van OAI worden geconfigureerd – voor deze configuratie kan geen *www*-bestand worden gebruikt. In *adlibweb.xml* voor *wwwopac.exe* bij Internet Server 3 vindt u standaard waarschijnlijk nog geen OAI-configuratie; in *adlibweb.xml* voor *oiserver.exe* echter, vindt u alléén een OAI-configuratie. Voor de Internet Server webapplicatie moet u deze configuratie dus toevoegen. Pas het onderstaande voorbeeld van een complete OAI-configuratie daartoe aan uw eigen situatie aan en voeg het toe aan *adlibweb.xml*.

```
<OAIConfiguration>
<databasepath>..\..\data</databasepath>
<database>collect</database>
<OAI_REPOSITORY_NAME>Mijn Museum</OAI_REPOSITORY_NAME>
<OAI_ADMIN_EMAIL>oai@mijnmuseum.nl</OAI_ADMIN_EMAIL>
<OAI_ADMIN_EMAIL>admin@mijnmuseum.nl</OAI_ADMIN_EMAIL>
<OAI_REPOSITORY_IDENTIFIER>mijnMuseum</OAI_REPOSITORY_IDENTIFIER>
<OAI_DELIMITER>;</OAI_DELIMITER>
<OAI_MAX_RECORDS>5</OAI_MAX_RECORDS>
<OAI_MAX_IDENTIFIERS>10</OAI_MAX_IDENTIFIERS>
<OAI_DIR>..\oai</OAI_DIR>
<OAI_SETS_DIR>..\oai\oai-sets</OAI_SETS_DIR>
<DM>dm</DM>
<OAI_ID_DESCR>id-description.xml</OAI_ID_DESCR>
<OAI_ID_DESCR>eprins.xml</OAI_ID_DESCR>
<metadataFormats>
  <metadataFormat>
    <metadataPrefix>oai_dc</metadataPrefix>
    <schema>http://www.openarchives.org/OAI/2.0/oai_dc.xsd</schema>
```

```

<metadataNamespace>http://www.openarchives.org/OAI/2.0/oai_dc/
</metadataNamespace>
</metadataFormat>
<metadataFormat>
  <metadataPrefix>dc_culture</metadataPrefix>
  <schema>http://urltoschema</schema>
  <metadataNamespace>URI:DC.Culture/XMLSchema/1.0
  </metadataNamespace>
</metadataFormat>
</metadataFormats>
</OAIConfiguration>

```

Toelichting:

Alles wat hieronder tussen de XML-tags staat is een omschrijving van wat u moet invullen. Nieuwe terminologie wordt verderop in dit hoofdstuk uitgelegd.

<database>de Adlib database waarin gezocht kan worden**</database>**

<databasepath>schijfstation:\mijn Adlib-map\data**</databasepath>**

<DM>datum-tag**</DM>**

In veel databases wordt in een date-of-modification veld bijgehouden wanneer een record voor het laatst gewijzigd is. Als u uw database middels het Open Archive Initiative beschikbaar stelt op internet, dan kunt u met de variabele **DM** in het *adlibweb.xml*-bestand de tag van het genoemde veld opgeven zodat zoekmachines of andere clients alleen die records kunnen ophalen (of hoeven te indexeren) die ná een bepaalde datum zijn gewijzigd, bijvoorbeeld na het vorige bezoek aan uw OAI-server. Een OAI-request met een datumselectie voor een database waarin geen **DM** is gespecificeerd, levert álle records op.

<OAI_ADMIN_EMAIL>administrator e-mailadres**</OAI_ADMIN_EMAIL>**

<OAI_DELIMITER>scheidingsteken**</OAI_DELIMITER>**

Kies een scheidingsteken (bijvoorbeeld een puntkomma) om de URI-waarden achter de identifier-variabele in een OAI-request van elkaar te scheiden. Als u geen **<OAI_DELIMITER>** opgeeft, dan veronderstelt *wwwopac/oaiserver* dat u een dubbelepunt wilt gebruiken.

<OAI_DIR>pad naar directory waar de OAI resumption tokens en stylesheets moeten worden opgeslagen**</OAI_DIR>**

Wanneer u een relatief pad opgeeft, bedenk dan dat dit pad relatief moet zijn aan het pad dat u instelt met <databasepath>.

<OAIFIELDS>velden of tags**</OAIFIELDS>**

Normaliter worden in eerste instantie alle datadictionary velden in het XML-resultaat van een OAI-zoekvraag opgeleverd. Pas daarna wordt eventueel met behulp van een stylesheet een selectie hieruit gemaakt. Via het optionele **<OAIFIELDS>** kunt u echter zelf opgeven welke velden moeten worden opgehaald; dit kan resulteren in snelheidswinst. Geef de gewenste tags of veldnamen op, elk tussen **<field>** tags (zie ook paragraaf 1.2.3).

<OAI_ID_DESCR>XML-bestand met id-description**</OAI_ID_DESCR>**

De optionele **OAI_ID_DESCR**-variabelen in het *adlibweb.xml*-bestand verwijzen naar XML-bestanden waarvan de inhoud door de Identify-request wordt getoond als description-entities (extra metadata over de repository zelf, die vooral binnen een bepaalde community van belang kan zijn). De betreffende bestanden worden in het XML-zoekresultaat onderaan erbij geplakt tussen **<description>**-tags.

Aangezien deze bestanden niet op juistheid worden gecontroleerd tijdens het invoegen in de response, is het belangrijk om altijd te controleren of ze correct zijn voordat ze worden opgenomen. Een incorrect bestand zal een Identify-request doen falen.

De XML-bestanden moeten in de data-directory staan, in de directory waar *wwwopac/oaiserver* in staat, of in de **OAI_DIR**-map. Eerst zoekt het programma in de data-map, dan in de **OAI_DIR**-map, en als laatste in de *wwwopac/oaiserver*-map. Er kunnen nul of meer **OAI_ID_DESCR**'s zijn.

<OAI_MAX_IDENTIFIERS>maximum aantal
identifiers**</OAI_MAX_IDENTIFIERS>**

<OAI_MAX_RECORDS>maximum aantal op te halen
records**</OAI_MAX_RECORDS>**

<OAI_REPOSITORY_NAME>naam**</OAI_REPOSITORY_NAME>**

<OAI_REPOSITORY_IDENTIFIER>repository
id**</OAI_REPOSITORY_IDENTIFIER>**

<OAI_SETS_DIR>pad naar sets-directory**</OAI_SETS_DIR>**

De variabele `OAI_SETS_DIR` bevat de directory waarin Adlib-pointerfiles in submappen met de relevante databasenaam zijn opgeslagen, die door het OAI-protocol als sets worden gebruikt.

Als u een relatief pad opgeeft, dan moet dat pad relatief zijn aan het pad dat u specificeert met `<databasepath>`.

Als u geen sets-directory opgeeft, dan wordt de standaard pointerfiles-map gebruikt.

<XMLTYPE>STRUCTURED of UNSTRUCTURED</XMLTYPE>

Met het optionele `<XMLTYPE>` kunt u de XML-structuur van het zoekresultaat bepalen. Standaard is dat `STRUCTURED`, maar u kunt ook opgeven dat het resultaat `UNSTRUCTURED` moet zijn. Voor bijvoorbeeld Dublin Core zal het weinig of niets uitmaken, maar wel voor hiërarchische datastructuren. (Zie pagina 13 voor meer informatie over deze twee XML-structuren.) Voor elk `XMLTYPE` moet u een andere stylesheet gebruiken.

In het voorbeeld ziet u ook dat binnen de eerste `<metadataFormat>`-tags de standaard stylesheet wordt aangegeven, waarvan de `<metadataPrefix>` de naam van het bestand is. In de tweede sectie wordt een fictieve (zelfgemaakte) stylesheet `dc_culture` aangeduid. U kunt zoveel stylesheets vermelden als u wilt. In de OAI-zoekvraag wordt opgegeven welke van deze stylesheets moet worden gebruikt.

De volgende variabelen moeten voor gebruik van het OAI-protocol verplicht worden opgegeven in het *adlibweb.xml*-bestand:

- `OAI_REPOSITORY_NAME`
- `OAI_ADMIN_EMAIL`
- `OAI_REPOSITORY_IDENTIFIER`
- `OAI_DELIMITER`
- `OAI_DIR`

De rest is optioneel.

3.1.1 AdlibXML

De `wwwopac/oaiserver` levert standaard XML op in het formaat *adlibXML*. Als u een OAI-zoekresultaat in dit formaat wilt opleveren, en dus niet wilt laten transformeren naar "metadata" door een stylesheet, dan moet u in *adlibweb.xml* de `<metadataFormat>`-sectie(s) verwijderen en in plaats daarvan het te gebruiken *adlibXML*-formaat expliciet in uw OAI-configuratie bij de `<metadataFormats>`

opgeven. Neem de onderstaande tekst daartoe letterlijk over in de OAI-configuratie:

```
<metadataFormat>
  <!-- the built-in type 'adlibXML'
  -->
  <metadataPrefix>adlibXML</metadataPrefix>
  <schema>http://www.adlibsoft.com/adlibXML.xsd</schema>
  <metadataNamespace>http://www.adlibsoft.com/adlibXML
  </metadataNamespace>
</metadataFormat>
```

Met <OAIFIELDS> en <XMLTYPE> kunt u eventueel inhoud en structuur van dit XML-formaat bepalen.

U kunt zo'n niet-getransformeerd XML-resultaat bijvoorbeeld gebruiken voor synchronisatie met XML-databases of ontwikkelingsdoeleinden.

3.2 Gebruik

■ Aanroep:

Wanneer bij de aanroep van *wwwopac.exe* of *oaiserver.exe* het keyword `verb` wordt meegegeven, verwerkt de server de zoekvraag als een OAI-request.

In principe wordt het zoekresultaat van een OAI-request opgeleverd als XML-bestand, en moet dus eigenlijk door de client eerst naar een HTML-pagina worden omgezet om het resultaat als webpagina te kunnen weergeven. Niettemin kan in de huidige browsers een XML-bestand ook als code worden weergegeven, en dat voldoet voor testdoeleinden. De syntaxis van een zoekvraag en de manieren waarop u die kunt opgeven zijn gelijk aan die van normale zoekvragen voor de *wwwopac*, met dien verstande dat er na het vraagteken in de CGI-string een aantal bijzondere variabelen moet worden opgegeven in verband met het OAI-protocol.

Om bijvoorbeeld een zogenaamde Identify-request te maken, gebruikt u de volgende syntaxis:

```
...wwwopac.exe?verb=Identify
```

De waarde `Identify` levert een aantal gegevens op over de betreffende repository die met het OAI-protocol ter beschikking is gesteld, zoals de repository-naam (de naam van uw collectie bijvoorbeeld), de BASE-URL om zoekvragen aan te richten, de

protocolversie die wordt ondersteund, het e-mailadres van de repository-beheerder, en eventueel begeleidende informatie.

Merk op dat in dit en de volgende voorbeelden `wwwopac.exe` vervangen kan worden door `oaiserver.exe`, afhankelijk van welke server u gebruikt.

Een reguliere OAI-protocol request heeft ten minste één naam/waarde-paar dat de request door de client specificeert. Bovenstaand Identify-request is daar een voorbeeld van, maar in plaats van Identify kan elk van de standaard OAI-protocol requests worden opgegeven. Het aantal en de aard van extra naam/waarde paren hangt af van de argumenten voor de specifieke protocol request, bijvoorbeeld:

```
...wwwopac.exe?verb=GetRecord&identificier=oai:MijnMuseumNL:3&
metadataPrefix=oai_dc
```

voor een GetRecord-aanroep van het record met identificer `oai:MijnMuseumNL:3`. Maar, in een URI (het waarde-deel in een OAI naam/waarde paar) moeten speciale tekens worden gecodeerd als een escape-reeks, een procentteken met een hexadecimale waarde. De geldige request is dus:

```
...wwwopac.exe?verb=GetRecord&identificier=oai%3AMijnMuseumNL%3
A3&metadataPrefix=oai_dc
```

De identificer-waarde wordt standaard voorafgegaan door `oai`, het "schema" dat moet worden gebruikt. Vervolgens komt de repository-identificer (hoofdlettergevoelig), en ten slotte een lokale identificer die het recordnummer van een Adlib-record aanduidt (de `preref`). Deze drie delen worden gescheiden door de gecodeerde delimiter (hier `%3A` voor een dubbelepunt). De `metadataPrefix` `oai_dc` specificeert het gebruik van het Dublin Core-outputformat, zoals vastgelegd is in het *adlibweb.xml*-bestand.

