

EXPERIMENT ON JSP, XML AND HDF

Kun Yan
Robert E. McGrath

September, 2001

CONTENTS

1. Introduction
2. Environment setting
3. Experiment Design
4. Experiment Implementation
5. Experiment Results
6. Summary
7. Acknowledgment
8. References

1. INTRODUCTION

This summer several experiments were done with Java Server Platform (JSP), XML, and HDF-5 for remote accessing scientific data. This experiment is using Tomcat as web server and JSP servlet container, converting HDF5 file into XML file and further transforming XML file into HTML file.

2. ENVIRONMENT SETTING

Before starting the experiment, we need to set up the required environment. First, we need to install the web server (using Apache), the JSP/Servlet container (Tomcat), and the communicator between Apache and Tomcat, mod_jk. The following table provides the information about where

to find the source/binary code and the instruction of configuration and installation. Hdf5 library and some tools are also needed as well as the DTD of HDF.

	Download	Instruction of Configuration and Installation
Apache	http://httpd.apache.org/dist/httpd/	http://httpd.apache.org/docs/install.html
Tomcat 3.3	http://jakarta.apache.org/builds/jakarta-tomcat/release/v3.3-b1/bin/	http://jakarta.apache.org/tomcat/tomcat-3.3-doc/tomcat-ug.html
mod_jk	http://jakarta.apache.org/downloads/binindex.html	http://jakarta.apache.org/tomcat/tomcat-3.3-doc/mod_jk-howto.html
hdf5 tools	http://hdf.ncsa.uiuc.edu/HDF5/	
DTD for hdf (5.1.4)	http://hdf.ncsa.uiuc.edu/DTDs/HDF5-File-1.4.txt	

Table 1. Apache, Tomcat, and other components used in this experiment.

The “mod_jk” is a Tomcat-Apache plugin that handles the communication between Tomcat and Apache. We need to reconfigure Tomcat and Apache to make them working with mod_jk. For Apache1.3.20 and Tomcat 3.3, the reconfiguration is much simpler than with earlier versions. (See http://jakarta.apache.org/tomcat/tomcat-3.3-doc/mod_jk-howto.html.)

The main reason that we need to connect Tomcat to Apache is that Tomcat is not as fast as Apache when it comes to static pages. After adding Tomcat, Apache just handles the static content, such as images and HTML documents, and forwards all requests for dynamic content to Tomcat. [2]

After installing the Apache and Tomcat, we need to set the following environment variables:

APACHE_HOME	set	Path_to_Apache/
TOMCAT_HOME	set	Path_to_Tomcat/
JAVA_HOME	set	Path_to_jdk/
CLASSPATH	add	Path_to_jdk/jre/lib/rt.jar
CLASSPATH	add	Path_to_java-hdf5/
CLASSPATH	add	Path_to_xalan.jar/
CLASSPATH	add	Path_to_xerces.jar/
LD_LIBRARY_PATH	add	Path_to_libjhdf5.so/

Table 2. environment variables that need to set.

Note: we are using Apache 1.3.20 and Tomcat 3.3 for the summer experiment. Since there is no binary mod_jk.so provided for Solaris and Unix, we use Tomcat as stand-alone servlet container for this experiment. I have set up both Tomcat and Apache in Windows 2000 and they are working fine. We are using jdk1.3.1 instead of jdk1.2 since there are some bugs in the JVM of jdk1.2 which would cause “segmentation violation” problem in JSP.

3. EXPERIMENT DESIGN

The first goal of this experiment was to explore the JSP and XML technology, to learn the terminology and techniques. The system was set up, and small JSP examples use. From this experience, a simple experiment was set up. Figure 1 shows the UML sequence diagram for the experiment.

