

# \*:96 Overheads

## Part 3d: Mailing lists

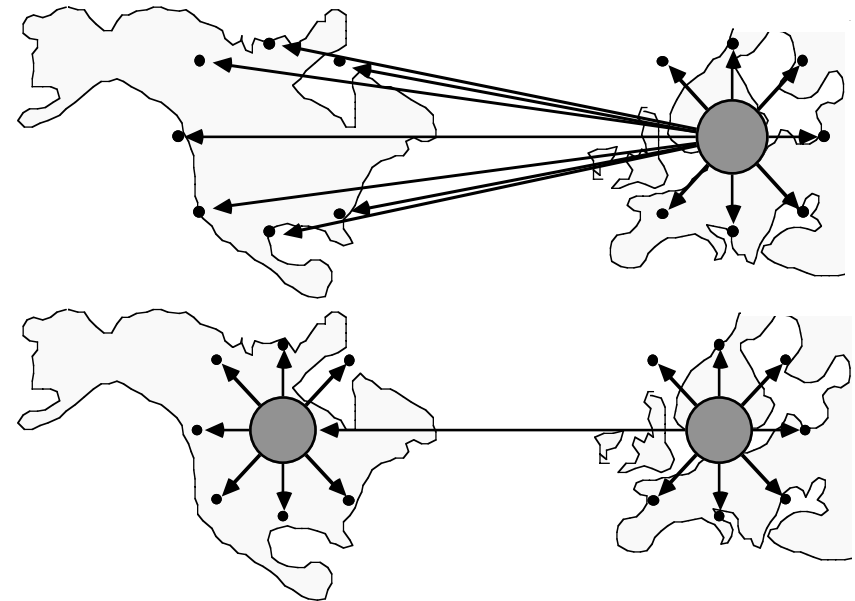
More about this course about Internet application protocols can be found at URL:

<http://www.dsv.su.se/jpalme/internet-course/Int-app-prot-kurs.html>

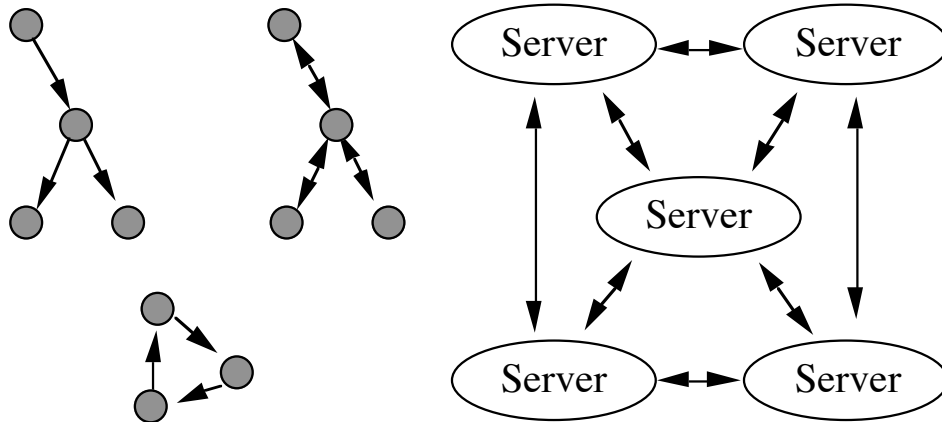
Last update: 30 Jan 2006

Compendium 8 page 184

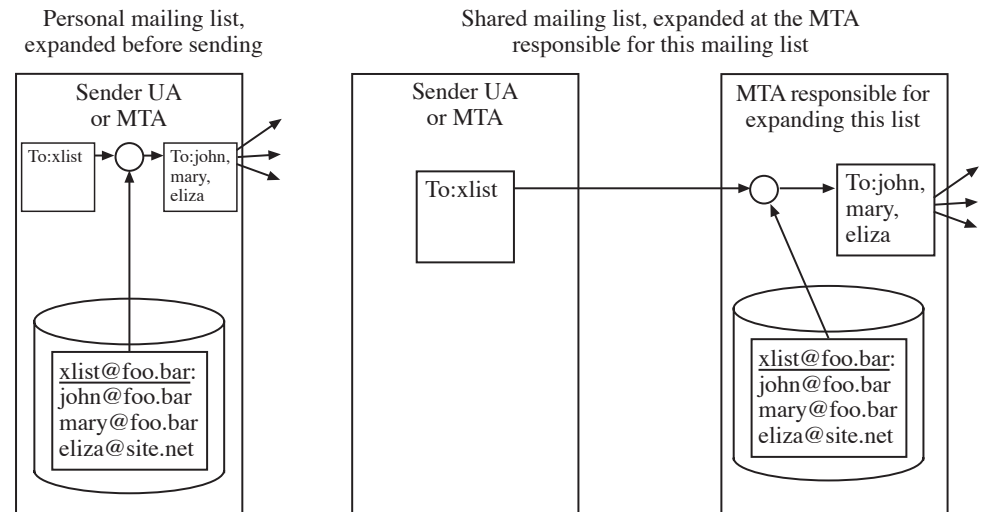
## Nested distribution lists



## Modes of distribution to many recipients



## Expansion of Nested Mailing Lists



## Loop control for Nested Distribution Lists



- (1) Full expansion by the originating UA or MTA.
- (2a) Trace list on the envelope, use to stop incoming messages.
- (2b) Trace list on the envelope, use to stop outgoing messages.
- (3) Registration system.
- (4a) Storing Message-ID-s with DL expanders.
- (4b) Storing content checksums with DL expanders.

X.400: Primarily 2a, Listserv: 4a and 4b, Usenet News: 4a

## List Headers (RFC 2369)

Meta-standard! Not specify a protocol, but specify how a mail header can specify a protocol for common actions on mailing lists:

List-Subscribe: <mailto:ietf-xml-mime-request@imc.org?body=subscribe>

List-Unsubscribe: <mailto:ietf-xml-mime-request@imc.org?body=unsubscribe>

List-Help: <http://www.imc.org/ietf-xml-mime/>

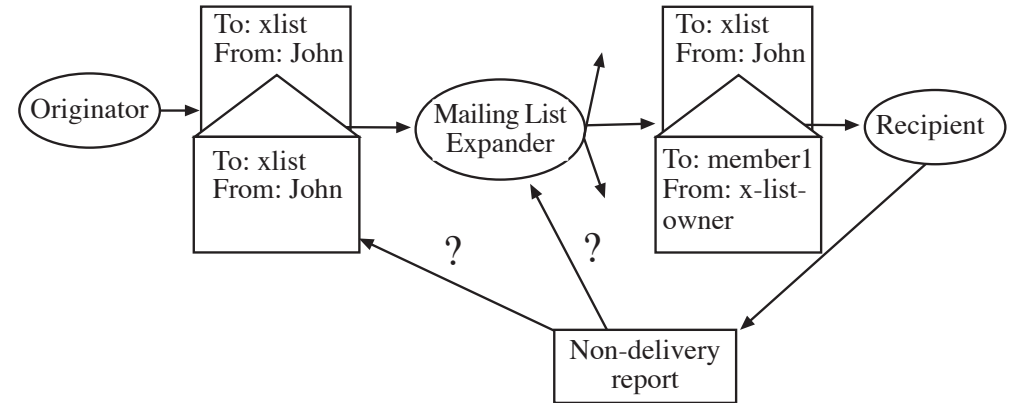
List-Archive: <http://www.imc.org/ietf-xml-mime/mail-archive/>

List-ID: <ietf-xml-mime.imc.org>

## Mailing Lists in Internet Mail

- No standardized loop control for nested lists
  - “-request”-convention
  - SMTP sender = address of list maintainer
- Non-delivery reports sent to SMTP sender

Note: Compare with Aliases



## Reply-To controversy

Original sender: Reply-To: My assistant ...

Mailing list: Reply-To: This mailing list ...

Mail programs often have two reply commands, “Reply to Sender” and “Reply to all”. How should these two commands interpret the “Reply-To” header??

## Gatewaying Usenet News to Mailing Lists

In Usenet news, the “Newsgroups” header indicates the name of the Newsgroup(s) to which this contribution was sent. Example:

`Newsgroups: alt.sex.fetisches.feet`

Some newsreaders provide a facility to answer personally by e-mail to only the author of a message. They sometimes copy the Newsgroups header to such answers.

Some gateways between Usenet news and e-Mail copy the Newsgroups header to the mail copy of a message, to indicate which newsgroups it comes from.