Ook wanneer u een `ListIdentifiers` of `ListRecords` request opgeeft (zie verderop in dit hoofdstuk), dan specificeert u met de `metadataPrefix`-parameter (indirect) welke stylesheet gebruikt moet worden.

Op de cd-rom van Adlib 5.# of hoger vindt u de standaard stylesheet voor dit doel: *oai_dc.xsl*. Plaats deze in uw eigen *InternetServer\OAI-directory* (die u met `<OAI_DIR>` opgeeft). (Als deze submap nog niet bestaat, maak die dan aan.) Op basis van deze stylesheet kunt u eventueel eigen stylesheets met een andere naam maken. Bewaar die in dezelfde directory; hierin bewaart u ook de `resumptionTokens` (zie

verderop), en in een submap voor OAI-sets de pointerfiles die u als sets beschikbaar wilt stellen.

De gereserveerde tekens voor gebruik in URI's zijn:

Teken	URI-rol	Escape-reeks
/	pad-component-separator	%2F
?	query-component-separator	%3F
#	fragment-identifier	%23
=	naam/waarde-separator	%3D
&	argument-separator in query-component	%26
:	host port separator	%3A
;	authority namespace separator	%3B
	spatie	%20
%	escape-indicator	%25
+	escaped spatie	%2B

Als het gebruik van deze tekens níet correspondeert met hun voorbestemde URI-rol, dan moeten ze dus worden gerepresenteerd door hun escape-reeks.

3.3 Protocol requests

Een client kan verschillende requests maken aan een repository. Elke request is de waarde van een zogenaamde *verb*, namelijk:
`verb=<request>` en volgt direct na het vraagteken in een zoekvraag.

Hieronder volgen de belangrijkste requests. Gedetailleerde informatie over hun syntaxis en de responses die ze opleveren vindt u op <http://www.openarchives.org/OAI/openarchivesprotocol.html>.
 (Requests zijn hoofdlettergevoelig.)

GetRecord

Deze verb wordt gebruikt om een individueel metadata-record uit een repository record te extraheren. Verplichte argumenten

specificeren de identifier van het opgevraagde record, en het uitvoerformaat van de metadata (bijv. `oai_dc`).

Identify

Identify vergaart informatie over de repository. Geen argumenten.

ListIdentifiers

Deze verb wordt gebruikt om de lokale identifiers (prirefs) van alle records te vergaren die in de repository beschikbaar zijn. Optionele argumenten maken selectieve vergaring mogelijk, bijvoorbeeld op basis van hun datum van laatste wijziging, of als onderdeel van een set.

ListRecords

Deze verb wordt gebruikt om alle* records te vergaren die in de repository beschikbaar zijn. Optionele argumenten maken selectieve vergaring mogelijk, bijvoorbeeld op basis van hun datum van laatste wijziging, of als onderdeel van een set.
* De *adlibweb.xml*-instelling `OAI_MAX_RECORDS` bepaalt hoeveel records er maximaal, per request, worden opgehaald. Volgende deellijsten van hetzelfde zoekresultaat kunnen via zogenaamde resumptiontokens (zie paragraaf 3.5) worden opgehaald.

3.4 OAI Sets in Adlib

Een OAI-zoekvraag kan ook een set van records opvragen. Ofschoon de OAI-set meer mogelijkheden heeft, worden deze sets in Adlib aangewend door middel van pointerfiles. Eén pointerfile stelt één set voor. Om onderscheid te kunnen maken tussen pointerfiles voor gebruik in Adlib en die voor OAI-requests kan in het *adlibweb.xml*-bestand worden aangegeven waar de sets zich bevinden. De pointerfiles kunnen dus met bijvoorbeeld *adlwin.exe* gecreëerd worden en vervolgens desgewenst naar de juiste directory gekopieerd worden. In de directory waarin de OAI-sets staan moet eenzelfde onderverdeling zijn (per gebruikte database) als in de data-directory. Dus als de pointerfile uit de directory `..\data\collect` komt en de OAI sets directory `\oai-sets` heet, dan moet de pointerfile in `..\oai-sets\collect` komen te staan.

Als deze onderverdeling niet wordt gemaakt, dan kan `wwwopac/oaiserver` de pointerfile niet vinden (en zal de set dus niet worden opgeleverd).

Met de verb `ListSets` kan een lijst met alle pointerfiles worden opgevraagd, bijvoorbeeld:

<http://demo.adlibsoft.com/wwwopac/oaiserver.exe?verb=ListSets>

De setstructuur van de repository wordt geretourneerd. In het geval van Adlib pointerfiles betekent dit dat de eerste `setSpec`-node de naam van de database bevat, waarop de daaronder opgesomde `setSpecs` (de nummers van de relevante pointerfiles) van toepassing zijn. In ons voorbeeld levert dat dus pointerfiles nummer 4 en 5 op voor de collect-database.

Via andere verbs kunt u nu de inhoud van een pointerfile zelf ophalen, via `ListRecords` bijvoorbeeld. U moet dan het extra argument `set` in de request gebruiken. Naar de pointerfile verwijst u via het volgende format: `set=pointerfilenummer`, bijvoorbeeld:

http://demo.adlibsoft.com/wwwopac/oaiserver.exe?verb=ListRecords&set=5&metadataPrefix=oai_dc

3.5 ResumptionTokens

Als het zoekresultaat een erg lange lijst is, wordt een beperkt deel daarvan geproduceerd en vergezeld van een `resumptionToken` (onder aan het XML-resultaat). Zo'n `resumptionToken` is nodig om het volgende deel van de lijst op te vragen als argument in OAI-request. (Het argument moet dan wel eerst als een escape-reeks worden gecodeerd, mocht dat nodig zijn.)

Merk op dat als u in een request een `resumptionToken` gebruikt, u géén `metadataPrefix`, noch andere argumenten mag opgeven: de vorige gebruikte argumenten worden automatisch opnieuw toegepast.

Stel dat u een eerste `ListRecords`-request hebt opgegeven die een lijst met bijvoorbeeld tien records heeft opgeleverd, en een `resumptionToken` "97EODx6A3phYxIY":

```
../oaiserver.exe?verb=ListRecords&metadataPrefix=oai_dc
```

De volgende tien records van het zoekresultaat haalt u nu op met:

```
../oaiserver.exe?verb=ListRecords&resumptionToken=97EODx6A3phYxIY
```

In de nieuwe lijst vindt u weer een nieuwe `resumptionToken` als er nog meer records beschikbaar zijn.

De `resumptionTokens` die worden uitgegeven door een OAI-response van `wwwopac/oaiserver`, worden ook tezamen in één XML-bestand op de server opgeslagen: *OAIResumptionTokens.xml*. Hieronder ziet u een voorbeeld:

```
<?xml version="1.0" ?>
<resumptionTokens>
  <resumptionToken>
    <name>z6LLHpGifhyW63KiUIOI</name>
    <fromDate />
    <untilDate />
    <metaDataPrefix>oai_dc</metaDataPrefix>
    <verb>ListIdentifiers</verb>
    <limit>25</limit>
    <hits>2066</hits>
    <startFrom>26</startFrom>
    <expires>1238657612</expires>
  </resumptionToken>

  <resumptionToken>
    <name>4NJeSGjY9XWXbL5RwK13</name>
    <fromDate />
    <untilDate />
    <metaDataPrefix>oai_dc</metaDataPrefix>
    <verb>ListIdentifiers</verb>
    <limit>25</limit>
    <hits>2066</hits>
    <startFrom>26</startFrom>
    <expires>1238658375</expires>
  </resumptionToken>

  <resumptionToken>
    <name>49DVgIjnJeKbzYWeJwND</name>
    <fromDate />
    <untilDate />
    <metaDataPrefix>oai_dc</metaDataPrefix>
    <verb>ListIdentifiers</verb>
    <limit>25</limit>
    <hits>2066</hits>
    <startFrom>26</startFrom>
    <expires>1238658417</expires>
  </resumptionToken>
</resumptionTokens>
```

Met behulp van deze gegevens kan de vervolgzoekevraag correct uitgevoerd worden. Een token verloopt na 24 uur; deze periode is niet configureerbaar. Er is wel een work-around om incidenteel een langere periode te bewerkstelligen: datum en tijdstip van verloop worden per resumptionToken opgeslagen als het aantal seconden vanaf een bepaalde historische datum tot aan datum en tijdstip van verloop. Als u van tevoren weet dat u bepaalde resumptionTokens langer dan 24 uur wilt bewaren, pas de gewenste <expires>-waarden

in het XML-bestand dan handmatig aan door er voldoende seconden bij op te tellen (86400 per dag uitstel).

Wanneer een `resumptionToken` wordt uitgegeven, en het bestand bestaat nog niet, dan wordt dat aangemaakt. Als het wel al bestaat, wordt de `resumptionToken` gewoon toegevoegd. Als een `resumptionToken` vervolgens wordt gebruikt, dan wordt die weer uit het bestand weggehaald. Een leeg *OAIResumptionTokens.xml* - bestand wordt niet verwijderd en mag blijven staan.

3.6 Voorbeelden

Ga bijvoorbeeld naar <http://re.cs.uct.ac.za/> om OAI in de praktijk te testen. Geef een URL van een Adlib repository op waarop een Adlib `wwwopac/oaiserver` werkt die het OAI-protocol ondersteunt – en noem in de URL ook expliciet `oaiserver.exe` of `wwwopac.exe` – of kies een van de archieven uit de keuzelijst. Vul parameters in de daarvoor bestemde invoervakken in, en kies een verb om metadata op te halen. Zie ook <http://oai.dlib.vt.edu/cgi-bin/Explorer/oai2.0/addarchive> om een nieuwe repository automatisch te laten testen. Als er tijdens het testen een fout optreedt en het is niet duidelijk hoe en wat, dan kan de gebruiker zelf de URL naar `oaiserver.exe` of `wwwopac.exe` invoeren met daarachter `?verb=Identify`. De uitvoer geeft dan aan wat er mis is. Merk op dat volledige validatie alleen met `oaiserver.exe` (versie 6.4.0.579 of hoger) mogelijk is, niet met `wwwopac.exe`, omdat de foutafhandeling hierin iets anders is.

3.7 OAI repositories en zoekmachines

Of en hoe zoekmachines zoals Yahoo en MSN OAI repositories doorzoeken verschilt per zoekmachine, en de manier en mate van indexeren is meestal geheime informatie. Gegevens hierover moeten dus uit onderzoek komen. Uit een onafhankelijk onderzoek gedaan in 2000 (<http://library.lanl.gov/cgi-bin/getfile?LA-UR-05-9158.pdf>) naar de dekking van het OAI-corpus door enkele zoekmachines, bleek bijvoorbeeld dat Yahoo 65%, Google 44% en MSN 7% van dat corpus had geïndexeerd. 21% werd door geen van deze drie geïndexeerd. (Merk op dat er sinds 2000 veel kan zijn veranderd.)

Om de kans op indexering van uw OAI repository zo groot mogelijk te maken, moet u een paar dingen doen:

1. Als u een website hebt waarop gebruikers records in detail kunnen inzien die uit dezelfde database(s) komen die u via OAI openstelt, dan moet u zich eerst afvragen of u wilt dat internetzoekmachines die detailrecords moeten kunnen indexeren. Het voordeel is een

mogelijk volledige indexering van uw databaserecords; het nadeel is een hogere belasting van uw webserver. Het wel of niet openstellen van uw waarschijnlijk dynamische gegenereerde webpagina's voor "crawling" door "spiders", wordt vaak door een *robots.txt*-bestand op uw webserver ingesteld. En de meeste zoekmachines respecteren een zogenaamde robots exclusion, zodat uw website niet wordt doorzocht/geïndexeerd.