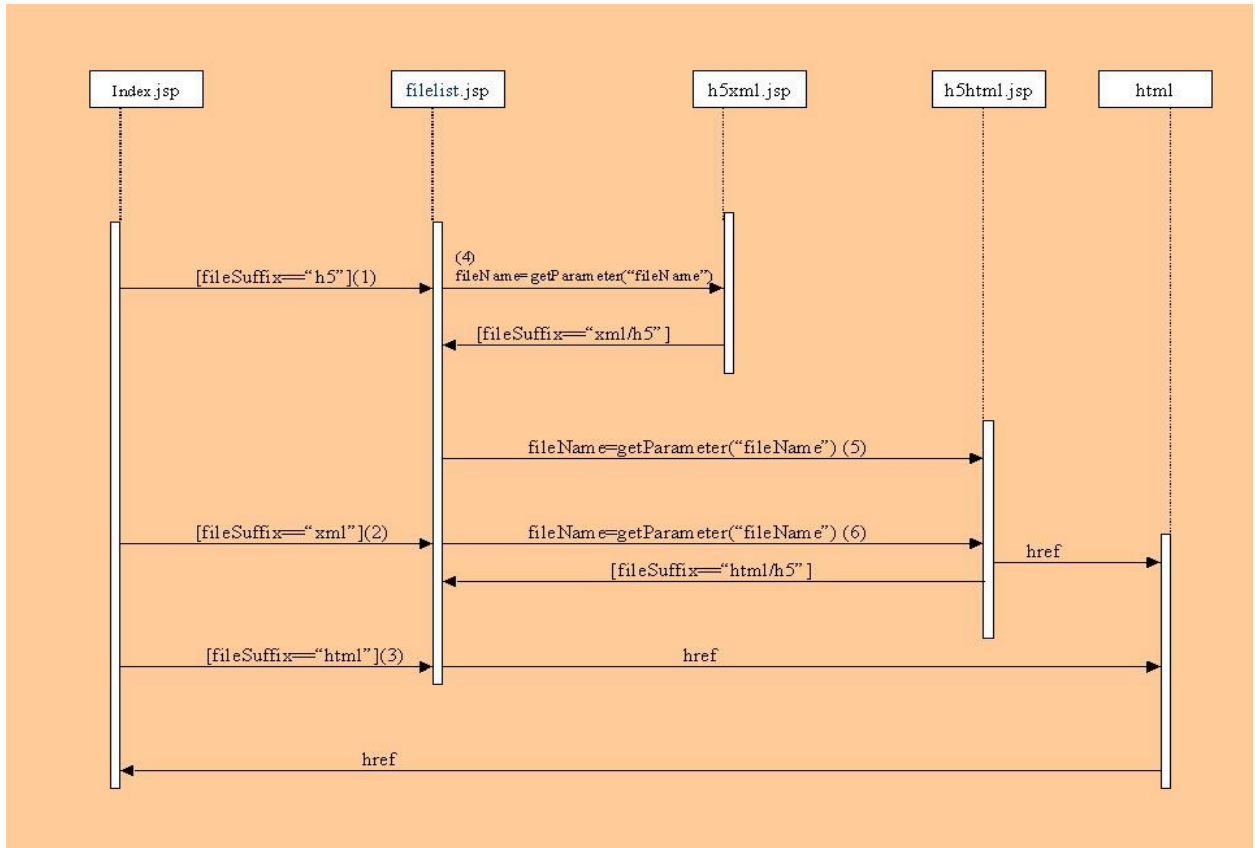
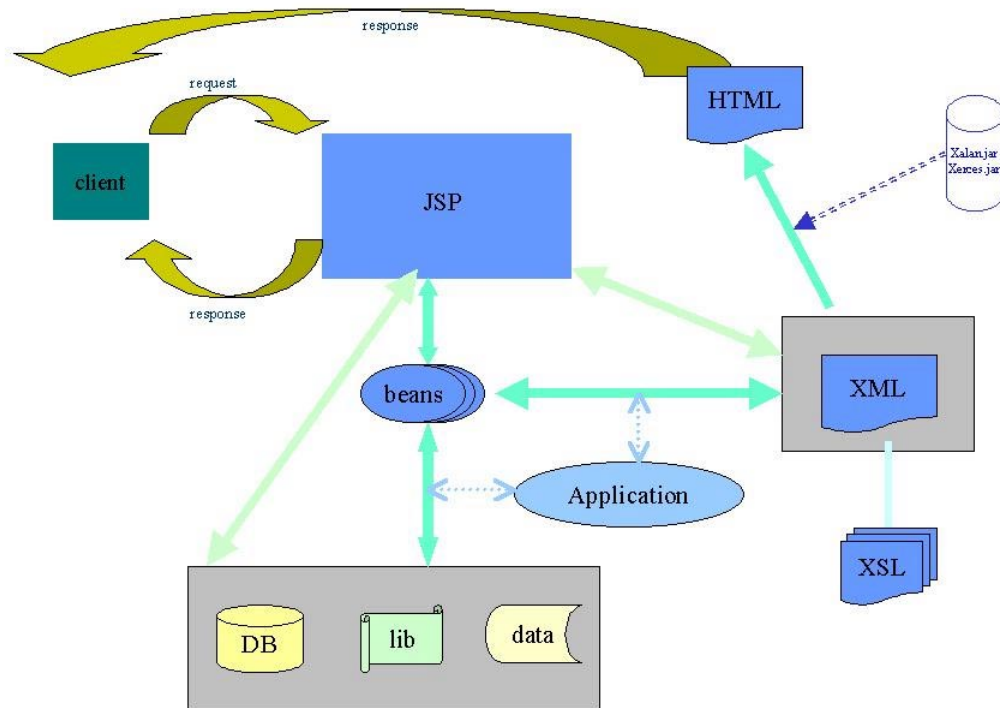


