

Web Server Logs Analyze Using the XML Technologies

Author: Tayeb L.

E-mail: Tayeb.Lemlouma@inrialpes.fr

July 2002.

We introduce here an approach to write and analyze server logs using the XML technology. The approach consists to transform the server log to an XML structure that we have defined and then to apply the log analyze. The analyze is done using XSLT and allows to have a clear idea about the server log in the form of a valid HTML [3] page generated from the XML log file.

1 The Server Logs Analyze

The approach that we have adopted in order to analyze and manipulate the server logs, works as follows:

- First we transform, using a Java program, the textual format of the server log to an XML [1] file. The supported textual format of the server log is the one compatible with the Apache 1.3.20 log format. This last is simple, it includes a set of string lines where each line corresponds to a visitor hit. Each line includes: the visitor IP address, the time and the date of the visit, the client request, the status code of the server reply, the file size of the requested content, the referrer URL, and the user agent type. In the following we give an example of an Apache 1.3.20 server log:

```
193.105.113.102 - - [01/Jul/2002:17:31:17 +0200] "GET
/people/Tayeb.Lemlouma/Papers/Programmation%20logique%20avec%20contraintes.pdf
HTTP/1.1" 206 1024 "-" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)"
193.105.113.102 - - [01/Jul/2002:17:31:18 +0200] "GET
/people/Tayeb.Lemlouma/Papers/Programmation%20logique%20avec%20contraintes.pdf
HTTP/1.1" 206 2395 "-" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)"
193.105.113.102 - - [01/Jul/2002:17:31:19 +0200] "GET
/people/Tayeb.Lemlouma/Papers/Programmation%20logique%20avec%20contraintes.pdf
HTTP/1.1" 206 66568 "-" "Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)"
208.13.106.20 - - [01/Jul/2002:17:40:56 +0200] "GET
/people/Tayeb.Lemlouma/MULTIMEDIA/CCPP/UPS-Package/UPSPfiles.html HTTP/1.0"
200 3332 "-" "Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0) Fetch API Request"
80.15.59.139 - - [01/Jul/2002:17:41:07 +0200] "GET
/people/Tayeb.Lemlouma/Papers/AdHoc_Presentation.pdf HTTP/1.1" 200 18535
"http://www.google.fr/search?q=%22applications+militaires%22+fr%C3%A9quence&hl=fr&lr=
&ie=UTF-8&oe=UTF8&start=20&sa=N" "Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)"
64.51.19.178 - - [01/Jul/2002:18:09:15 +0200] "GET
/people/Tayeb.Lemlouma/MULTIMEDIA/CCPP/UPS-Package/UPSPfiles.html HTTP/1.0"
200 3332 "-" "Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0) Fetch API Request"
64.51.19.178 - - [01/Jul/2002:18:19:08 +0200] "GET
/people/Tayeb.Lemlouma/NegotiationSchema/index.htm HTTP/1.0" 304 - "-" "Mozilla/4.0
(compatible; MSIE 5.5; Windows NT 5.0) Fetch API Request"
```