2. Voor een nuttige indexering van uw OAI repository is het niet per se nodig dat uw website ook geïndexeerd wordt. Wél relevant is dat alle URI's naar de volledige gegevensbronnen (resources), bijvoorbeeld URL's voor het opvragen van detailweergaven van uw databaserecords, worden geïndexeerd. U moet er dus voor zorgen dat de URI naar een resource deel uitmaakt van het betreffende OAI-record. U gebruikt hiervoor het Dublin Core *identifier*-element. In de Adlib voorbeeldstylesheets met de veldmapping naar Dublin Core wordt de *identifier*-tag nog gevuld met een recordnummer of objectnummer, maar u kunt hier eenvoudig de URL naar de pagina op uw website in plaatsen waar de detailweergave van het volledige record te zien is. Vervang daartoe in *oai_dc.xsl* de *priref*-template door iets als:

```
<xsl:template match="priref">
  <xsl:element name="dc:identifier">
    <xsl:text>http://our_website/dispatcher.aspx?action=search
&amp;database=ChoiceFullCatalogue&amp;search=priref=</xsl:text>
    <xsl:value-of select="."/ >
  </xsl:element>
</xsl:template>
```

Er wordt nu gewoon een URL voor het recordnummer geplakt. Een vergelijkbare URL kan in Adlib Internet Server 3 worden gebruikt om detailrecords op te halen. Merk op dat u ampersandtekens die in de URL voorkomen, in een stylesheet moeten escapen, d.w.z. vervangen door `&`, zoals in het voorbeeld is te zien.

U slaat zo twee vliegen in één klap: uw website zelf hoeft niet te worden geïndexeerd, want de OAI-records worden al geïndexeerd, en omdat alle URL's naar detailweergaven ook worden geïndexeerd hebben gebruikers van interzoekmachines zo toch toegang tot de beste weergave van uw databaserecords.

3. Meld uw OAI repository aan bij OAI-registers zoals dat van open-archives: <http://www.openarchives.org/data/registerasprovider.html> . De basis-URL die u hier moet opgeven is uw repository-URL, in principe dus tot en met *oaiserver.exe* of *wwwopac.exe* (alhoewel

bij het gebruik van een HTTP-handler de executablenaam mogelijk niet meer in de URL voorkomt).

- Op: <http://www.oaister.org/dataproviders.html> kunt u uw repository vervolgens bij OAIster aanmelden die direct met het oogsten van uw OAI-data aan de slag gaat. OAIster is een zoekmachine speciaal voor OAI repositories, en u kunt daarmee zelfs in afzonderlijke Dublin Core-elementen zoeken naar data: deze zoekmachine ondersteunt dus zelf OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting), dit in tegenstelling tot de "normale" zoekmachines die op indirecte manier oogsten. Yahoo doet dat bijvoorbeeld via OAIster. (Google is mei 2008 helaas gestopt met de ondersteuning van OAI-PMH-invoer als sitemap.)

3.8 Bepaalde records uitsluiten van OAI-zoekresultaten

Met de Adlib OAI-server stelt u een database open voor OAI-zoekvragen vanaf het internet. Maar het kan zijn dat niet alle records in die database openbaar toegankelijk mogen zijn. Het is gelukkig mogelijk specifieke records uit te sluiten van OAI-zoekresultaten. Adlib *oaiserver.exe* draait namelijk onder een bepaalde account in IIS (Internet Information Services) – de naam van deze account is standaard `IUSR_<servernaam>`, om de anonieme internetgebruiker aan te duiden – en deze accountnaam kunt u als gebruikersnaam invullen in een speciaal hiervoor toe te voegen veld in uw applicatie dat in de database moet worden ingesteld als *Authorisation user field* met *Exclude* als *Authorisation type*. Elk record waarin daarna de genoemde accountnaam wordt opslagen, zal nooit in het resultaat van enige OAI-zoekopdracht te vinden zijn.

Zie de paragraaf *Use the authorisation functionality* in het onderwerp *User authentication and access rights* onder het hoofdstuk *General topics* in de Designer Help voor uitleg over het instellen van dit type toegangsbeperking.

4 SRW/SRU

De opvolger van Z39.50 is SRW (Search/Retrieve Webservice). SRW is een XML-georiënteerd protocol om zoekopdrachten in een database met metadata en objecten, en andere informatieverzoeken op afstand over het internet te kunnen uitvoeren. Het voorziet in een gestandaardiseerde manier om deze functionaliteit uit te voeren, dit in tegenstelling tot de Adlib Internet Server webapplicatie die vergelijkbare functionaliteit al op een Adlib-eigen (dus niet gestandaardiseerde) manier implementeert (via invoer met de Adlib selectietaal en het resultaat als AdlibXML-uitvoer).

Met de SRW webtechnologie wordt het gemakkelijker voor aanbieders van informatie om hun databases via een standaard webservice toegankelijk te maken. Ter verduidelijking: door het gestandaardiseerde karakter van deze webservice kunnen derden standaard technologieën aanwenden om onafhankelijk, eigen webapplicaties te bouwen die de genoemde databases kunnen doorzoeken en de data in een eigen interface presenteren aan bezoekers van het internet. De database-eigenaren hoeven in principe alleen nog Adlib *wwwopac.exe* via het *adlibweb.xml*-bestand te configureren voor het SRW-protocol; een eigen Adlib Internet Server webapplicatie is voor dit doel niet nodig. Merk op dat SRW de hiervoor ingestelde databases als het ware openstelt (read-only natuurlijk) voor het gehele internet en dat iedereen de informatie daaruit naar believen kan gebruiken. Normaal gesproken zullen vooral openbare gegevensbanken hiervoor geschikt geacht worden. Als u data wilt verkopen of gegevens alleen in context via een eigen webapplicatie (misschien ook vanwege copyright op bepaalde data) wilt aanbieden, dan is SRW niet geschikt.

SRU (SearchRetrieve by URL) is een SRW-begeleidende service: één van ten minste twee manieren om het SRW-protocol te transporteren, in dit geval als naam/waarde-parameters (niet in XML-vorm) in een URL (in principe een http GET request) – de andere manier is via SOAP, maar dat is niet beschikbaar in *wwwopac*. (Merk op dat SRW in toenemende mate "SRU via SOAP" wordt genoemd.) SRW en SRU requests en resultaten zijn vergelijkbaar, het verschil zit hem in de manier van insluiten van zoekvragen en resultaten. SRU is bovendien complementair aan OAI, het Open Archives Initiative, dat ook door Adlib wordt ondersteund. Waar OAI vooral bedoeld is voor het harvesten (oogsten) van bulk(meta)data, is SRU geschikt om via specifieke zoekvragen alleen gezochte records op te halen. De overeenkomst tussen beide protocollen is dat het XML-formaat waarin het zoekresultaat wordt opgeleverd, tot op zekere hoogte door de gebruiker (de client) kan worden gekozen.

CQL (Common Query Language) is de querytaal die gebruikt wordt in SRW/SRU.

Versie 1.1 van het SRU zoekprotocol en de querytaal CQL worden door Adlib ondersteund. *Wwwopac.exe* versie 6.3 of hoger wordt aanbevolen.

4.1 Hoe Adlib SRU implementeert

Adlib *wwwopac* implementeert SRU/CQL versie 1.1 en conformeert aan het zogenaamde SRU Base Profile (zie <http://www.loc.gov/standards/sru/base-profile.html>) en het Adlib Base Profile (zie bijlage 2). Het Adlib Base Profile kan worden uitgeleverd aan derden.

Base profiles zijn tekstuele specificaties die nodig zijn om interoperabiliteit tussen SRU servers en clients te garanderen, en worden als een vereiste beschouwd wanneer een client of server het SRU-protocol zou moeten ondersteunen. De functionaliteit die in deze documenten wordt beschreven, kunt u zien als de minimale implementatiespecificaties; als zelfs die specificaties niet zijn geïmplementeerd, is er een grote kans dat client en server niet interoperabel zijn.

Het Adlib Base Profile is een uitbreiding van het SRU Base Profile.

SRW/U en CQL zijn ruime protocollen, met veel optionele en serverafhankelijk opties. De CQL context set bijvoorbeeld, beschrijft veel features die alleen maar mogelijkheden zijn. Sommige van deze mogelijkheden zijn in *wwwopac* anders geïmplementeerd. Ook zijn niet alle optionele features geïmplementeerd, bijvoorbeeld: SRW (SRU via SOAP) is niet geïmplementeerd, noch de *scan*-actie. Dit is allemaal gedocumenteerd in het Adlib Base Profile.

De Adlib SRU server is geïmplementeerd in de bestaande Adlib *wwwopac* executable, maar is bij uitlevering niet geconfigureerd voor daadwerkelijk gebruik omdat dat voor de meeste klanten ongewenst is. Net als voor OAI (het Open Archive Initiative), moet deze sub-server in het *adlibweb.xml* webconfiguratiebestand door uzelf worden geconfigureerd voordat hij kan worden gebruikt.

De configuratie bestaat uit drie delen:

- instellingen toevoegen aan *adlibweb.xml*;
- een XSLT-stylesheet creëren (of een bestaande aanpassen) voor de verlangde recordformaten in het zoekresultaat;
- een XSD-schema maken om de opgeleverde recordformaten te definiëren.

Ook moet waarschijnlijk een eigen profile-document worden gecreëerd. Als u wilt dat Adlib deze configuratie en documentatie voor u maakt, neem dan alstublieft contact op met onze verkoopafdeling of helpdesk, voor een offerte.

4.1.1 Uw eigen SRU Profile maken

Voordat SRU-clients uw database kunnen gebruiken, moeten ze weten welke indexen gedefinieerd zijn, welke recordformaten ondersteund worden en hoe zoekopdrachten zich gedragen. Dit wordt gedefinieerd in één of meer zogenoemde "context sets" en een "profile".

Een context set is een lijst met indexen, relations, relation modifiers en hun gedrag. Het Adlib Base Profile (zie bijlage 2) beschrijft onder andere al de relations, relation modifiers en hun gedrag, die niet gewijzigd kunnen worden door de configuratie. De context set die u zelf definieert zal dus alleen indexbeschrijvingen bevatten.

Een profile beschrijft dus de context sets die worden gebruikt, de recordformaten, en het specificeert URI's.

Het wordt aangemoedigd om al bestaande context sets te gebruiken, indien mogelijk. Het Bath-profile bijvoorbeeld, definieert zijn eigen context set maar leent ook van de Dublin Core, CQL en Record Meta-data context sets. Dit bevordert transparantie en standaardisatie. Hetzelfde geldt voor het gebruik van recordschema's. Het Bath-profile bijvoorbeeld, gebruikt de al bestaande Dublin Core en MarcXML-formaten.

Configureer eerst *wwwopac* (via *adlibweb.xml*) om de indexen en het recordformaat te ondersteunen die u via SRU wilt exporteren. Beschrijf deze indexen en recordformaten dan in uw eigen profile-document. Geef voor elke index expliciet aan of het een woord, term, datum (ISO of European/US), of numerieke index is. Dit is erg belangrijk omdat clients voor woord- en termindexen verschillende CQL-relations moeten gebruiken. Geef ook het *sortpath* aan dat clients moeten opgeven voor een sortering op de index.

Uw profile-document is een uitbreiding van het Adlib Base Profile, dus verwijst naar dat document in uw eigen profile.

Creëer een XSD XML-schema voor elk nieuw recordformaat dat u definieert, of hergebruik bestaande formaten (zoals Dublin Core of MarcXML). Zo weet de client hoe de opgeleverde records zijn opgebouwd, dit in verband met de verdere verwerking ervan (bijvoorbeeld een transformatie naar HTML via een eigen XSLT-stylesheet).