Thus: The Newsgroups header has two different interpretations:

1. The newsgroup to which this message was also sent
2. The newsgroup of the message, which this is a personal reply to.

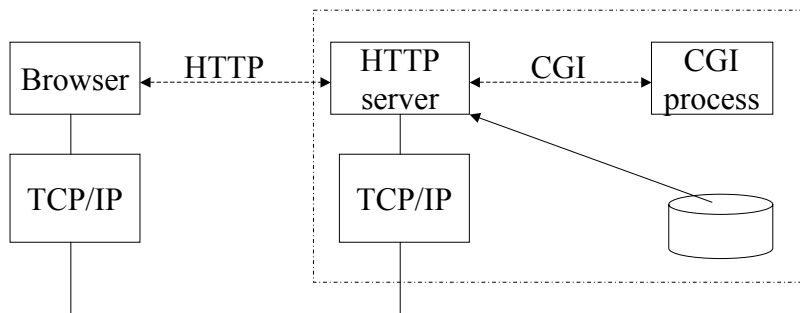
# CGI och CGI-programmering

Fredrik Kilander  
DSV

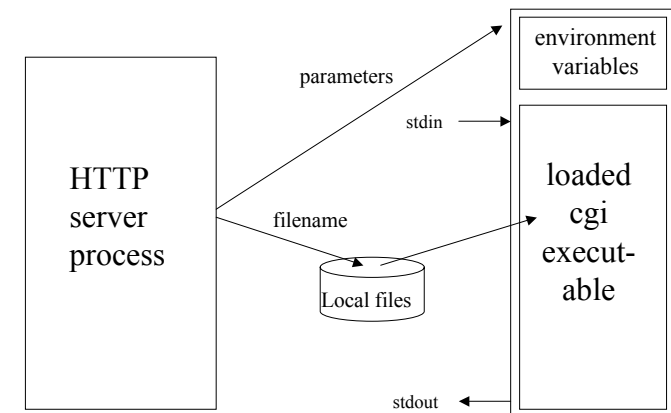
## Innehåll

- Common Gateway Interface (CGI)
- Alternativ för dynamiska WWW-sidor
- HTTP-servern
- CGI-processen
- Programmeringsspråk
- Säkerhet
- Applikationsdesign för WWW

## Common Gateway Interface



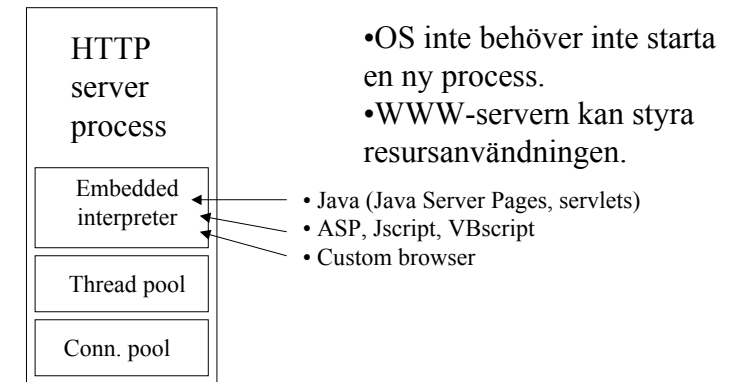
## Common Gateway Interface



## CGI: en de facto standard

- Informell överenskommelse
- Beskriver server och CGI-program
- Internet draft version 1.1, Juni 1999, (utgången)
- Version 1.2 (inga framsteg sedan 1998).
- <http://www.w3.org/CGI>
- <http://cgi-spec.golux.com>

## Kompletterande lösningar



- OS inte behöver inte starta en ny process.
- WWW-servern kan styra resursanvändningen.

- Java (Java Server Pages, servlets)
- ASP, Jscript, VBscript
- Custom browser

## Dynamiska websidor i WWW-läsaren

- Javascript, JScript, VBscript
- Java (applets)
- ActiveX
- Vendor plug-ins (Flash, Quicktime...)
- CGI är inte beroende av klientimplementationen

## HTTP-servern

- WWW-läsaren och servern använder HTTP
- WWW-läsaren anropar med GET eller POST
- GET : hämta URL
- GET : `www.bz.com/db.exe?nm=John+Smith&tel=123+987`
- POST : skicka data till servern och få svar
- POST : `<form action="http://www.bz.com/db.exe" method="POST">`

# HTTP-servern

- Skicka URL eller exekvera CGI bestäms av serverns konfiguration.
- Hitta exekverbar fil och skapa en process.
- Initiera miljövariabler (environment).
- Skicka POST data till stdin.
- Starta processen.
- Skicka stdout till klienten.