Figure 1. An example of a server log file

In order to avoid the size explosion of the generated XML, we have chosen a simple format that contains only the required information. In the following we give an example of the generated XML from the precedent log file:

```
<?xml version="1.0"?>
<ServerLog>
<Visitor IP="193.105.113.102" accessDate="01/Jul/2002:17:31:17 +0200" request="GET
/people/Tayeb.Lemlouma/Papers/Programmation%20logique%20avec%20contraintes.pdf
HTTP/1.1" statusCode="206" fileSize="1024" referrer="-" userAgent="Mozilla/4.0 (compatible;
MSIE 5.01; Windows NT)" />
<Visitor IP="193.105.113.102" accessDate="01/Jul/2002:17:31:18 +0200" request="GET
/people/Tayeb.Lemlouma/Papers/Programmation%20logique%20avec%20contraintes.pdf
HTTP/1.1" statusCode="206" fileSize="2395" referrer="-" userAgent="Mozilla/4.0 (compatible;
MSIE 5.01; Windows NT)" />
<Visitor IP="193.105.113.102" accessDate="01/Jul/2002:17:31:19 +0200" request="GET
/people/Tayeb.Lemlouma/Papers/Programmation%20logique%20avec%20contraintes.pdf
HTTP/1.1" statusCode="206" fileSize="66568" referrer="-" userAgent="Mozilla/4.0
(compatible; MSIE 5.01; Windows NT)" />
<Visitor IP="208.13.106.20" accessDate="01/Jul/2002:17:40:56 +0200" request="GET
/people/Tayeb.Lemlouma/MULTIMEDIA/CCPP/UPS-Package/UPSProfiles.html HTTP/1.0"
statusCode="200" fileSize="3332" referrer="-" userAgent="Mozilla/4.0 (compatible; MSIE 5.5;
Windows NT 5.0) Fetch API Request" />
<Visitor IP="80.15.59.139" accessDate="01/Jul/2002:17:41:07 +0200" request="GET
/people/Tayeb.Lemlouma/Papers/AdHoc_Presentation.pdf HTTP/1.1" statusCode="200"
fileSize="18535"
referrer="http://www.google.fr/search?q=%22applications+militaires%22+fr%C3%A9quence&
amp;hl=fr&amp;lr=&ie=UTF-8&oe=UTF8&start=20&sa=N"
userAgent="Mozilla/4.0 (compatible; MSIE 6.0; Windows 98)" />
<Visitor IP="64.51.19.178" accessDate="01/Jul/2002:18:09:15 +0200" request="GET
/people/Tayeb.Lemlouma/MULTIMEDIA/CCPP/UPS-Package/UPSProfiles.html HTTP/1.0"
statusCode="200" fileSize="3332" referrer="-" userAgent="Mozilla/4.0 (compatible; MSIE 5.5;
Windows NT 5.0) Fetch API Request" />
<Visitor IP="64.51.19.178" accessDate="01/Jul/2002:18:19:08 +0200" request="GET
/people/Tayeb.Lemlouma/NegotiationSchema/index.htm HTTP/1.0" statusCode="304"
fileSize="-" referrer="-" userAgent="Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0) Fetch
API Request" />
</ServerLog>
```

Figure 2. The generation of the XML log file

- After transforming the server log file to XML, we use the XSLT language [2] to analyze the XML content. One of the proposed processing is to organize the log information in elements concerning each visitor with giving its IP address, the number of hits, the date of the first access, the first visit server resource and the referrer visitor URL. The following XML file represents an example of the generated XML form:

```
<?xml version="1.0" encoding="UTF-8"?>
<analysResult> <accessNumber>0</accessNumber>
<totalAccessNumber>7</totalAccessNumber> <VisitorIP>193.105.113.102</VisitorIP>
<VisitorAccessNumber>3</VisitorAccessNumber>
<firstAccessDate>01/Jul/2002:17:31:17 +0200</firstAccessDate>
<firstVisitorRequest>GET
/people/Tayeb.Lemlouma/Papers/Programmation%20logique%20avec%20contraintes.pdf
HTTP/1.1</firstVisitorRequest>
<referrer>-</referrer> <VisitorIP>208.13.106.20</VisitorIP>
<VisitorAccessNumber>1</VisitorAccessNumber>
```

```

<firstAccessDate>01/Jul/2002:17:40:56 +0200</firstAccessDate>
<firstVisitorRequest>GET /people/Tayeb.Lemlouma/MULTIMEDIA/CCPP/UPS-
Package/UPSPProfiles.html HTTP/1.0</firstVisitorRequest>
<referrer>-</referrer> <VisitorIP>80.15.59.139</VisitorIP>
<VisitorAccessNumber>1</VisitorAccessNumber>
<firstAccessDate>01/Jul/2002:17:41:07 +0200</firstAccessDate>
<firstVisitorRequest>GET /people/Tayeb.Lemlouma/Papers/AdHoc_Presentation.pdf
HTTP/1.1</firstVisitorRequest>
<referrer>http://www.google.fr/search?q=%22applications+militaires%22+fr%C3%A9quence&
amp;hl=fr&lr=&ie=UTF-8&oe=UTF8&start=20&sa=N</referrer>
<VisitorIP>64.51.19.178</VisitorIP>
<VisitorAccessNumber>2</VisitorAccessNumber>
<firstAccessDate>01/Jul/2002:18:09:15 +0200</firstAccessDate>
<firstVisitorRequest>GET /people/Tayeb.Lemlouma/MULTIMEDIA/CCPP/UPS-
Package/UPSPProfiles.html HTTP/1.0</firstVisitorRequest>
<referrer>-</referrer> </analysResult>