Ken URI's* en (wanneer van toepassing) short id's toe aan:

- elk recordformaat;
- elke context set (elke set van indexen dus);

- uw profile.

Voeg deze URI's en id's toe aan *adlibweb.xml* en uw profile-document. Voorbeelden van profile-documenten kunt u vinden op <http://zthes.z3950.org/srw/> en <http://zing.z3950.org/srw/bath/2.0/>.

* Een URI is een identifier die eruitziet als een URL, maar die niet naar een webadres hoeft te wijzen. De URI voor het SRU Base Profile bijvoorbeeld, is "*info:sw/profile/1/base-profile-v1.1*". Door bijvoorbeeld het webadres van uw organisatie, een naam en een versienummer toe te voegen, kunt u er zeker van zijn dat dit een wereldwijd unieke identifier is, nu en in de toekomst.

Nu kunt u de volgende informatie publiceren:

- de URL naar uw webserver (inclusief '*wwwopac.exe*');
- uw profile-document;
- alle XSD's die u hebt gedefinieerd.

Dit maakt uw SRU-serverbeschrijving compleet, zodat iedereen die de server wil gebruiken dat kan doen.

Het is mogelijk om meerdere profiles tegelijk te implementeren.

4.1.2 De Adlib SRU-server (wwwopac) configureren

Het *adlibweb.xml*-bestand moet worden uitgebreid met een afzonderlijke sectie, genaamd `<SRUConfiguration>`. Zie verderop in dit hoofdstuk voor een voorbeeld van zo'n sectie. De volgende subsecties worden in dit XML-element onderscheiden:

1. database-instellingen;
2. geïmplementeerde eigen profiles;
3. instellingen voor het ophalen van records;
4. tekstuele informatie;
5. recordschema's;
6. context sets.

1. Database-instellingen

Verwijs naar de Adlib-database die u via SRU wilt openstellen voor het internet, met:

- `<databasepath>`, `<database>`: het pad naar de data-map en de naam van de Adlib-database, bijvoorbeeld:

```
<databasepath>D:\demo.adlibsoft.com\IS3\data</databasepath>
<database>document</database>
```

Merk op dat het SRU-protocol het niet mogelijk maakt de te doorzoeken database in de CGI-string mee te geven: alleen de ene, in het *adlibweb.xml*-bestand opgegeven database kan doorzocht worden. Als u toch meer dan één database wilt openstellen, maak dan een kopie van de hoofdmap (en dus alle submappen ervan) van uw Adlib Internet Server-module – de data-map moet niet worden gekopieerd – voor elke database die u beschikbaar wilt stellen, en plaats die kopie in een andere map. Creëer dan een verschillende virtuele map voor elke kopie, en pas in het *adlibweb.xml*-bestand van elke kopie de naam van de te doorzoeken database aan (en controleer of het pad naar de database nog steeds klopt). Overigens moet elke kopie van dit webconfiguratiebestand waarschijnlijk op meer punten worden aangepast, omdat de andere database andere velden heeft. Voor bijvoorbeeld drie te doorzoeken databases hebt u dus twee kopieën en het origineel van de Adlib Internet Server-mappen nodig, en daarin verschillende *adlibweb.xml*-bestanden. De basis-URL van elke *wwwopac/database*, voor gebruik door de SRU-client, zal dus verschillend zijn.

- `<sruFields>` en `<XMLType>`: de velden (of tags) en het xmltype (structured of unstructured) die worden gebruikt om de voorlopige XML-records te genereren. Deze records kunnen vervolgens via XSLT stylesheets worden getransformeerd naar een van de recordformaten die uw profiel mogelijk maakt. In het *adlibweb.xml*-bestand gebruikt u deze instellingen bijvoorbeeld als volgt:

```
<XMLType>structured</XMLType>
<sruFields><allDataDictionaryFields/></sruFields>
```

2. Geïmplementeerde eigen profielen

Geef via `<sruProfile>` de URI's van eigen profielen op (die gebaseerd moeten zijn op het SRU Base Profile en het Adlib Base Profile). Alle URI's naar eigen profielen die hier worden opgegeven plus de in *wwwopac* hardgecodeerde URI's naar de twee default profielen, zullen aanwezig zijn in het SRU *explain*-record. Zo weten clients welke profielen u implementeert. In *adlibweb.xml* schrijft u bijvoorbeeld:

```
<sruProfile>extraProfile1-URI</sruProfile>
```

In een *explain*-record komt dat terug als:

```
<zrx:supports type="profile">info:srw/profile/1/base-profile-v1.1</zrx:supports>
```

```
<zrx:supports type="profile">info:srw/profile/6/1.0</zrx:supports>
<zrx:supports type="profile">extraProfile1-URI</zrx:supports>
```

Merk op dat alhoewel de default URI's nog "srw" in de identificatie hebben, deze verwijzen naar profiles die nu als SRU profile worden aangeduid.

3. Instellingen voor het ophalen van records

De <limit> en <maxlimit> definiëren respectievelijk het standaard maximum aantal records dat wordt opgeleverd wanneer door de client geen maximumRecords-argument wordt opgegeven, en het maximum aantal records dat kan worden opgeleverd ongeacht wat de client wenst. Beide zijn optioneel. Bijvoorbeeld:

```
<limit>10</limit>
<maxlimit>100</maxlimit>
```

4. Tekstuele informatie

Dit betreft informatie voor SRU-clients die aan de gebruiker moet worden getoond. De <sruTitle> is verplicht, de andere instellingen zijn optioneel. Als <sruLangUsage> wordt opgegeven, dan moet <sruLangUsageCodes> een spatie-gescheiden lijst met RFC-1766 landcodes bevatten (<http://www.fags.org/rfc/rfc1766.html>), bijvoorbeeld 'en nl'. De betekenis van al deze velden wordt uitgelegd in de ZeeRex-documentatie op <http://explain.z3950.org/>. Een voorbeeld van de betreffende *adlibweb.xml*-sectie:

```
<sruTitle>Adlib document database</sruTitle>
<!-- optional -->
<sruContact>E-mail: info@adlibsoft.com Address: Postbox 1436
Maarssen the Netherlands</sruContact>
<sruDescription>Our SRU database</sruDescription>
<sruExtent>Extent</sruExtent>
<sruDataAuthor>Adlib</sruDataAuthor>
<sruHistory>history</sruHistory>
<sruLangUsage>Database contains records in Dutch and
English</sruLangUsage>
<sruLangUsageCodes>nl en</sruLangUsageCodes>
<sruRestrictions>There are no restrictions on the use of this
data</sruRestrictions>
```

5. Recordschema's

Er moet ten minste één schema worden gedefinieerd, en één daarvan moet de default zijn. De volgende gegevens zijn vereist:

- **id:** dit is de short identifier om het schema in een SRU request mee aan te duiden, is hoofdletterongevoelig. Mag niet de volgende tekens bevatten . < > & " ' "

- `description`: een in spreektaal opgestelde beschrijving van de set.
- `schemaXSD`: volledig URL-pad naar de XSD van het recordschema.
- `schemaURI`: URI van het recordschema. Mag niet de volgende tekens bevatten . < > & " ' "

Optioneel is:

- `stylesheet`: het relatieve pad naar de locatie van een stylesheet die het AdlibXML-record transformeert naar het verlangde recordformaat. De locatie is relatief aan de locatie van *adlibweb.xml*. Als geen stylesheet wordt opgegeven zal het AdlibXML-record worden opgeleverd. Dit heeft dan wel nog steeds een XSD nodig die het specifieke formaat van deze records beschrijft!

Een voorbeeld van recordschema-instellingen in *adlibweb.xml*:

```
<sruSchemas>
  <sruSchema default="true">
    <id>dc</id>
    <description>Dublin Core</description>
    <schemaXSD>http://www.loc.gov/standards/sru/dc-schema.xsd
    </schemaXSD>
    <schemaURI>info:srw/schema/1/dc-v1.1</schemaURI>
    <stylesheet>..\sru\dc.xsl</stylesheet>
  </sruSchema>
</sruSchemas>
```

6. Context sets

U moet ten minste één context set definiëren, via `<sruSet>`, en één daarvan moet de default zijn. Elke context set moet ten minste één index bevatten. Eén of geen van de indexen kan de default zijn; als er geen default index wordt aangegeven, dan is *qcl.anywhere* de default.

Voor elke context set zijn de volgende gegevens vereist:

- `id`: de short identifier voor deze set (behorende bij een recordschema), is hoofdletterongevoelig. Mag niet de volgende tekens bevatten . < > & " ' " 'adlib' (zonder aanhalingstekens) is een speciale context set. In dit geval mogen geen `contextURI` of indexen worden opgegeven. Als deze speciale id wordt gespecificeerd, dan worden gewoon alle indexen en gekoppelde velden toegevoegd aan de ingebouwde context set 'adlib'.

- `contextURI`: URI van de context set. Mag niet de volgende tekens bevatten . < > & " '
- een lijst met indexen die u beschikbaar wilt stellen. Voor elke index zijn de volgende gegevens vereist:
 - `field`: Adlib-veldnaam of index-tag (hoofdlettergevoelig). Het wordt sterk aangeraden alleen geïndexeerde velden te gebruiken! Dit XML-element kan herhaald worden, indien nodig, om meerdere Adlib-indexen "tegelijk" te doorzoeken; de gesplitste "delen" (voor de afzonderlijke indexen) van de zoekopdracht – Adlib doorzoekt namelijk maar één index tegelijk – zullen worden gecombineerd via de booleaanse OR.
 - `name`: naam voor deze index voor gebruik in CQL (de zoekvraag in de SRU request), is hoofdletterongevoelig. Mag niet de volgende tekens bevatten . < > & " '
 - `description`: zelfverklarende veldnaam (voor het geval de `name` nog niet duidelijk genoeg is).

Optioneel zijn:

- `sortXPath`: hoofdlettergevoelig XML-pad naar de inhoud van het geïndexeerde veld in het opgeleverde XML zoekresultaat, dat kan worden gebruikt om op deze index te sorteren.
- `sortAllOccurrences`: stel dit in op `true` als op alle occurrences van dit veld moeten worden gesorteerd, in plaats van alleen op de eerste occurrence. `False` is de default. Als het `true` is dan worden records mogelijk meer dan eens opgeleverd als ze meer dan één occurrence van het huidige veld bevatten.
- `sortField`: als meer dan één veld in de huidige `sruIndex` is gedefinieerd, dan wordt standaard het eerste veld gebruikt voor sorteren – sorteren kan niet op meer dan één index worden uitgevoerd. Geef een andere tag of veld op in dit `sortField`-element, als u juist dat voor sorteren wilt gebruiken.