# CGI-programmet

- Läsa miljövariabler.
- Läsa stdin (om POST).
- Avkoda parametrar.
- Formatera och skicka ett svar på stdout.
- Svaret är ett Internet-dokument

# CGI-programmet

## Läsa miljövariabler

REQUEST\_METHOD  
QUERY\_STRING  
CONTENT\_LENGTH

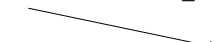
REQUEST\_METHOD = "GET"

REQUEST\_METHOD = "POST"

params = QUERY\_STRING

len = CONTENT\_LENGTH

params = read(stdin, len)



# Fler miljövariabler

- SERVER\_SOFTWARE
- SERVER\_NAME
- GATEWAY\_INTERFACE
- SERVER\_PROTOCOL
- SERVER\_PORT
- REQUEST\_METHOD
- PATH\_INFO
- PATH\_TRANSLATED
- SCRIPT\_NAME
- QUERY\_STRING
- REMOTE\_HOST
- REMOTE\_ADDR
- AUTH\_TYPE
- REMOTE\_USER
- REMOTE\_IDENT
- CONTENT\_TYPE
- CONTENT\_LENGTH
- HTTP\_ACCEPT
- HTTP\_USER\_AGENT
- HTTP\_\*

<http://hoohoo.ncsa.uiuc.edu/cgi/env.html>

# CGI-programmet

- Avkoda parametrarna (URL encoding, RFC 2396)
- GET: `www.bz.com/db.exe?name=%c5ke+%d6rn&age=22`
- POST: `name=%c5ke+%d6rn&age=22`
- Parametersträngen: `name=%c5ke+%d6rn&age=22`
- `s ::= namn '=' [värde] ['&' namn '=' [värde]] ...`

# Avkoda CGI-parametrarna

`name=%c5ke+%d6rn&age=22`

## 1. Dela upp vid '&'

`name=%c5ke+%d6rn&age=22`

`name=%c5ke+%d6rn` `age=22`

# Avkoda CGI-parametrarna

`name=%c5ke+%d6rn&age=22`

## 2. Dela upp vid '='

`name=%c5ke+%d6rn` `age=22`

`name` `%c5ke+%d6rn` `age` `22`

# Avkoda CGI-parametrarna

`name=%c5ke+%d6rn&age=22`

## 3. Byt '+' mot ' ' (blank)

`name %c5ke+%d6rn age 22`

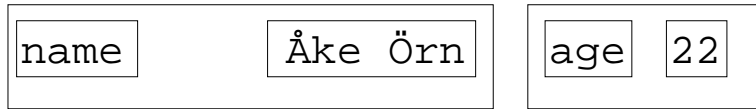
`name` `%c5ke %d6rn` `age` `22`

# Avkoda CGI-parametrarna

name=%c5ke+%d6rn&age=22

4. Byt '%xx' mot tecken

name %c5ke %d6rn age 22



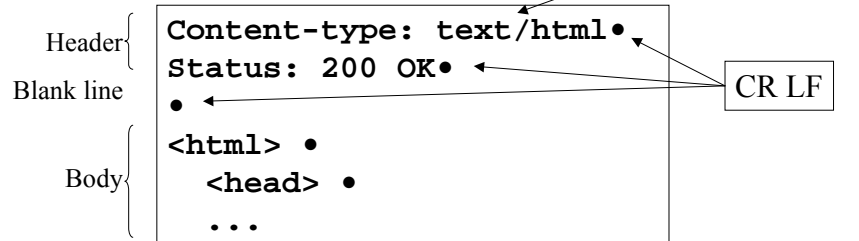
# Avkoda CGI-parametrarna

- 1. Dela upp vid '&' (par av namn och värden)
- 2. Dela upp vid '=' (mellan namn och värde)
- 3. Byt '+' mot ' ' (blank)
- 4. Byt '%xx' mot tecken (hexadecimal kod)
- Färdiga rutiner finns ofta att tillgå

# CGI-programmets respons

- Parsed Header Output (HTTP-servern kollar)
- <header> [ <blank line> <body>]

• T ex:



# Vad ett CGI-script måste göra (v 1.1)

- Avvisa ej understödda metoder\* med

Status: 405 Method Not Allowed

- \* GET POST DELETE HEAD PUT OPTIONS TRACE extension-method



## Vad ett CGI-script bör göra (v 1.1)

- Vara beredd på att dö närsomhelst (svårt!)
- Hantera PATH\_INFO eller svara 404 Not Found
- Verifiera CONTENT\_TYPE (indataformat)
- Vara vaksam på '//', '.' och '..' i sökvägar
- Inte generera relativa länkar utan <BASE>
- Sända headerns CGI-fält så snart som möjligt och före HTTP-fält

## Programmeringsspråk CGI

- Nästan vilket språk som helst:
- Perl-script ("cgi-script")
- Shell-script
- C, C++ (lång utvecklingstid)
- Java