```

Figure 3. An XSLT analyze of the server log file

The above XML form is generated from the XML log file (Figure 2) using the following XSLT style sheet:

```

<?xml version="1.0" encoding="iso-8859-1"?> <xsl:stylesheet
xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:template match="/"><analysResult>
<xsl:text>&#xA;&#xA;</xsl:text>
<accessNumber><xsl:value-of select="count(ServerLog/Visitor[@IP='4.33.55.30'])"
/></accessNumber>
<xsl:text>&#xA;</xsl:text>
<totalAccessNumber><xsl:value-of select="count(ServerLog/Visitor)" /></totalAccessNumber>
<xsl:text>&#xA;</xsl:text> <xsl:for-each select="ServerLog/Visitor">
<xsl:variable name="value" select="@IP"/>
<xsl:if test="count(preceding::Visitor[@IP=$value]) = 0">
<xsl:text>&#xA;</xsl:text>
<VisitorIP><xsl:value-of select="@IP"/></VisitorIP>
<xsl:text>&#xA;</xsl:text>
<VisitorAccessNumber><xsl:value-of
select="count(/ServerLog/Visitor[@IP=$value])"/></VisitorAccessNumber>
<xsl:text>&#xA;</xsl:text>
<firstAccessDate><xsl:value-of select="@accessDate"/></firstAccessDate>
<xsl:text>&#xA;</xsl:text>
<firstVisitorRequest><xsl:value-of select="@request"/></firstVisitorRequest>
<xsl:text>&#xA;</xsl:text>
<referrer><xsl:value-of select="@referrer"/></referrer>
<xsl:text>&#xA;</xsl:text>
</xsl:if>
</xsl:for-each><xsl:text>&#xA;</xsl:text>
</analysResult></xsl:template></xsl:stylesheet>

```

Figure 4. The XSLT stylesheet used in the server log analyze

- An other possible transformation of the XML server log is to analyze the log and output the result in the form of an HTML page that can be easily visualized. The following figure shows a possible analyze of the XML log file in the form of an HTML page:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
<META http-equiv="Content-Type" content="text/html; charset=UTF-8">

```