Een voorbeeld van een `sruSet`-specificatie ziet u hieronder:

```
<sruSets>
  <sruSet default="true">
    <id>dc</id>
    <contextURI>info:srw/cql-context-set/1/dc-v1.1</contextURI>
  <sruIndexes>
    <sruIndex default="true">
      <field>ti</field>
```



```

        <name>title</name>
        <description>Title</description>
        <sortXPath>/record/title</sortXPath>
        <sortAllOccurrences>>false</sortAllOccurrences>
    </sruIndex>
    <sruIndex>
        <field>au</field>
        <name>author</name>
        <description>Author</description>
        <sortXPath>/record/author</sortXPath>
        <sortAllOccurrences>>false</sortAllOccurrences>
    </sruIndex>
</sruIndexes>
</sruSet>
</sruSets>

```

Een complete SRU-configuratie

Een voorbeeld van een complete SRU-configuratie in het *adlibweb.xml*-bestand vindt u hieronder:

```

<SRUConfiguration>
<!-- Adlib db settings -->
<databasepath>D:\demo.adlibsoft.com\IS3\data</databasepath>
<database>document</database>
<XMLType>structured</XMLType>
<sruFields>
    <allDataDictionaryFields />
</sruFields>
<!-- optional additional SRU profiles. Defaults: SRU Base Profile
& Adlib Base Profile -->
<sruProfile>extraProfile1-URI</sruProfile>

<!-- SRU settings -->
<limit>10</limit>
<!-- defaultNumRecs -->
<maxlimit>100</maxlimit>
<!-- maxRecs -->

<!-- textual information -->
<sruTitle>Adlib document database</sruTitle>
<!-- optional -->
<sruContact>Email: info@adlibsoft.com Address: Postbus 1436
Maarssen</sruContact>
<sruDescription>My SRU database</sruDescription>
<sruExtent>Extent</sruExtent>
<sruDataAuthor>Adlib</sruDataAuthor>
<sruHistory>history</sruHistory>
<sruLangUsage>Database contains records in Dutch and English
</sruLangUsage>
<!-- if sruLangUsage is supplied: must contain space-separated
list of RFC-1766 country codes -->
<sruLangUsageCodes>nl en</sruLangUsageCodes>
<sruRestrictions>There are no restrictions on the use of this
data</sruRestrictions>

```

```

<!-- record schemas -->
<sruSchemas>
<!-- at least one schema required -->
  <sruSchema default="true">
    <!-- must be filled. id and URI must be CQL-able and XML-able:
      no . < > & " ' characters! -->
    <id>dc</id>
    <description>Dublin Core</description>
    <schemaXSD>http://www.loc.gov/standards/sru/dc-schema.xsd
      </schemaXSD>
    <!-- should be full URL path -->
    <schemaURI>info:srw/schema/1/dc-v1.1</schemaURI>
    <!-- optional -->
    <stylesheet>..\sru\dc.xsl</stylesheet>
    <!-- relative to localdir. if not available: adlibXML -->
  </sruSchema>
</sruSchemas>

<!-- context sets -->
<sruSets>
  <!-- at least one set required -->
  <sruSet default="true">
    <!-- only one set can be default -->
    <!-- must be filled. id, URI and sruIndex/name must be
      CQL-able and XML-able: no . < > & " ' chars! -->
    <id>dc</id>
    <contextURI>info:srw/cql-context-set/1/dc-v1.1</contextURI>
    <sruIndexes>
      <!-- at least one index per set required -->
      <sruIndex default="true">
        <!-- note: default-attr is only useful if set is default-->
        <!-- must be filled -->
        <field>ti</field>
        <!-- Adlib field or index tag. Case sensitive -->
        <name>title</name>
        <!-- CQL index. Case insensitive. Must contain no
          . < > & " ' characters! -->
        <description>Title</description>
        <!-- human readable field name -->
        <!-- optional -->
        <sortXPath>/record/title</sortXPath>
        <!-- case sensitive -->
        <sortAllOccurrences>false</sortAllOccurrences>
        <!-- false is the default -->
      </sruIndex>
    <sruIndex>
      <!-- default-attr is only useful if set is default -->
      <!-- must be filled -->
      <field>au</field>
      <!-- Adlib field or index tag. Case sensitive -->
      <name>author</name>
      <!-- CQL index. Case insensitive. Must contain no
        . < > & " ' characters! -->
      <description>Author</description>
    </sruIndex>
  </sruSet>
</sruSets>

```

```

    <!-- human readable fieldname -->
    <!-- optional -->
    <sortXPath>/record/author</sortXPath>
    <!-- case sensitive -->
    <sortAllOccurrences>false</sortAllOccurrences>
    <!-- false is the default -->
  </sruIndex>
</sruIndexes>
</sruSet>
</sruSets>
</SRUConfiguration>

```

4.2 Enkele kenmerken van SRW/SRU requests

Beide protocollen, SRW en SRU, definiëren de volgende drie acties:

- **searchRetrieve** – Gebruik een `searchRetrieve`-actie om een zoekvraag aan de database op te geven via CQL. Dit kunnen complexe zoekvragen zijn. Merk wel op dat de server niet elk aspect van CQL hoeft te implementeren. De client kan het resultaat (records) in een van de door de databasebeheerder gespecificeerde formaten laten opleveren. Dit kan bijvoorbeeld Dublin Core zijn, of MarcXML, en dergelijke.
U kunt dit uitproberen op onze online demo van `wwwopac`, bijvoorbeeld met een zoekvraag die getrunceerd naar "na" in titels zoekt:
`http://demo.adlibsoft.com/internetserver3/wwwopac/wwwopac.exe?version=1.1&operation=searchRetrieve&query=dc.title='na'*&recordSchema=dc&startRecord=2&maximumRecords=12`
- **scan** – Een scan vraagt een lijst met indextermen op die voldoet aan de request. (Deze actie is echter niet geïmplementeerd in `wwwopac`.)
- **explain** – Een explain-request vraagt informatie over de database op, zoals ten minste de locatie ervan, wat de database bevat, en welke features van het protocol worden ondersteund door de server. Het resultaat is in de vorm van een ZeeRex-record.
U kunt dit uitproberen op onze online demo van `wwwopac`:
`http://demo.adlibsoft.com/internetserver3/wwwopac/wwwopac.exe?version=1.1&operation=explain`

De `searchRetrieve` request heeft in ieder geval de volgende parameters:

- *query* – Deze parameter bevat een CQL-string als zoekvraag, en is noodzakelijk in een search request.

- *version* – Dit is de andere verplichte parameter in een request. Het specificeert de hoogste versie van het SRW-protocol dat door de client kan worden begrepen.

Overige (optionele) parameters bieden meer controle over het zoekresultaat, waaronder: *maximumRecords*, *startRecord*, *recordSchema*, en *sortKeys*. Voor een volledige lijst met parameters, en de specificatie van de syntaxis van een SRU request zie: <http://www.loc.gov/standards/sru/sru-spec.html#parameters>.

Zie <http://www.loc.gov/standards/sru/> voor meer informatie over SRW/SRU en CQL.

4.3 Uw SRU-server testen

Rob Sanderson (azaroth@liv.ac.uk) heeft een stel XSLT-stylesheets gemaakt die de XML-output van elke willekeurige SRU-server transformeren naar browsable HTML. Deze stylesheets zijn gepubliceerd onder de GPL (General Public License) en zijn enigszins aangepast door Adlib Information Systems om wat beter bruikbaar te zijn. Deze stylesheets zijn een gemakkelijke manier om uw SRU-configuratie te testen. Neem contact op met onze helpdesk als u deze stylesheets wilt ontvangen, inclusief informatie over het gebruik ervan. U zult een webbrowser nodig hebben die in staat is client-side XSL-transformatie uit te voeren. De meeste moderne browsers beschikken echter over die functionaliteit.

5 Vanaf het web data schrijven

U kunt gebruikers van uw website toestaan om naar een of meerdere van uw databases te schrijven. In het webconfiguratiebestand moet daarvoor de toekenning `<write_allowed>true</write_allowed>` worden geplaatst. Als u dat boven aan dat bestand doet, in de `<globalConfiguration>`, dan geldt de toekenning voor alle databases. Als u alleen specifieke databases schrijfbaar wilt maken, plaats de toekenning dan alleen in de betreffende secties, in een `<databaseConfiguration>` of in een `<groupConfiguration>`, en niet bovenaan. Om data te schrijven, moet u vanaf de website een POST-request sturen met `DATABASE=naam_van_doelatabase&DATA=Een_CGI_escaped_UTF-8_adlibXML_document&isutf8=1`.

Aangezien een GET-request (URL+data) maar een beperkte lengte kan hebben – neem aan 1024 tekens, alhoewel dat niet precies vastgelegd is in het HTTP-protocol – zou u hier alleen gebruik van kunnen maken als u zeker weet dat er erg weinig data wordt geschreven. Veiliger is het dus om altijd POST te gebruiken.

Merk op dat deze functionaliteit in de Adlib Internet Server 3 model-webapplicatie niet wordt toegepast.

5.1 CGI escapen

Een adlibXML-document dat u wilt wegschrijven, moet eerst in UTF-8 worden gecodeerd, waarbij letters met accenten in twee bytes worden weergegeven, en daarna voor CGI worden geëscaped. Dat laatste houdt in dat van elk teken de hexadecimale ASCII-waarde moet worden genomen met een %-teken ervoor, maar scriptingtalen hebben hier functies voor. Als er in de data zelf tekens voorkomen die verward kunnen worden met XML-tekens, zoals `<`, `>` en `&`, dan moeten die voorafgaande aan het CGI-escapen volgens de XML-regels worden geëscaped, namelijk tot respectievelijk: `<`, `>` en `&`.

Het volgende record bijvoorbeeld:

```
<record>
  <preref>10</preref>
  <auteur>tweeën</auteur>
</record>
```

zal verzonden moeten worden als:

```
data=%3crecord%3e%3cpriref%3e10%3c%2fpriref%3cauteur%3etwee%3c3%abn%3c%2fauteur%3e%3c%2frecord%3e
```

Merk op dat dit slechts een zeer kort voorbeeld is; in een echt adlibXML-document dient dit record te staan binnen:

```
<adlibXML><recordList>..</recordList></adlibXML>
```

5.2 Records schrijven, bijwerken of verwijderen

In het adlibXML-document dat u naar de database stuurt, kunnen één of meer records voorkomen. Elk record wordt apart en verschillend behandeld, afhankelijk van de waarde van de priref daarin:

- Als de priref groter dan 0 is, dan wordt het record aangemaakt als het recordnummer nog niet bestond en gevuld met de opgegeven data. Als het recordnummer wel al bestaat, dan wordt dat record alleen bijgewerkt met de opgegeven velden in de nieuwe data; alle niet-genoemde velden blijven in het bestaande record zoals ze zijn. Neem bijvoorbeeld het volgende bestaande record in de database:

```
<record>
  <priref>10</priref>
  <auteur>Bert</auteur>
  <auteur>Hedzer</auteur>
  <titel>Een boek</titel>
</record>
```

en het record dat u wilt schrijven is als volgt:

```
<record>
  <priref>10</priref>
  <auteur>Hedzer</auteur>
  <prijs>100</prijs>
</record>
```

dan resulteert in de database het volgende record:

```
<record>
  <priref>10</priref>
  <auteur>Hedzer</auteur>
  <titel>Een boek</titel>
  <prijs>100</prijs>
</record>
```

- Als de priref 0 is of ontbreekt, dan wordt een nieuw record toegevoegd; de priref van het resulterende databaserecord wordt dan automatisch bepaald.
- Als de priref kleiner dan 0 is, dan wordt de min er afgehaald en wordt het record met de resulterende priref verwijderd uit de database.

5.3 Na het schrijven van een record

Na het schrijven van een record stuurt wwwopac een adlibXML-document als antwoord terug. Per behandeld record wordt het record teruggegeven zoals dat in de database is komen te staan (of stond, na het verwijderen van een record), plus eventueel een foutcode (het attribuut *stat*), bijvoorbeeld:

```
<record stat="34">
  <priref>10</priref>
  <auteur>Hedzer</auteur>
  <titel>Een boek</titel>
  <prijs>100</titel>
</record>
```

In de volgende tabel vindt u veelvoorkomende fouten:

stat=	Oorzaak	Hoe te handelen
0	Geen fouten; alles is oké.	
1	Een tag is niet gevonden.	Controleer de toekenning DATABASE=...
11	Database is niet open.	Controleer de permissies en de toekenning DATABASE=...&DATABASEPATH=...
19	Dongle-fout.	Plaats <i>adlib.lic</i> in de juiste map.
31	Priref heeft een waarde die buiten de dataset valt.	Kies een andere priref of pas de grenzen van de dataset aan.
34	Het record is (al) gelockt.	Gebruik de recordlockmanager van Adlib Designer om het lock te verwijderen, als u zeker weet dat niemand het record nu bewerkt.
145	Useblock schrijffout.	Controleer of IIS permissie heeft om in de database te schrijven.
147	Een veldnaam is niet gevonden.	Controleer de toekenning DATABASE=...
157	Er werd een duplicate term opgegeven.	Controleer de thesaurustermen.

196	Time-out; het verwerken van de schrijfpdracht duurde te lang.	Gebruik een snellere computer, of maak een foutrapportage als het gaat om een foutieve syntaxis in het adlibXML-document.
301	Parsing-fout; in het adlibXML-document staat een syntaxisfout.	Maak een foutrapportage omdat het gaat om een foutieve syntaxis in het adlibXML-document.
303	MSXML-fout.	Controleer de installatie van MSXML4.0 en de XML-syntaxis.