## Språk inbäddade i HTTP-servern

- Java (Java Server Pages, servlets)
- ASP (Active Server Pages)
- Servermoduler

## Java Server Pages

- Källkoden innehåller text till klienten (HTML) och anrop till script-språk (vanligtvis Java).

```
<H1>Welcome to Our Store</H1>
<SMALL>Welcome,
<!-- User name is "New User" for first-time visitors -->
<% out.println(Utils.getUserNameFromCookie(request)); %>
To access your account settings, click
<A HREF="Account-Settings.html">here.</A>
</SMALL>
```

## Active Server Pages (ASP)

- Microsoft Internet Information Server IIS
- Körs i HTTP-servern (snabb start)
- Blandar flera språk och syntaxer i samma källkod:
- ASP-script, VBScript, JScript, SSI (servern), VBScript, JScript, JavaScript, HTML, CSS (klienten).

## Active Server Pages (ASP)

- + snabb utveckling
- + samma funktionalitet som CGI
- + bra stöd för databaskopplingar (VBscript)
- - kan hänga servern
- - svårt att avlusa
- - ingen modularitet, trasslig syntax
- - klarar inte hög belastning

## Servermoduler

- + total kontroll
- + effektivt
- - lång utveckling
- - kan hänga servern
- - kan vara svårt att avlusa
- - dålig portabilitet pga bindning till server

## Säkerhet

- Det som CGI-programmet får göra kan också besökarna göra.
- Förhindra ”buffer overruns” (rutinbibliotek)
- Kolla alla indata, inklusive CONTENT\_LENGTH.
- Låt aldrig indata exekveras utan inspektion.
- Förbjud allt och släpp in det som är tillåtet, inte tvärtom.
- Logga, och analysera loggarna.

## Applikationsdesign för WWW

- WWW är i grunden tillståndslöst.
- Tillstånd behövs i t ex dialoger.
- Cookies, dolda fält

## Applikationsdesign för WWW

- Var försiktig med finesser
- Hur stor är användargruppen?
- Vilken utrustning har de?
- Vilka krav ställer applikationen på dem?
- Vad kan man förvänta sig av dem?

## Applikationsdesign för WWW

- Enkla användargränssnitt:
- Använd inte mer teknik än som krävs.
- Undvik skärmrullning.
- Använd multipla indikatorer: ledtexter, färger, bilder

## Applikationsdesign för WWW

- Grundlig HTML-kodning:
- Följ en (1) standard.
- Stäng alla markörer som får stängas (XHTML).
- Koda speglade indata i formulär.
- Använd analysverktyg.
- Kolla med många www-klienter.

# XSL

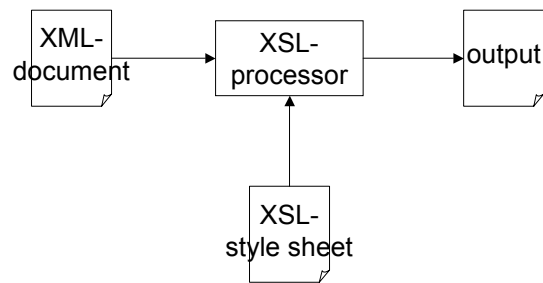
The  
Extensible Stylesheet Language  
Family

## XSL parts

- XSL Transformations (XSLT)
- XML Path Language (XPath)
- XSL Formatting Objects (XSL-FO)

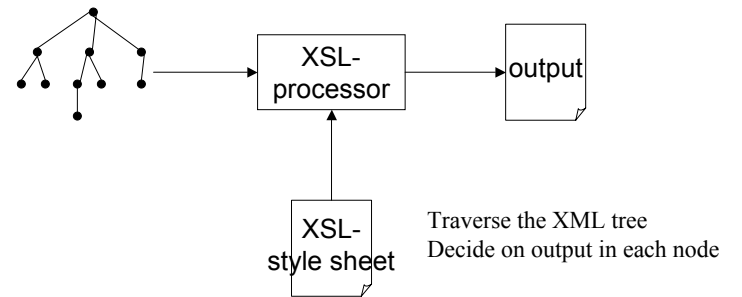
<http://www.w3c.org/Style/XSL>

## XSL process view



## XSL conceptual view

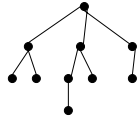
An XML document  
has a tree structure



# XML sample

## XSL conceptual view

An XML document has a tree structure



The programmer writes a set of templates.

Each template may match zero, one or more nodes.

The body of a template may specify output, or other templates to try in sequence.

The body of a template may also select any part of the tree and generate data from it, or apply templates to it.

The syntax is XML.