Figure 1. The UML sequence diagram of experiment design

The design was based on the understanding of how JSP works [2, 3, 4]. Figure 2 shows a functional diagram of the how JSP works in this experiment.



How JSP Works

Figure 2. How JSP works in this experiment

In addition, we also need to understand the inside of JSP/Servlet engine and how it processes after receiving the client's request [2, 5].

When the client request a JSP page, the file's extension, .jsp, tells the server a special handling needed. Then the request is forwarded to Tomcat from Apache. The special handling involves four steps [5]:

- The JSP engine parses the page and creates a Java source file.*
- It then compiles the source file into a class file. The class file is a servlet, and from this point on, the servlet engine handles the class file in the same manner as all other servlets.*
- The servlet engine loads the servlet class for execution.*
- The servlet executes and streams back the results to the requestor.*

Figure 3 shows the flow of a client's request. (Adapted from Timothy Eden and Ed Ludke's Introducing JavaServer Pages) [5].

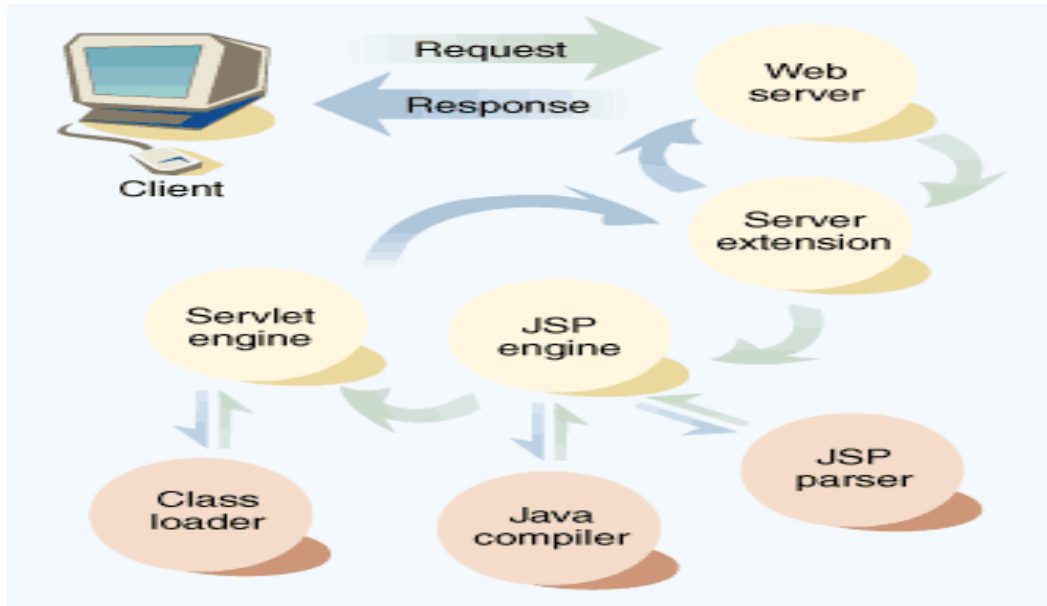


Figure 3. The flow of a request. (Adapted from [5].)

4. EXPERIMENT IMPLEMENTATION

This experiment was implemented by four JSP pages and three JavaBean classes. The pages of HDF5, XML and HTML file lists are generated by the same JSP file and JavaBeans with different parameters passing to corresponding JSP pages.

The following issues have caught more attentions during implementation phase:

1. Where to put JavaBeans classes?

Java classes have to be in the /PATHTOTOMCAT/webapps/APPLICATION/WEB-INF/classes. Jar files could be put into /PATHTOTOMCAT/webapps/APPLICATION/WEB-INF/lib.