Als een record een fout bevat, dan wordt het volgende record normaal behandeld. Alle record-acties die *stat="0"* opleveren zijn dus geslaagd. Naast deze statuscode per record kan er natuurlijk ook een globale fout optreden, die staat in de diagnostics-sectie van het adlibXML-document (XPath: `/adlibXML/diagnostics/error/code`).

6 Aanverwante zaken

6.1 Status

Gebruik de opdracht `Status` om allerlei informatie over de server te genereren; er wordt een XML-bestand opgeleverd.

In CGI is de aanroep:

```
http://mylocalhost.com/webdata/wwwopac.exe?Status=1
```

Dit levert bijvoorbeeld het volgende resultaat op:

```
<?xml version="1.0" encoding="UTF-8" ?>
<status>
  <dllCreateDateTime>May 31 2005 20:59:13</dllCreateDateTime>
  <build>Unicode</build>
  <moduleName>D:\demo.adlibsoft.com\InternetServer3\demo\wwwopac\
wwwopac.exe</moduleName>
  <moduleBuildVersion>Version 6.1.0.100</moduleBuildVersion>
  <buildVersion>Unified Version 6.1.0, build 100</buildVersion>
  <operatingSystem>Windows 2003 server 5.2 (Build 3790) Service
Pack 1</operatingSystem>
  <clientName>ADLIB Internal use only</clientName>
  <processID>2412</processID>
  <threadID>3444</threadID>
  <numberOfConnections>1</numberOfConnections>
  <numberOfActiveConnections>1</numberOfActiveConnections>
  <computerName>EUROPA</computerName>
  <userName>IUSR_EUROPA</userName>
  <protocolVersion>601</protocolVersion>
  <runningTime>0.0</runningTime>
  <totalRequestTime>0.0</totalRequestTime>
  <load>0.000</load>
  <availableDLLs>
  <dll name="adlibimg.dll">0x00e20000</dll>
  <dll name="adapl.dll">0x00000000</dll>
  <dll name="adaplU.dll">0x00000000</dll>
  <dll name="gdiplus.dll">0x4dd60000</dll>
  <dll name="unicows.dll">0x7f2d0000</dll>
  <dll name="rtf2xml.dll">0x01710000</dll>
  </availableDLLs>
  <localWebDirectory>D:\demo.adlibsoft.com\InternetServer3\demo\
wwwopac</localWebDirectory>
  <currentDirectory>D:\demo.adlibsoft.com\InternetServer3\demo
</currentDirectory>
  <loadedConfig>
  <localDirectory>D:\demo.adlibsoft.com\InternetServer3\demo\
wwwopac</localDirectory>
  <configFile timestamp="1150404402">D:\demo.adlibsoft.com\
InternetServer3\demo\wwwopac\adlibweb.xml</configFile>
  </loadedConfig>
  <connections>
```

```

<connection>
<handle>0x00d14008</handle>
<remoteAddress>10.1.1.155</remoteAddress>
<queryString>http://demo.adlibsoft.com:80/internetserver3/
wwwopac/wwwopac.exe?Status=1</queryString>
<numberOfRequests>0</numberOfRequests>
<numberOfErrors>0</numberOfErrors>
<running>yes</running>
<myself>yes</myself>
<global>yes</global>
<lockInfo>
<locks />
<lockStatistics>
<maxWait>0</maxWait>
<totalWaits>0</totalWaits>
<numberOfLockCalls>0</numberOfLockCalls>
<numberOfUnlockCalls>0</numberOfUnlockCalls>
<maxNumberOfTicksLocked>0</maxNumberOfTicksLocked>
<numberOfLengthErrors>0</numberOfLengthErrors>
<numberOfUnlockErrors>0</numberOfUnlockErrors>
</lockStatistics>
</lockInfo>
</connection>
</connections>
<memory>
<system>
<commitTotal>271056</commitTotal>
<commitLimit>3057116</commitLimit>
<commitPeak>307660</commitPeak>
<physicalTotal>1048008</physicalTotal>
<physicalAvailable>584360</physicalAvailable>
<cache>796300</cache>
<kernelTotal>125340</kernelTotal>
<kernelPaged>97192</kernelPaged>
<kernelNonPaged>28148</kernelNonPaged>
<handleCount>9796</handleCount>
<processCount>32</processCount>
<threadCount>469</threadCount>
</system>
<process>
<work>7675904</work>
<pageFaultCount>2111</pageFaultCount>
<peakWork>7712768</peakWork>
<quotaPeakPagedPoolUsage>37500</quotaPeakPagedPoolUsage>
<quotaPagedPoolUsage>37400</quotaPagedPoolUsage>
<quotaPeakNonPagedPoolUsage>7040</quotaPeakNonPagedPoolUsage>
<quotaNonPagedPoolUsage>5832</quotaNonPagedPoolUsage>
<pageFileUsage>4255744</pageFileUsage>
<peakPageFileUsage>4362240</peakPageFileUsage>
</process>
</memory>
</status>

```

6.2 Veldnamen

Gebruik voor de veldnamen in uw Adlib-databses geen gereserveerde woorden (afkomstig uit de selectietaal), om fouten te voorkomen. Dit zijn: `adapl, adlibce, all, and, as, ascending, contains, date, descending, diagnostic, end, equals, flat, from, generic, greater, key, limit, nand, narrower, not, numeric, occurrences, or, pointer, print, profile, raw, record, related, resolved, set, smaller, sort, startfrom, structured, style, text, to, transformed, tree, unstructured, when, xml`. Gebruik in veldnamen zeker geen spaties of het `&`-teken. En gebruik bij voorkeur ook de volgende tekens niet: `" ` @ = [] { } < > / ^`. Gebruik bij voorkeur alleen letters en cijfers, en eventueel de volgende drie tekens: `- _ .`

6.3 Meertalige velden

Velden kunnen op verschillende manieren meertalig zijn. Afhankelijk van de taal waarin een bezoeker/gebruiker de webapplicatie gebruikt, wordt data altijd automatisch in de juiste taal gezocht en gevonden records worden in dezelfde taal weergegeven; voor meertalige velden die in de betreffende taal geen waarde bevatten, maar wel in de zogenaamde invariante taal, *kán** de waarde in die taal worden weergegeven in het record. De taal stelt u in zo'n applicatie vaak in door op een vlaggetje te klikken.

In een `wwwopac`-zoekopdracht kunt u de parameter `LANGUAGE=<x>` op twee verschillende manieren gebruiken:

- U kunt `<x>` vervangen door een Adlib-taalnummer. U bepaalt zo alleen de presentatietaal van opgehaalde records: die is bijvoorbeeld nodig om de juiste vertaling van een enumeratief veld te kunnen tonen. Maar u stelt zo niet de zoektaal in: in elk meertalig veld zoekt u dan automatisch in alle talen.
- U kunt `<x>` vervangen door een standaard taalcode, bijvoorbeeld `[en-GB]` voor Engels of `[nl-NL]` voor Nederlands, om de taal op te geven waarin alle meertalig velden moeten worden doorzocht. De eerste twee letters van de code bepalen in Adlib de presentatietaal. Dit werkt alleen in een Adlib SQL of Oracle-database.

* Alleen als u `INVARIANT=true` meegeeft aan de zoekopdracht, wordt in het zoekresultaat een leeg veld (in de actieve zoektaal) vervangen door de inhoud van dat veld in de invariante taal.

Als u geen zoektaalparameter aan de zoekopdracht meegeeft, dan wordt in elk meertalig veld in alle talen gezocht, maar kan van ieder

meertalig veld in gevonden records alleen de waarde in de invariante taal worden weergegeven als u `INVARIANT=true` meegeeft; als u geen zoektaalparameter noch `INVARIANT=true` meegeeft, dan worden meertalige velden leeg opgehaald en ziet u van gevonden records alleen de inhoud van niet-meertalige velden. (De invariante taal van een meertalig veld in een record is niets meer dan de taal waarin de eerste waarde in dat veld in dat record werd ingevoerd.)

In een `wwwopac`-zoekopdracht op een Adlib SQL of Oracle-database kunt u ook per meertalig veld een taalattribuut meegeven in de vorm `tag[taalcode]=waarde&...`. Een taalattribuut heeft prioriteit over de `LANGUAGE`-parameter. Met een taalattribuut wordt het betreffende veld alleen in de aangegeven taal doorzocht. Voor ieder meertalig veld in een zoekopdracht kunt u een (andere) taalattribuut-waarde opgeven.

6.4 Afhankelijkheden

Bij een normale installatie van de Adlib Internet Server komen alle benodigde bestanden beschikbaar. Maar als bepaalde bestanden per ongeluk van uw systeem worden verwijderd, kan de Internet Server/`wwwopac` niet goed meer functioneren. Aan de hand van optredende fouten kunt u wellicht het ontbrekende bestand duiden:

- Voor de afhandeling van `imagemetadata` en `thumbnail` gebruikt Internet Server het Adlib-bestand `adlibimg.dll` en de Windows uitbreiding `gdipplus.dll`. Zonder (een van) deze bestanden werkt alleen de genoemde afhandeling niet.
- Voor de SDI-mailafhandeling gebruikt `wwwopac` de bekende SMTP-dll's, maar zonder deze dll's werkt het ook.
- Als MSXML4.0 (of hoger) niet geïnstalleerd is, werkt de Internet Server niet omdat het dan geen XML-operaties kan uitvoeren.

7 Bijlage 1: een eigen recordformaat

Zoekresultaten worden in een XML-bestand opgeleverd dat voldoet aan *adlibXML.xsd* (Extensible Schema Definition: een validatieschema voor eigen XML-code) voor Spectrum XML.

Vanaf *wwwopac* 6.1 is het mogelijk uw eigen AdlibXML recordformaat te definiëren. Dit kan wenselijk zijn als de standaard *STRUCTURED* en *UNSTRUCTURED* recordformaten voor uw toepassing niet voldoen.

U definieert zo'n XML-schema in het *adlibweb.xml* configuratiebestand voor uw *wwwopac*. (Deze functionaliteit wordt niet ondersteund voor *.www*-configuratiebestanden, de voorloper van *adlibweb.xml*.) Een handige manier om zo'n schema te definiëren is om het op basis van een bestaand XML-schema te doen.

Door een zoekvraag aan *wwwopac* direct als CGI-string in uw browser op te geven, krijgt u het zoekresultaat in XML te zien. U kunt daarbij uiteraard aangeven of u dat resultaat in *STRUCTURED* of *UNSTRUCTURED*-formaat wilt zien. De specificatie van het formaat zelf kunt u tegelijk ook opvragen, door in de CGI-string ook een van de opties *schema=template* of *schema=text* op te geven:

- **schema=template** toont boven de recordlijst in XML de specificatie van het gebruikte schema in de node `<recordTemplate>`. Metadata wordt zoals gewoonlijk in attributen meegegeven.
- **schema=text** toont boven de recordlijst als commentaartekst in XML de specificatie van het gebruikte schema: velden, koppelingen, typen etc.

Via het zo getoonde XML recordschema komt u dus ook te weten wat de namen van de huidige tag-nodes zijn, en welke tags allemaal opgeleverd kunnen worden. Selecteer en kopieer de complete `<recordTemplate>` en plak die in uw *adlibweb.xml*-bestand als child van een `<groupConfiguration>`-node of `<databaseConfiguration>`-node. (Vergeet daarbij niet om tekens zoals "&" te escapen. En als u kopieert en plakt vanuit de browser, haal dan ook de "-" tekens weg die daar staan om nodes in en uit te klappen.) U kunt dat schema nu naar wens aanpassen.