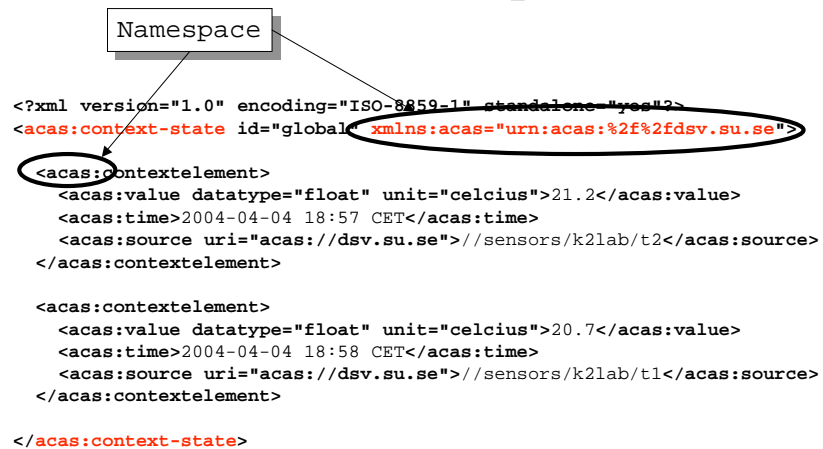
```
<?xml version="1.0" encoding="ISO-8859-1" standalone="yes"?>
<acas:context-state id="global" xmlns:acas="urn:acas:%2f%2fdsv.su.se">

  <acas:contextelement>
    <acas:value datatype="float" unit="celcius">21.2</acas:value>
    <acas:time>2004-04-04 18:57 CET</acas:time>
    <acas:source uri="acas://dsv.su.se"//sensors/k2lab/t2</acas:source>
  </acas:contextelement>

  <acas:contextelement>
    <acas:value datatype="float" unit="celcius">20.7</acas:value>
    <acas:time>2004-04-04 18:58 CET</acas:time>
    <acas:source uri="acas://dsv.su.se"//sensors/k2lab/t1</acas:source>
  </acas:contextelement>

</acas:context-state>
```

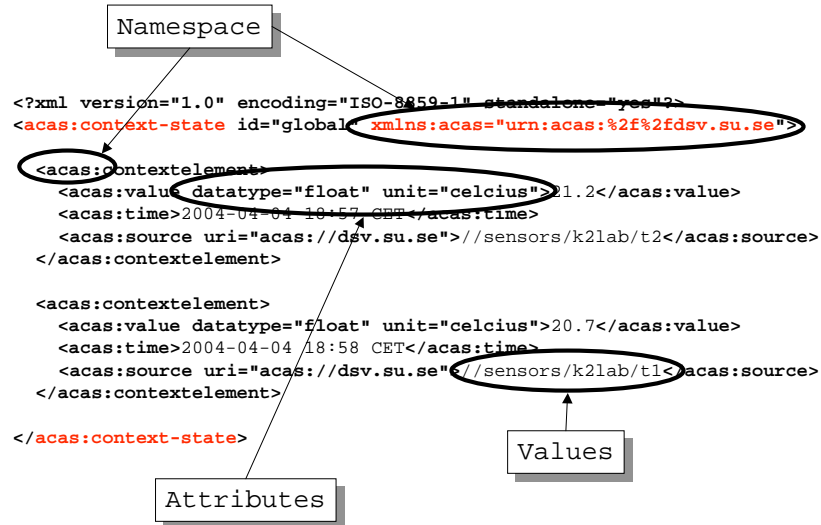
## XML sample



## XML sample



# XML sample



# XSLT sample

```

<?xml version="1.0" encoding="iso-8859-1"?>
<!-- q one.xsl -->
<!-- 08-mar-2004/FK -->
<xsl:stylesheet xmlns:acas="urn:acas:%2f%2fdsv.su.se"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  exclude-result-prefixes="acas"
  version="1.0">

  <xsl:output method="text" encoding="iso-8859-1" standalone="yes"/>

  <xsl:template match="/acas:context-state">
    <xsl:apply-templates
      select="acas:contextelement/acas:value[@unit='geo']"/>
    <xsl:apply-templates
      select="acas:contextelement/acas:value[@unit='celcius']"/>
  </xsl:template>

  <xsl:template match="acas:contextelement/acas:value[@unit='geo']">
    <xsl:text>LOCATION OF </xsl:text>
    <xsl:value-of select="../acas:source"/>
    <xsl:text> IS </xsl:text>
    <xsl:value-of select="."/>
    <xsl:text>&#xA;</xsl:text>
  </xsl:template>