2. Syntax of JSP.

In order to access bean, the following lines have to be added to JSP pages:

```

<jsp:useBean id="beanid" class="BeansClassName" scope=""/>
<jsp:setProperty name="beanid" property="*" />
</jsp:useBean>

```

3. Setting correct classpath of xalan.jar and xerces.jar (for converting xml file to html file).

The following figure shows the source code of *h5xml.jsp*.

```

<%@ page language="java" contentType="text/html" %>
<jsp:useBean id="h5xml" class="H5XMLBean" scope="session"/>
<jsp:setProperty name="h5xml" property="*" />

<%@ include file="header.html" %>

<h5>Testing: convert h5 file to xml file</h5>
<%=h5xml.getFileName()%>
<ul>
  <li>HDF5 file name:<%=h5xml.getH5fileName()%>
  <li>XML file name:<%=h5xml.getXmlfileName()%>
  <li>HDF5 lib path:<%=h5xml.getLibPath()%>
</ul>
  Converting ....
  <% h5xml.h5ToXml();
    String done = "testing"; %>
  <%=done%>

<ul>
  <li><a href="/hdf5/fileList.jsp?fileSuffix=xml">List of xml files</a>
  <li><a href="/hdf5/fileList.jsp?fileSuffix=h5">List of hdf5 files</a>
</ul>
<BR>
<a href="/hdf5/index.jsp">home</a>
<BR>

<%@ include file="footer.html" %>

```

Figure 4. sample JSP source code

Obviously, the content and presentation were separated by using JavaBeans. The bean is the middleware between data library and JSP. For future work, we could design several layers of beans and Java classes to accomplish complex task.

5. EXPERIMENT RESULTS

The results of this experiment show that we could receive request from client through browser, invoke corresponding JavaBeans classes used in JSP, access HDF5 library, convert HDF5 file into XML and then into HTML file, and send HTML file back to client.

The sample files of HDF5 could be successfully and quickly converted to XML and further HTML. Also we have tested some real NASA data files. Those files were converted from HDF4 to HDF5 by *b4toh5* utility. Here is the list of HDF5 files that can be converted in this experiment:

HDF5 file converted from NASA dataset	Data Size (byte)
NISE_SSMIF11_19911227.h5	2M
CER_ES8_Terra-FM2_Test_SCF_016011.20000830.subset_70_20_-140_-40.20001012_204110Z.h5	76M
98034001632_GOES08_IMAGER.h5	24M
avhrr8kmmonthly.h5	24M
balloon_sp.h5	51k

Table 3. list of HDF5 file (NASA data) that can be converted to XML and HTML in this experiment.

The following figures are snapshots of some JSP pages. It shows a sequence of requests from a client.

1. Select “hdf5 file” and send the request (Figure 5)
2. Get the hdf5 file list, select file “tattr.h5”, request converting to xml file (Figure 6)
3. Get the converting result, request the list of xml file (Figure 7)
4. Get the xml file list, select “tattr.h5.xml”, request transforming to html file (Figure 8)
5. Get the transforming result, request the transformed html file (Figure 9)
6. Get tattr.h5.xml.html file (Figure 10)

Here is the link to the JSP pages: <http://arabica.ncsa.uiuc.edu:7070/hdf5/> .