- De topnode is altijd `<recordTemplate>`.
- De subnodes bestaan uit een `<node>` met verschillende mogelijke attributen, namelijk:
 - `tag` – de op te vragen database-tag;

- `type` – het veldtype, namelijk een van de volgende:
 - `normalField` – dit is het default type;
 - `group` – voor te groeperen velden;
 - `richTextField` – converteert regelovergangen in dit soort velden naar `
` tags;
 - `RTF` – converteert de inhoud van RichText-velden naar XML via `rtf2xml.dll`;
 - `referenceLink` – voor gekoppelde velden;
 - `enumeratedField` – voor enumeratieve velden;
 - `reservedTag` – voor het meegeven van opties;
 - `termLink`;
 - `attribute` – voor mee te geven attributen;
 - `occurrence`;
 - `text` – om vaste tekst mee te geven, geen element;
 - `attributeText`;
 - `expandedField` – voor interne links, leest het gekoppelde record en geeft het recursief weer. Subnodes (zoals in het voorbeeld hieronder) worden genegeerd! Alle soorten interne koppelingen kunnen hiermee worden geëxpandeerd. Er wordt echter niet automatisch gecontroleerd op cirkelkoppelingen.
- `name` – het te gebruiken XML-element. Als de `name` ontbreekt zal de tag zelf worden gebruikt, waarbij een procentteken in de tag wordt geëscaped als `"perc"`. De `name` moet XML-compliant zijn, dus er mogen geen spaties in worden gebruikt;
- `linkdb` – alleen nodig voor `referenceLink` en `termLink`;
- `linktag` - alleen nodig voor `referenceLink` en `termLink`;
- `lreftag` - alleen nodig voor `referenceLink`;
- `text` – de tekst die moet worden gebruikt voor node-types `text` en `attributeText`.

- Nodes van het type `group`, `referenceLink` en `termLink` kunnen subnodes hebben.
- Wanneer een node attribute-subnodes heeft, dan moeten die als de eerste subnodes worden opgenomen. Attribute-subnodes na de eerste "normale" subnode zullen niet worden weergegeven!

Een fictief voorbeeld, voor testdoeleinden, ziet u hieronder:

```
<groupConfiguration group="template">
  <recordTemplate>
    <node tag="QQ">
      <node type="attributeText" name="staticAttr" text="xxx"/>
      <node name="occ" type="occurrence"/>
      <node type="text" text="drieën &lt; 10"/>
      <!-- this text is not shown, see last remark in the syntax
        description -->
      <node type="attributeText" name="notshown" text="yyy"/>
      <node tag="QQ" name="QQattr" type="attribute"/>
      <node tag="WW" name="WWattr" type="attribute"/>
      <node type="text" text="end"/>
    </node>
    <node type="expandedField" tag="bt" name="part_of_reference"
      termtag="IN">
      <node type="group" name="mygroup">
        <node type="richTextField" tag="WW" />
        <node tag="GG" />
      </node>
      <node tag="%0" name="prireref" />
      <node type="referenceLink" tag="AA" name="thename" lreftag="l1"
        linktag="BB" linkdb="d:\mydbA+yourdb">
        <node tag="AA" name="Oliver" />
        <node tag="D1" name="mergedfield" />
      </node>
      <node tag="AI" name="autoinc" />
      <node tag="DD" name="date" />
      <node type="element" name="extra">
        <node tag="DD" name="date" />
        <node type="text" name="fixed text">
      </node>
      <node type="enumeratedField" tag="EE" />
      <node tag="FF" />
      <node type="group" name="yourgroup">
        <node tag="H1" name="Heylang1" />
        <node tag="H2" name="Heylang2" />
        <node tag="H3" />
      </node>
      <node tag="ID" name="ISO_date" />
      <node tag="II" name="Integer" />
      <node tag="IS" />
      <node tag="LL" />
      <node type="termLink" tag="TI" linktag="ti"
        linkdb="d:\mydbA+yourdb">
        <node tag="ss" name="tt" />

```

```
</node>
<node tag="TT" />
<node tag="ll" name="linkref_info" />
</recordTemplate>
</groupConfiguration>

<databaseConfiguration database="mydb" groups="template">
  <DATABASEpath>c:\something</DATABASEpath>
  <DATABASE>mydb</DATABASE>
</databaseConfiguration>
```

Sla het gewijzigde *adlibweb.xml*-bestand op in Unicode UTF-8 codering.

In principe kan een eigen schema ook via de CGI-string worden meegegeven, indien u geen webconfiguratiebestand wenst te gebruiken. Dat doet u na een `xmltemplate=...`, waarin de gehele `<recordTemplate>`-node URL-encoded moet worden toegevoegd. Controleer wel of de gehele node naar de URL geplakt is, want de lengte van een CGI-string is beperkt.

Merk op dat als u een eigen recordschema hebt gedefinieerd, daar altijd gebruikt van zal worden gemaakt. U hebt dan dus niet meer de keuze soms ook de standaard `STRUCTURED` en `UNSTRUCTURED`-formaten te gebruiken. Ook kunt u (nog) geen onderscheid maken tussen op te halen velden voor een lijstweergave en een detailweergave. Dit zal in de toekomst misschien wel geïmplementeerd worden.

8 Bijlage 2: het Adlib Base Profile

Adlib wwwopac implementeert SRU/CQL versie 1.1 en conformeert aan het zogenaamde SRU Base Profile (zie <http://www.loc.gov/standards/sru/base-profile.html>) en het volgende Adlib Base Profile:

Adlib Base Profile (version 1.0)

Beta-4

Initiated: Monday December 13th, 2004

Last updated: June 14th, 2006

Hedzer Westra, Erik Lange

Adlib Information Systems

<http://www.adlibsoft.com>

Contents

- Introduction
- URIs
- SRW, SRU and CQL
- SRW Requirements
- Optional and server-dependent features
- Configuration
- Implemented SRW/CQL features
- Unimplemented optional features
- Adlib server-specific implementations
 - Indexes

- Relations
- Terms
- Modifiers
- Sorting
- The Adlib Context set
- CQL to Adlib bridging

Introduction

This document describes the Adlib Base Profile 1.0 of the SRU server that is implemented by Adlib Information Systems in its wwwopac software. It is only a partial profile: it only describes the software capabilities. For each installation and configuration of this software a full profile document should be defined (by the client).

URIs

The URI for the Adlib Base Profile is "info:srw/profile/6/1.0".

The Adlib context set is identified by the URI "info:srw/cql-context-set/6/1.0". The preferred identifier for this URI is 'adlib'.

These URIs are defined using the SRW 'info URI' convention, using the Adlib authority string "6", which was assigned by the ZING working group.

SRW, SRU and CQL

The SRW and SRU protocols, and the CQL language, CQL context set and SRU Base Profile are all described on <http://www.loc.gov/standards/sru/> . Please refer there for information about this search & retrieval protocol.

SRW Requirements

The Adlib wwwopac.exe implements all features required by the SRU Base Profile, refer to <http://www.loc.gov/standards/sru/base-profile.html>.

Optional and server-dependent features

SRW/U and CQL are quite broad protocols which allow for many optional and server-dependent features.

This document defines which optional and server-dependent features are implemented, and how these function.

Configuration

Adlib supplies a number of applications to its customers, who can make changes and additions to their applications. Therefore there is no single 'Adlib' context set or profile. Each customer can configure their wwwopac software (through settings in the *adlibweb.xml* file) to produce the SRU context sets and profiles that are required. For any such configuration a custom context set and profile document should be created - or existing ones should be adapted. This document describes the Adlib Base Profile and the Adlib Context Set, which must form the basis of such a customer-defined profile. Therefore, only meta-indexes are defined here. Metadata formats (like Dublin Core or MarcXML) are not defined here.

Implemented SRW/CQL features

- protocol version 1.1
- the SRU protocol, i.e., HTTP GET/POST CGI requests. This is a slight extension of SRU, since the current documentation does not mention SRU POST.
- the explain operation
- the searchRetrieve operation
- CQL 1.1 parsing
- CQL 1.1 handling
- CQL context set as far as the Base Profile requests it, plus some extras:
 - and, or, not Booleans
 - =, >, <, <=, >=, <> relations
 - exact, all, any, scr relations
 - the within relation
 - cql.anywhere, adlib.allIndexes and adlib.record meta-indexes
 - cql.serverChoice surrogate index
- sorting
- surrogate & non-surrogate diagnostics generation
- recordSchemas
- request echoing, xSortKeys and XCQL

Unimplemented optional features

- recordXPath handling
- result sets
- full proximity searches, i.e. the 'prox' boolean
- word anchoring (^)
- matching on a single character using '?'

- scan operation
- SRW (SOAP) as communication layer

Adlib server-specific implementations

Indexes

- The `adlib.allIndexes` meta-index searches all indexes defined in the Adlib database at once. This is different from `cql.anywhere`, which searches all indexes in all context sets.

- The `adlib.record` meta-index searches the whole record, so all of the fields in each record. This includes data that is not indexed (and possibly not even displayed in any record schema) and therefore not searchable using CQL indexes. The relation must be '='.

Note: In future versions, CQL might support a `cql.record` meta-index with the same semantics.

Relations

- `cql.scr` is always handled as '='.

- The 'within' relation is implemented using range searching. Exactly two words must be supplied, separated by a single space. The range search type can be selected by using an `adlib.range` modifier, with the following values:

leftexclusive

rightexclusive

exclusive

inclusive (default)

Example: `date within/adlib.range=leftexclusive "2000 2004"`

Note: the CQL context set always uses inclusive range searching; there is no range modifier.

Terms

- Empty term searches are not supported.
- * for pattern matching is only usable at the beginning and/or end of a search term.
- A maximum of 250 characters per term is supported.

Booleans

- A maximum depth of 250 nested expressions is supported.

Modifiers

- The 'and' boolean accepts one of the modifiers 'adlib.when' and 'adlib.whennot'. See below for more information.
- Thesaurus-enabled searches can be executed by issuing an adlib.thesaurus modifier with one of the following values:

generic

broader

narrower

related

topterm

parents

These only work correctly on indexes with thesaurus links defined. Otherwise, they fall back on normal searching. The modifier is supported only for the 'exact' and '=' relations.

- There are two types of CQL context set modifiers: data type modifiers and pattern modifiers. The accepted data type modifiers are:

cql.string

cql.word

cql.isoDate

cql.number

Note: cql.uri is invalid in the Adlib Base Profile.

The accepted pattern modifiers are:

cql.masked

cql.unmasked (not yet defined in CQL context set)

Of each modifier type only one can be active for each search clause. The combinations /cql.masked/cql.number and /cql.masked/cql.isoDate are invalid.

- Adlib interprets terms in the following manner:

If modifiers are sent in the query, those are used. If they are not, modifiers are implied according to the following rules:

1. pattern modifier: cql.masked is always assumed, unless cql.number or cql.isoDate are supplied or implied.
2. data type modifier, implied using the relation and sometimes the Adlib index type:

- 'exact': implied modifier is cql.string.
- '=': implied modifier is cql.word, cql.number or cql.isoDate, depending on the index type.
- 'any' and 'all': implied modifier is cql.word. Words are combined using OR (for 'any') or AND (for 'all').
- 'within': there is no implied data type modifier; there must be two words separated by a single space

Modifier cql.unmasked means: the CQL pattern match character * has no special meaning; pattern matching is not possible using this modifier. The characters ^, ? and * do not have to be escaped.

Modifier cql.masked means: pattern matching on * is possible. The characters ^, ?, * and \ must be escaped with \.

Modifier cql.word means: words are split and then re-combined using the Adlib separators and concatenators rule. Word adjacency is not used when searching. An error will be returned when searching with cql.word on a string index.

Separator characters are: [] ; , ! @ () { } < > ? carriage return newline space tab

Concatenator characters are: ` - = \ / ~ # \$ % ^ & _ + : " ' *

Note: the CQL context set says nothing about how words are to be split but instead leaves that up to implementations to be specified.

Modifier `cql.string` means: terms are not inspected for separators or concatenators. An error will be returned when searching with `cql.string` on a word index.

Modifier `cql.isoDate` means: term is interpreted as a single ISO 8601 date. If the Adlib index type is European or US, the term will be translated before the actual search takes place.