```

Compendium 8 page 197

The xsl:stylesheet element wraps the whole document

# XSLT sample

```

<?xml version="1.0" encoding="iso-8859-1"?>
<!-- q one.xsl -->
<!-- 08-mar-2004/FK -->
<xsl:stylesheet xmlns:acas="urn:acas:%2f%2fdsv.su.se"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  exclude-result-prefixes="acas"
  version="1.0">

  <xsl:output method="text" encoding="iso-8859-1" standalone="yes"/>

  <xsl:template match="/acas:context-state">
    <xsl:apply-templates
      select="acas:contextelement/acas:value[@unit='geo']"/>
    <xsl:apply-templates
      select="acas:contextelement/acas:value[@unit='celcius']"/>
  </xsl:template>

  <xsl:template match="acas:contextelement/acas:value[@unit='geo']">
    <xsl:text>LOCATION OF </xsl:text>
    <xsl:value-of select="../acas:source"/>
    <xsl:text> IS </xsl:text>
    <xsl:value-of select="."/>
    <xsl:text>&#xA;</xsl:text>
  </xsl:template>

```

Declaration of namespace in input XML

# XSLT sample

```

<?xml version="1.0" encoding="iso-8859-1"?>
<!-- q one.xsl -->
<!-- 08-mar-2004/FK -->
<xsl:stylesheet xmlns:acas="urn:acas:%2f%2fdsv.su.se"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  exclude-result-prefixes="acas"
  version="1.0">

  <xsl:output method="text" encoding="iso-8859-1" standalone="yes"/>

  <xsl:template match="/acas:context-state">
    <xsl:apply-templates
      select="acas:contextelement/acas:value[@unit='geo']"/>
    <xsl:apply-templates
      select="acas:contextelement/acas:value[@unit='celcius']"/>
  </xsl:template>

  <xsl:template match="acas:contextelement/acas:value[@unit='geo']">
    <xsl:text>LOCATION OF </xsl:text>
    <xsl:value-of select="../acas:source"/>
    <xsl:text> IS </xsl:text>
    <xsl:value-of select="."/>
    <xsl:text>&#xA;</xsl:text>
  </xsl:template>

```

# XSLT sample

```

<?xml version="1.0" encoding="iso-8859-1"?>
<!-- q one.xml -->
<!-- 08-mar-2004/FK -->
<xsl:stylesheet xmlns:acas="urn:acas:%2f%2fdsv.su.se"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  exclude-result-prefixes="acas"
  version="1.0">
  <xsl:output method="text" encoding="iso-8859-1" standalone="yes"/>

  <xsl:template match="/acas:context-state">
    <xsl:apply-templates
      select="acas:contextelement/acas:value[@unit='geo']"/>
    <xsl:apply-templates
      select="acas:contextelement/acas:value[@unit='celcius']"/>
  </xsl:template>

  <xsl:template match="acas:contextelement/acas:value[@unit='geo']">
    <xsl:text>LOCATION OF </xsl:text>
    <xsl:value-of select="../acas:source"/>
    <xsl:text> IS </xsl:text>
    <xsl:value-of select="."/>
    <xsl:text>&#xA;</xsl:text>
  </xsl:template>

```

Type of output

# XSLT sample

```

<?xml version="1.0" encoding="iso-8859-1"?>
<!-- q one.xml -->
<!-- 08-mar-2004/FK -->
<xsl:stylesheet xmlns:acas="urn:acas:%2f%2fdsv.su.se"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  exclude-result-prefixes="acas"
  version="1.0">
  <xsl:output method="text" encoding="iso-8859-1" standalone="yes"/>

  <xsl:template match="/acas:context-state">
    <xsl:apply-templates
      select="acas:contextelement/acas:value[@unit='geo']"/>
    <xsl:apply-templates
      select="acas:contextelement/acas:value[@unit='celcius']"/>
  </xsl:template>

  <xsl:template match="acas:contextelement/acas:value[@unit='geo']">
    <xsl:text>LOCATION OF </xsl:text>
    <xsl:value-of select="../acas:source"/>
    <xsl:text> IS </xsl:text>
    <xsl:value-of select="."/>
    <xsl:text>&#xA;</xsl:text>
  </xsl:template>