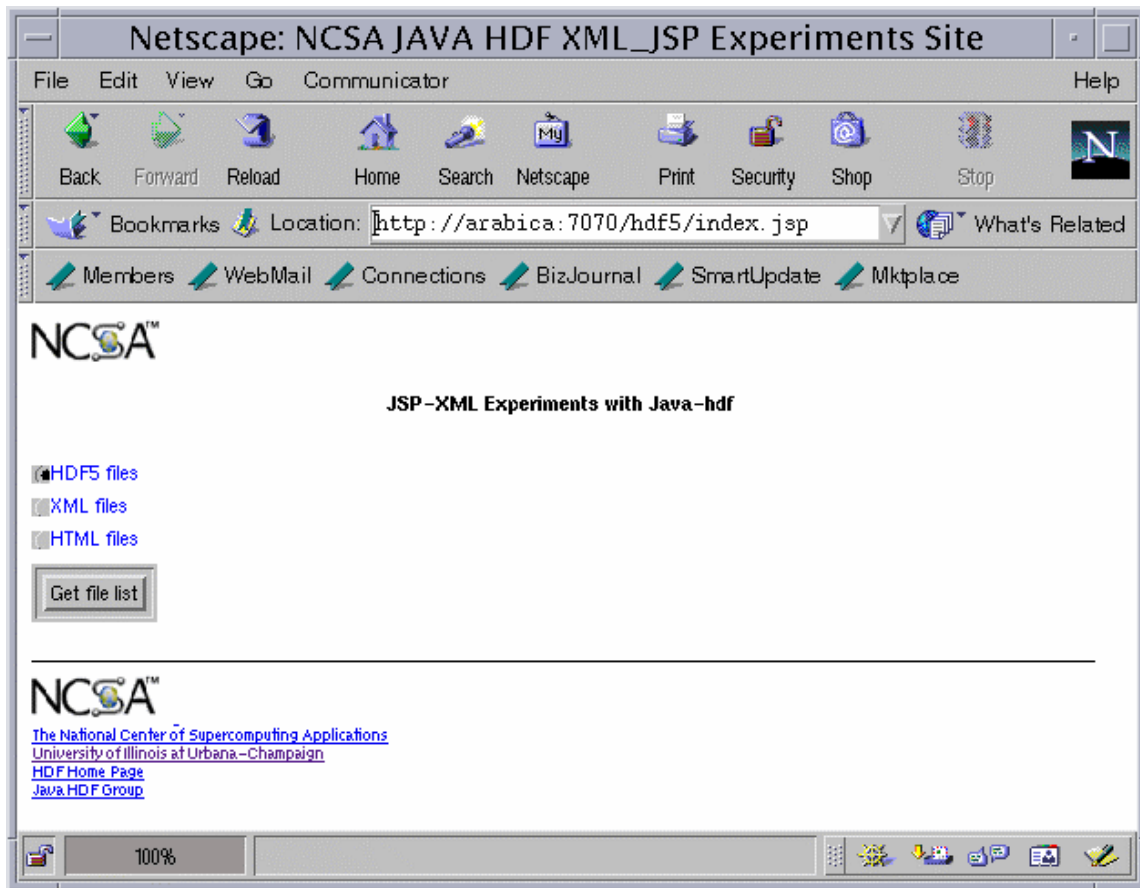


Figure 5. Index page

Netscape: NCSA JAVA HDF XML_JSP Experiments Site

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Bookmarks Location: <http://arabica:8080/hdf5/fileList.jsp?fileSuffix=h5> What's Related

Members WebMail Connections BizJournal SmartUpdate Mktplace

NCSA™

JSP-XML Experiments with Java-hdf

File Name	Size	Last Modified	to XML	to HTML
SDScompound.h5	1608	Thu Jul 26 15:16:08 CDT 2001	get xml file	get html file
copy1.h5	1440	Thu Jul 26 15:16:08 CDT 2001	get xml file	get html file
copy2.h5	1440	Thu Jul 26 15:16:08 CDT 2001	get xml file	get html file
dset.h5	1488	Thu Jul 26 15:16:08 CDT 2001	get xml file	get html file
groups.h5	3440	Thu Jul 26 15:16:08 CDT 2001	get xml file	get html file
iterate.h5	2296	Thu Jul 26 15:16:08 CDT 2001	get xml file	get html file
mount1.h5	1768	Thu Jul 26 15:16:08 CDT 2001	get xml file	get html file
mount2.h5	1472	Thu Jul 26 15:16:08 CDT 2001	get xml file	get html file
ref_obj.h5	3452	Thu Jul 26 15:16:08 CDT 2001	get xml file	get html file
ref_reg.h5	5904	Thu Jul 26 15:16:08 CDT 2001	get xml file	get html file
sds.h5	1464	Thu Jul 26 15:16:08 CDT 2001	get xml file	get html file
tattr.h5	1208	Thu Jul 26 15:16:08 CDT 2001	get xml file	get html file
tgroup.h5	11096	Thu Jul 26 15:16:16 CDT 2001	get xml file	get html file
NISE.h5	2125852	Thu Jul 26 15:47:49 CDT 2001	get xml file	get html file

Total number of files:14

[home](#)

NCSA™
 The National Center of Supercomputing Applications
 University of Illinois at Urbana-Champaign
[HDF Home Page](#)
[Java HDF Group](#)