Modifier `cql.number` means: term is interpreted as a single 32-bit signed integer.

The difference(s) between the CQL context set is/are the following:

- word adjacency, which should be used with the `cql.word` data type modifier, is a feature which the Adlib database engine currently does not implement.

Sorting

- sorting is only supported for hard-wired (case-sensitive) paths, not for full XPath's. The customer can define a path for each CQL index.

The Adlib Context set (version 1.0)

The Adlib Context Set version 1.0 defines:

(meta-)indexes:

- `adlib.record` (whether this will be added to the CQL context set is still unclear, so the prefix is `adlib`, not `cql`)
- `adlib.allIndexes`

modifiers:

- `adlib.thesaurus` for the '=' and `exact/cql.word` relation with its six accepted values (generic, broader, etc.)
- `adlib.range` for the 'within' relation
- `adlib.when` and `adlib.whennot` for the 'and' boolean
- `cql.unmasked` (it is assumed this modifier will eventually be added to the CQL context set, hence the `cql` prefix)

CQL to Adlib bridging considerations

The 'and' boolean modifiers:

The and-relation modifiers are implemented using the WHEN and WHEN NOT booleans in Adlib. In Adlib, the WHEN (NOT) booleans on two distinct indexes will first search records using the left operand index. For each record the right operand operator and value will be checked in the same occurrence. The Adlib-specific property of occurrences comes down to the following: each field can have 0 or more values (unless explicitly stated as non-repeatable), whereas conventional RDBMS's can only hold 0 or 1 values in a field. Indexes can be specified to index either the first or all occurrences. The WHEN operator explicitly checks matching occurrences. See the Adlib User Guide (available from <http://www.adlibsoft.com/>) for more information.

'within':

This relation is implemented using range searching, i.e. the Adlib WHEN boolean used on two identical indexes. Note that this is a very different operation from the WHEN boolean used on two distinct indexes.

Performance:

- The `adlib.allIndexes` and `cql.anywhere` meta-indexes might have slow search responses if there are a lot of indexes.
- The `adlib.record` meta-index cannot use any index and will always be slow.
- Searching on `*...*` is done using the Adlib 'contains' operator, which is slow since no index can be used.

9 Index

-	24, 29, 34	,	33
#	14	?	25
%		`	34
%27,	33	“	34
%0	15, 33, 35	+	24, 29, 33
%0D%0A	33	<	27
%20	33	<=	27
%22	34	<allDataDictionaryFields/> ..	22
%25	33	 	22
%250	33	<database>	49
%26	33	<databasepath>	49
%27	34	<DM>	49
%2B	33	<FIELD>	22
%2C	33	<HIGHLIGHT>	10
%2D	34	<OAI_ADMIN_EMAIL>	49
%d	12	<OAI_DELIMITER>	49
%HH	33	<OAI_DIR>	50
%m	12	<OAI_ID_DESCR>	50
%y	12	<OAI_MAX_IDENTIFIERS> ..	50
%Y	12	<OAI_MAX_RECORDS>	50
&		<OAI_REPOSITORY_IDENTIFIE R>	50
&	25, 33	<OAI_REPOSITORY_NAME> ..	50
&B[1]	34	<OAI_SETS_DIR>	50
&B[2]	34	<OAIFIELDS>	22, 50
&B[3]	34	<recordTemplate>	81
&B[4]	34		
&B[5]	34		
>;	27		
&I	34		
<;	27		
*			
*	14, 32		

<XMLTYPE> 51
 =
 = 24
 >
 > 27
 >= 27

A

aanhalingstekens 29, 34
 absolute paden 4
 absoluut pad 4
 adapl 12
 Adlib Base Profile 63
 Adlib foutmeldingen 22
 Adlib Internet Server
 opdrachten 7
 ADLIB_DIRi 14
 ADLIB_OUTPUT 15
 ADLIB_PTR 7, 37
 adlibweb.xml 3
 adlibXML 51
 AdlibXML
 aangepast 81
 adlibXML.xsd 81
 adlibXML-schema 81
 Adres 35
 afbeeldingen 38, 41
 watermerken 43
 all 32
 All 27
 allow 19
 ALLOW_ALL_DBs 7
 ampersand 25, 33
 ampersandtekens 59
 AND 24, 25
 AND NOT 24
 ascending 31
 ASPX 2
 auteursrechten 43

B

BOOL 25
 booleaanse functies 25
 booleaanse operatoren 35
 BRIEFADAPL 15
 BRIEFFIELDS 8, 21
 BRIEFSTYLE 18
 browser 23
 BT 35

C

callerIP 12
 catalogus 1
 CGI 1
 CGI escapen 73
 CGI-string 23, 33
 client-side 23
 commentaar 14
 Common Gateway Interface .. 1
 contains 27
 CONTENT-TYPE 15
 copyrightvermelding 43
 CQL 62
 CR LF 33
 crawling 59
 CROPABSOLUTE 42
 CROPHEIGHT 42
 CROPWIDTH 42
 CROPX 41
 CROPY 41
 CSS 2

D

DATABASE 3, 9, 25
 databasenaam 3
 databasepad 3
 DATABASEPATH 4, 9
 date-of-modification 48, 49
 datumvelden 27
 DEBUG 25
 debug=1 19
 default.www 4
 DEFAULTTHUMBNAIL 43

DELETE_ALLOWED 21
 descending 31
 DETAILADAPL 16
 DETAILFIELDS 9, 21
 DETAILSTYLE 16, 18
 DIAGNOSTIC 13
 disallow 19
 dispatcher.aspx 24
 DOM 26
 domein 26
 DONTKEEPASPECTRATIO 39
 dubbelepunt 26
 Dublin Core 59
 Dublin Core metadata-
 standaard 47

E

escaped 33
 Expires 17
 explain
 in SRW/SRU 71

F

FACS-naam 9
 First 27
 FLD 24, 26
 FLD/VAL-paren 36
 forward slashes 7
 foutcode 75
 foutcode 135 37
 foutmeldingen 22
 foutmeldingsbestanden . 14, 16
 FULLIMAGE 41

G

galerij 37
 gebruikersinterface 1
 gerelateerde term 35
 gereserveerde tekens ... 33, 36
 gereserveerde woorden 24, 33,
 79
 gesorteerd 27
 GetRecord 54

GetRecord-aanroep 53
 Google 58

H

haakjes 36
 harde return 33
 hexadecimale cijfers 33
 hexadecimale waarden ..27, 36
 HIGHLIGHT 9
 hoofdlettergevoelig 33
 hoofdletters 33
 HTML-formulieren 23

I

identificer 59
 Identify 55
 Identify-request 50, 52
 IIS 5 1, 6
 imagemetadata 44, 80
 IMAGEOVERLAYBLEND 43
 IMAGEOVERLAYFILE 43
 IMAGEOVERLAYMINPIXELSIZE
 44
 IMAGEOVERLAYPERCENTAGE
 44
 IMAGEOVERLAYPOSITION ... 43
 index 26, 27, 28
 indexnummer 24
 integer velden 27
 internet 1
 Internet Server-opdrachten ... 7
 intranet 1
 INVALIDFIELD 16
 IP-adres 11
 ISAPI 23
 isutf8=1 37

K

komma 24
 komma's 33

L

LANGUAGE 16, 79

language=nld..... 17
 leesrechten.....7
 leestekens 33, 36
 LIMIT 27, 32
 LISTADAPL 17
 ListIdentifiers..... 53, 55
 ListRecords..... 53, 55, 56
 ListSets 55
 LOGFILE 11

M

mediatype 23
 meertalige velden..... 79
 metadata..... 47
 afbeeldingen 44
 metadata harvesting protocol
 47
 metadataFormats 51
 metadataPrefix..... 53
 MIME-type 23
 miniatures
 maximumgrootte..... 40
 miniatuurafbeelding
 standaard 42
 miniatuurafbeeldingen..... 38
 minteken 34
 MODE 27
 MODEi..... 27, 29
 MSXML4.0 80

N

n 22, 33
 NAND..... 25
 nauwere ter 35
 nesten 36
 NetScape..... 17
 niet-alfanumerieke tekens ... 33
 niet-voorkeursterm..... 35
 NOEXPIRES 17
 norecs..... 14
 NORECS 18
 norecs.html 18
 not 25
 NT 35

O

OAI 48, 61
 OAI protocol request..... 47
 OAI repositories 58
 OAI Sets 55
 oai_dc..... 53
 oai_dc.xsl..... 48, 53, 59
 OAIFIELDS 21
 OAI-PMH 60
 OAI-protocol..... 47
 OAI-record 48
 OAI-registers 59
 OAI-request..... 47, 52
 OAIResumptionTokens.xml .. 56
 OAIster 60
 OAI-zoekvraag 47
 OCC..... 27
 occurrences 27
 opdrachten 24
 Open Archives Initiative 47
 operator 27
 operator SEARCHBOOL 36
 OPR..... 27
 OR..... 24, 25
 OUTPUTTYPE 39
 overlay 43

P

plusteken 33
 pointer 37
 pointerfiles 7, 36, 55
 Pointerfiles 37
 priref 33
 prirefs..... 74

R

rangschikken 31
 RAW 13
 relatieve paden 4
 repository..... 47
 request 54
 resumptionTokens 56
 richText 22

robots.txt 59
 RT 35
 ruimere term 35
 RUNADAPL 12

S

scan 34
 in SRW/SRU 71
 SCAN 8, 14, 27
 SCANDOM 27
 SCANEND 27
 SCANUNIQUE 28
 SCANVAL 28, 32
 SCANXMLFORCE 28
 scheidingsteken 49
 SDI 80
 SEARCH 27, 29, 36, 37
 SEARCHBOOL operator 36
 SEARCHMODE 27, 29
 searchRetrieve
 in SRW/SRU 71
 selectietaalsyntaxis 24
 SEQ 31
 server-side 23
 set 56
 setSpec 56
 sorteren 27, 33
 Sorteren 31
 spaties 14, 33
 spiders 59
 SRT 31
 SRU 61
 SRW 61
 SRW Profile 63
 standaard miniatuurafbeelding
 42
 STARTFROM 32
 stat 75
 Status 77
 STRUCTURED 13, 81
 STYLE 18, 19
 stylesheet 53
 SYNC_ALLOWED 21
 SYNTAXERROR 19
 syntaxes 35

syntaxis 24, 29
 syntaxis OAI-requests 54

T

tags 24, 26, 36
 TE 35
 tekens 24
 term 35
 thumbnail 80
 THUMBNAILBACKCOLOR 40
 thumbnails 38
 achtergrondkleur 40
 maximumgrootte 40
 TIMEOUT 12
 toegangsrechten 6
 TRANSFORMED 19
 transformed=1 19
 TRC 32
 truncatie 32
 TYP 33
 type="richText" 22

U

UF 35
 uitvoerrechten 7
 UNSTRUCTURED 13, 81
 urlencoded 23
 URL-invoervak 23
 US 35
 UTF-8 37
 UTF-8 header 18

V

VAL 24, 29, 32, 33
 validatieschema 81
 variabelen 24
 variabele-paren 35
 veiligheidsoverwegingen 4
 veldcodes 26
 veldnaam 26
 veldnamen 36, 79
 verb 52, 54
 virtuele directory's 6

volgorde..... 24, 36
 voorkeursterm35
 vraagteken25

W

watermerken 43
 weband 29
 webconfiguratiebestand.....3
 webor 29
 webserver1
 website1
 Windows 20001
 WinLatin1 37
 WQY 29
 write_allowed..... 73
 WRITE_ALLOWED 21
 www-bestand..... 3, 25
 wwwopac-opdrachten.....4

X

XML 2, 61
 XMLADAPL..... 13
 xmltemplate 84
 XMLTYPE 13
 XML-uitvoer..... 34
 XSIZE 39, 42
 XSLT..... 2, 18

Y

Yahoo 58
 YSIZE 39, 42

Z

Z39.50..... 61
 zoekformulier.....1
 zoekmachines 29, 58
 zoektermen 23
 zoekvragen..... 36