```

<title>Web Analys Result</title>
<meta content="text/html; charset=iso-8859-1" http-equiv="Content-Type">
</head>
<body text="#000000" bgcolor="#FFFFFF">
<h1 align="left">
<b>Server Web Analyze</b>
</h1>
<table border="0" width="97%">
<tr bgcolor="#FFFF00">
<td width="9%">
<div align="left">
<b><font face="Arial">Visitor IP Adress</font></b>
</div>
</td><td width="12%"><b><font face="Arial">Hits Number</font></b></td><td
width="30%"><b><font face="Arial">Date of the First Access</font></b></td><td
width="6%"><b><font face="Arial">First Request</font></b></td><td width="8%"><b><font
face="Arial">Referrer</font></b></td>
</tr>
<tr bgcolor="#EFEFEF">
<td bgcolor="#EFEFEF" width="9%"><b><font size="2" face="Arial"
color="#990000">193.105.113.102</font></b></td><td width="12%"><font size="2"
face="Arial">3</font></td><td width="30%"><font size="2" face="Arial">01/Jul/2002:17:31:17
+0200</font></td><td width="6%"><font size="2" face="Arial">GET
/people/Tayeb.Lemlouma/Papers/Programmation%20logique%20avec%20contraintes.pdf
HTTP/1.1</font></td><td width="8%"><font size="2" face="Arial">-</font></td>
</tr>
<tr bgcolor="#EFEFEF">
<td bgcolor="#EFEFEF" width="9%"><b><font size="2" face="Arial"
color="#990000">208.13.106.20</font></b></td><td width="12%"><font size="2"
face="Arial">1</font></td><td width="30%"><font size="2" face="Arial">01/Jul/2002:17:40:56
+0200</font></td><td width="6%"><font size="2" face="Arial">GET
/people/Tayeb.Lemlouma/MULTIMEDIA/CCPP/UPS-Package/UPSProfiles.html
HTTP/1.0</font></td><td width="8%"><font size="2" face="Arial">-</font></td>
</tr>
<tr bgcolor="#EFEFEF">
<td bgcolor="#EFEFEF" width="9%"><b><font size="2" face="Arial"
color="#990000">80.15.59.139</font></b></td><td width="12%"><font size="2"
face="Arial">1</font></td><td width="30%"><font size="2" face="Arial">01/Jul/2002:17:41:07
+0200</font></td><td width="6%"><font size="2" face="Arial">GET
/people/Tayeb.Lemlouma/Papers/AdHoc_Presentation.pdf HTTP/1.1</font></td><td
width="8%"><font size="2"
face="Arial">http://www.google.fr/search?q=%22applications+militaires%22+fr%C3%A9quenc
e&amp;hl=fr&amp;lr=&amp;ie=UTF-8&amp;oe=UTF8&amp;start=20&amp;sa=N</font></td>
</tr>
<tr bgcolor="#EFEFEF">
<td bgcolor="#EFEFEF" width="9%"><b><font size="2" face="Arial"
color="#990000">64.51.19.178</font></b></td><td width="12%"><font size="2"
face="Arial">2</font></td><td width="30%"><font size="2" face="Arial">01/Jul/2002:18:09:15
+0200</font></td><td width="6%"><font size="2" face="Arial">GET
/people/Tayeb.Lemlouma/MULTIMEDIA/CCPP/UPS-Package/UPSProfiles.html
HTTP/1.0</font></td><td width="8%"><font size="2" face="Arial">-</font></td>
</tr>
</table>
<p>
<font size="2">Analyze done using log2XML utility. <br>Author: Tayeb Lemlouma, <br>
July 2002.</font>
</p>
</body>
</html>

```

Figure 5. An HTML form of an XSLT analyze of the server log

Web Analsys Result - Microsoft Internet Explorer

Address http://opera.inrialpes.fr/people/Tayeb.Lemlouma/MULTIMEDIA/XSLT/log2XML/analysed.log.html

Server Web Analyze

Visitor IP Address	Hits Number	Date of the First Access	First Request	Referrer
193.105.113.102	3	01/Jul/2002:17:31:17 +0200	GET /people/Tayeb.Lemlouma/Papers/Programmation%20logique%20avec%20contraintes.pdf HTTP/1.1	-
208.13.106.20	1	01/Jul/2002:17:40:56 +0200	GET /people/Tayeb.Lemlouma/MULTIMEDIA/CCPP/UPS-Package/UPSProfiles.html HTTP/1.0	-
80.15.59.139	1	01/Jul/2002:17:41:07 +0200	GET /people/Tayeb.Lemlouma/Papers/AdHoc_Presentation.pdf HTTP/1.1	http://www.google.fr/search?q=%22applications+militaires%22+fr%3%A9quence&hl=fr&lr=&ie=UTF-8&oe=UTF8&start=20&sa=N
64.51.19.178	2	01/Jul/2002:18:09:15 +0200	GET /people/Tayeb.Lemlouma/MULTIMEDIA/CCPP/UPS-Package/UPSProfiles.html HTTP/1.0	-

Analyse done using log2XML utility.
 Author: Tayeb Lemlouma,
 July 2002.

Figure 6. The generated HTML page

2 The application

The source code used in the log server transformation is give as follows:

```
import java.io.*;
import java.lang.String;
import java.net.*;

public class log2XML {

    /**
     * Replace a char by a string of char in the input string
     */
    /**
     * @param str1 the input string
     * @param str2 the tring used to replace the input character
     * @param c the character to be replaced
     */
    public String replaceString(String str1, String str2, char c){
        String str="";
        for(int i=0; i<str1.length(); i++) if (str1.charAt(i)==c) str+=str2; else str+=str1.charAt(i);
        return str;
    }//replaceString