Figure 6. HDF5 file list

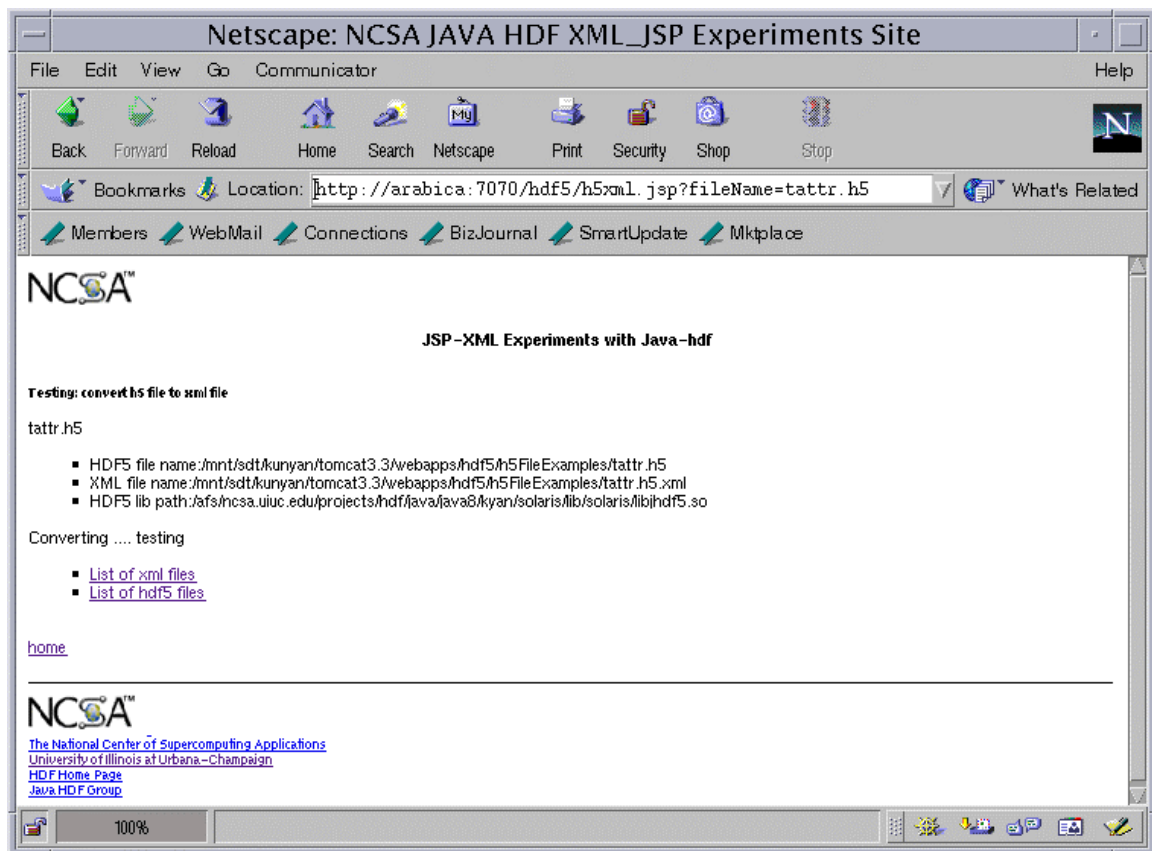


Figure 7: Result page after converting tattr.h5 to tattr.h5.xml

Netscape: NCSA JAVA HDF XML_JSP Experiments Site

File Edit View Go Communicator Help

Back Forward Reload Home Search Netscape Print Security Shop Stop

Bookmarks Location: <http://arabica:8080/hdf5/fileList.jsp?fileSuffix=xml> What's Related