```

Template to match root of input tree

# XSLT sample

```

<?xml version="1.0" encoding="iso-8859-1"?>
<!-- q one.xml -->
<!-- 08-mar-2004/FK -->
<xsl:stylesheet
  version="1.0">
  <xsl:output method="text" encoding="iso-8859-1" standalone="yes"/>

  <xsl:template match="/acas:context-state">
    <xsl:apply-templates
      select="acas:contextelement/acas:value[@unit='geo']"/>
    <xsl:apply-templates
      select="acas:contextelement/acas:value[@unit='celcius']"/>
  </xsl:template>

  <xsl:template match="acas:contextelement/acas:value[@unit='geo']">
    <xsl:text>LOCATION OF </xsl:text>
    <xsl:value-of select="../acas:source"/>
    <xsl:text> IS </xsl:text>
    <xsl:value-of select="."/>
    <xsl:text>&#xA;</xsl:text>
  </xsl:template>

```

Expressions that select and compose are written in the XPath language.

# XSLT sample

```

<?xml version="1.0" encoding="iso-8859-1"?>
<!-- q one.xml -->
<!-- 08-mar-2004/FK -->
<xsl:stylesheet xmlns:acas="urn:acas:%2f%2fdsv.su.se"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  exclude-result-prefixes="acas"
  version="1.0">
  <xsl:output method="text" encoding="iso-8859-1" standalone="yes"/>

  <xsl:template match="/acas:context-state">
    <xsl:apply-templates
      select="acas:contextelement/acas:value[@unit='geo']"/>
    <xsl:apply-templates
      select="acas:contextelement/acas:value[@unit='celcius']"/>
  </xsl:template>

  <xsl:template match="acas:contextelement/acas:value[@unit='geo']">
    <xsl:text>LOCATION OF </xsl:text>
    <xsl:value-of select="../acas:source"/>
    <xsl:text> IS </xsl:text>
    <xsl:value-of select="."/>
    <xsl:text>&#xA;</xsl:text>
  </xsl:template>

```

Output can be constants in the stylesheet or derived from the input.

## XSLT element samples

- `xsl:template` - *defines a template*
- `xsl:value-of` - *take the value of an XPath expression*
- `xsl:text` - *text constant*
- `xsl:apply-templates` - *recurse on a selected set of nodes*

## `xsl:template`

```
<order>
  <id>1234</id>
  <customer>Alice</customer>
  ...
</order>
```

```
<xsl:template match="order">
  ...
</xsl:template>
```

## `xsl:apply-templates`

```
<order>
  <id>1234</id>
  <customer>Alice</customer>
  ...
</order>
```

```
<xsl:template match="order">
  <xsl:apply-templates select="customer"/>
</xsl:template>
```

## `xsl:value-of`

```
<order>
  <id>1234</id>
  <customer>Alice</customer>
  ...
</order>
```

```
<xsl:template match="order">
  <xsl:apply-templates select="customer"/>
</xsl:template>

<xsl:template match="customer">
  <xsl:value-of select="."/>
</xsl:template>
```



# xpath

```
<xsl:template match="order">
  <xsl:apply-templates select="customer" />
</xsl:template>
```

```
<xsl:template match="customer">
  <xsl:value-of select="." />
</xsl:template>
```

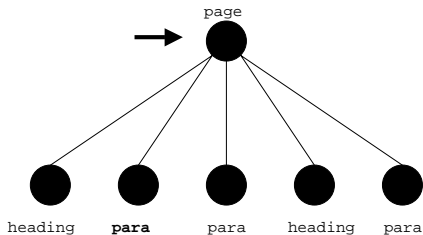
Axis, location paths, expressions

# xpath axis

- ancestor
- ancestor-or-self
- attribute
- child
- descendant
- descendant-or-self
- following
- following-sibling
- namespace
- parent
- preceding
- preceding-sibling
- self

Axis are used in location paths.  
 Paths can be absolute or relative.  
 Axis describe subsets of nodes.  
 Axis + expression = node set

# xpath expression



```
<page>
  <heading>...</heading>
  <para>...</para>
  <para>...</para>
  <heading>...</heading>
  <para>...</para>
</page>
```

```
... select="child::para[position()=1]" ...
```

# xpath core function library

- *number* last()
- *number* position()
- *number* count(*node-set*)
- *string* local-name(*node-set*?)
- *string* namespace-uri(*node-set*?)
- *string* name(*node-set*?)
- *string* string(*object*?)
- *string* concat(*string*, *string*, *string*\*)
- *boolean* starts-with(*string*, *string*)
- *boolean* contains(*string*, *string*)
- *string* substring-before(*string*, *string*)
- ...

# XSL in Java

- javax.xml.parsers.\*
- javax.xml.transform.\*
- org.w3c.dom.\*
- org.xml.sax.\*

End

All in Java SDK 1.4