    /**
     * Build a part of the XML data
     */
    /**
     * @param out the output stream writer of the XML file
     * @param IP the string of the IP address
     * @param accessDate the string of the access date
     * @param request the string of the visitor request
     * @param statusCode the string of the status code
     * @param fileSize the string of the file size
     * @param referrer the string of the referrer
     * @param userAgent the string of the user agent
     */
    public void buildXMLData(OutputStreamWriter out, String IP, String accessDate, String request, String
    statusCode, String fileSize, String referrer, String userAgent){
        try{
            out.write("<Visitor IP=\"" +IP+"\" accessDate=\"" +accessDate+"\" request=\"" +request+"\"
            statusCode=\"" +statusCode+"\" fileSize=\"" +fileSize+"\" referrer=\"" +referrer+"\" userAgent=\"" +userAgent+"\"
            />\r\n");
        }catch(Exception e){
```

```

        System.err.println("can't write the XML file: "+e);
    }
} //end buildXMLData

/**
 * Extract the user agent information
 *
 * @param logLine a line of the server log
 */
public String extractUserAgent(String logLine){
    String userAgent="";
    int i,j,k,l,m;
    for(i=0; logLine.charAt(i)!=' ' && i<logLine.length(); i++){
        i+=35; //skip "- - ["
        for(j=i; logLine.charAt(j)!=''" && j<logLine.length(); j++){
            j+=2; //skip " "
            for(k=j; logLine.charAt(k)!=' ' && k<logLine.length(); k++){
                k+=1; //skip " "
                for(l=k; logLine.charAt(l)!=' ' && l<logLine.length(); l++){
                    l+=2; //skip ' "'
                    for(m=l; logLine.charAt(m)!=''" && m<logLine.length(); m++){
                        m+=3; //skip ' "'
                        for(int n=m; logLine.charAt(n)!=''" && n<logLine.length(); n++){userAgent+=logLine.charAt(n);}
                    }
                }
            }
        }
    }
    return userAgent;
} //end extractUserAgent

/**
 * Extract the referrer information
 *
 * @param logLine a line of the server log
 */
public String extractReferrer(String logLine){
    String referrer="";
    int i,j,k,l;
    for(i=0; logLine.charAt(i)!=' ' && i<logLine.length(); i++){
        i+=35; //skip "- - ["
        for(j=i; logLine.charAt(j)!=''" && j<logLine.length(); j++){
            j+=2; //skip " "
            for(k=j; logLine.charAt(k)!=' ' && k<logLine.length(); k++){
                k+=1; //skip " "
                for(l=k; logLine.charAt(l)!=' ' && l<logLine.length(); l++){
                    l+=2; //skip ' "'
                    for(int m=l; logLine.charAt(m)!=''" && m<logLine.length(); m++){referrer+=logLine.charAt(m);}
                }
            }
        }
    }
    return referrer;
} //end extractUserAgent

/**
 * Extract the file size information
 *
 * @param logLine a line of the server log
 */
public String extractFileSize(String logLine){
    String fileSize="";
    int i,j,k;
    for(i=0; logLine.charAt(i)!=' ' && i<logLine.length(); i++){
        i+=35; //skip "- - ["
        for(j=i; logLine.charAt(j)!=''" && j<logLine.length(); j++){
            j+=2; //skip " "
            for(k=j; logLine.charAt(k)!=' ' && k<logLine.length(); k++){
                k+=1;
                for(int l=k; logLine.charAt(l)!=' ' && l<logLine.length(); l++){fileSize+=logLine.charAt(l);}
            }
        }
    }
    return fileSize;
} //end extractUserAgent

/**
 * Extract the status code information

```

```

*
*@param logLine a line of the server log
*/
public String extractStatusCode(String logLine){
    String statusCode="";
    int i,j;
    for(i=0; logLine.charAt(i)!=' ' && i<logLine.length(); i++){
        i+=35; //skip "- - ["
        for(j=i; logLine.charAt(j)!=" " && j<logLine.length(); j++){
            j+=2; //skip " "
            for(int k=j; logLine.charAt(k)!=' ' && k<logLine.length(); k++){statusCode+=logLine.charAt(k);}
        }
    }
    return statusCode;
} //end extractUserAgent

/**
 * Extract the request information
 */
*@param logLine a line of the server log
*/
public String extractRequest(String logLine){
    String request="";
    int i;
    for(i=0; logLine.charAt(i)!=' ' && i<logLine.length(); i++){
        i+=35; //skip "- - ["
        for(int j=i; logLine.charAt(j)!=" " && j<logLine.length(); j++){request+=logLine.charAt(j);}
    }
    return request;
} //end extractRequest

/**
 * Extract the date information
 */
*@param logLine a line of the server log
*/
public String extractDate(String logLine){
    String extractDate="";
    int i;
    for(i=0; logLine.charAt(i)!=' ' && i<logLine.length(); i++){
        i+=6; //skip "- - ["
        for(int j=i; logLine.charAt(j)!=']' && j<logLine.length(); j++){extractDate+=logLine.charAt(j);}
    }
    return extractDate;
} //end extractDate

/**
 * Extract the IP information
 */
*@param logLine a line of the server log
*/
public String extractIP(String logLine){

    String extractIP="";
    for(int i=0; logLine.charAt(i)!=' ' && i<logLine.length(); i++) {extractIP+=logLine.charAt(i);}
    return extractIP;
} //end extractIP

/**
 * Create the log in XML
 * Format:
 * <?xml version="1.0"?>
 * <ServerLog>
 *     <Visitor IP="4.33.55.30" accessDate="30/Jun/2002:04:02:27 +0200" request="GET
 *     /people/Tayeb.Lemlouma/NegotiationSchema/index.htm HTTP/1.0" statusCode="200" fileSize="2674"
 *     referrer="-" userAgent="Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0) Fetch API Request" />
 *     <Visitor .. />
 *     ...

```

```

*      <Visitor .. />
* </ServerLog>
*
*@param   logFileName   The file name of the log.
*@param   xmlFileName   The file name of the output XML file.
*@param   serverType    The type of the server that determines the format of the log
*/
public log2XML(String logFileName, String xmlFileName, String serverType){
    try {
        String logLine;
        String IP;
        String accessDate;
        String request;
        String statusCode;
        String fileSize;
        String referrer;
        String userAgent;
        if(serverType.equals("Apache/1.3.20")){
            OutputStreamWriter out;
            FileOutputStream fout = new FileOutputStream(xmlFileName);
            out = new OutputStreamWriter(fout, "ISO-8859-1");

            FileReader fr = new FileReader(logFileName);
            BufferedReader br = new BufferedReader(fr);

            out.write("<?xml version='1.0'?>\r\n");
            out.write("<ServerLog>\r\n");

            while ((logLine=br.readLine())!=null){
                IP = extractIP(logLine);
                accessDate = extractDate(logLine);
                request = extractRequest(logLine);
                statusCode = extractStatusCode(logLine);
                fileSize = extractFileSize(logLine);
                referrer = extractReferrer(logLine);
                userAgent = extractUserAgent(logLine);
                request = replaceString(request,"&","'");
                referrer = replaceString(referrer, "&","'");
                buildXMLData(out, IP, accessDate, request,statusCode, fileSize,
                    referrer, userAgent);
            }
            out.write("</ServerLog>\r\n");
            out.close();
        }else{System.err.println("The server log type: "+serverType+"is not yet supported");}

    }catch(Exception e){

    }
}

/**
 * Written to test the log2XML method
 */
public static void main(String[] args) throws Exception {
    System.out.println("XML file creating...");
    //new log2XML("log.txt","file.xml", "Apache/1.3.20");
    System.out.println("XML file created.");
}
}

```


3 How to run the application?

- 1- Download the different resources: Log2XML.class, logAnalyzer.xsl, logAnalyzer2.xsl
- 2- To transform the server log file, access_log.1, (which must be compatible with Apache/1.3.20 log format), run: **java Log2XML access_log.1**
- 3- The generated file 'output.xml' represents the XML server log. It can be so processed and used to do many analyzes.
- 4- To transform the XML server log to an XML file in the form of the figure 3, apply the XSLT style sheet: **logAnalyzer.xsl**
- 5- To transform the XML server log to an HTML page in the form of the figure 4, apply the style sheet: **logAnalyzer2.xsl**

References

- [1] W3C, Extensible Markup Language (XML) 1.0, W3C Recommendation: <http://www.w3.org/TR/1998/REC-xml-19980210>, 10 February 1998.
- [2] W3C, The Extensible Stylesheet Language (XSL), <http://www.w3.org/Style/XSL/>.
- [3] W3C, HyperText Markup Language, <http://www.w3.org/MarkUp/>.