Members WebMail Connections BizJournal SmartUpdate Marketplace

NCSA™

JSP-XML Experiments with Java-hdf

File Name	Size	Last Modified	to HTML
tattr.h5.xml	2239	Mon Jul 30 13:40:24 CDT 2001	get_html file
groups.h5.xml	433	Wed Aug 08 09:28:28 CDT 2001	get_html file
tgroup.xml	1119	Thu Jul 26 15:16:08 CDT 2001	get_html file
copy1.h5.xml	678	Wed Aug 01 13:29:25 CDT 2001	get_html file
SDScompound.h5.xml	1283	Fri Aug 03 13:21:39 CDT 2001	get_html file
copy2.h5.xml	678	Tue Aug 07 17:15:40 CDT 2001	get_html file
dset.h5.xml	1133	Wed Aug 08 14:08:52 CDT 2001	get_html file
tgroup.h5.xml	1119	Mon Jul 30 10:39:02 CDT 2001	get_html file
iterate.h5.xml	1476	Fri Jul 27 15:44:25 CDT 2001	get_html file
mount1.h5.xml	249	Tue Aug 07 09:37:27 CDT 2001	get_html file
mount2.h5.xml	689	Tue Aug 07 09:37:54 CDT 2001	get_html file
ref_obj.h5.xml	1527	Wed Aug 01 10:03:07 CDT 2001	get_html file
ref_reg.h5.xml	1134	Tue Aug 07 17:15:59 CDT 2001	get_html file
sds.h5.xml	728	Mon Aug 06 15:25:41 CDT 2001	get_html file
NISE.h5.xml	7420447	Mon Jul 30 10:39:37 CDT 2001	get_html file
MISR_AM1_AS_AEROSOL_P027_0000027_01_dw.h5.xml	49152	Mon Jul 30 15:02:49 CDT 2001	get_html file
MISR_AM1_AS_LANDSFC_P027_0000027_01_dw.h5.xml	40960	Mon Jul 30 13:12:30 CDT 2001	get_html file
tahoe-north-middle.h5.xml	2441216	Mon Jul 30 13:48:53 CDT 2001	get_html file

100%

Figure 8: XML file list

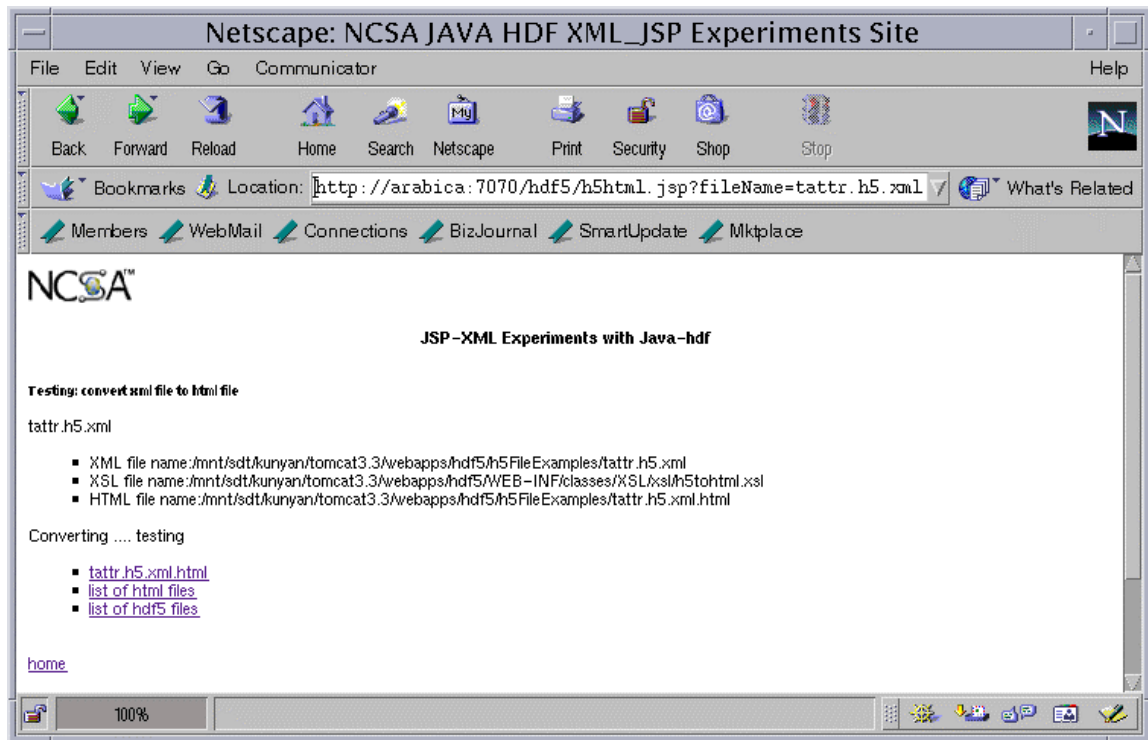


Figure 9: Result page after converting tattr.h5.xml to tattr.h5.xml.html

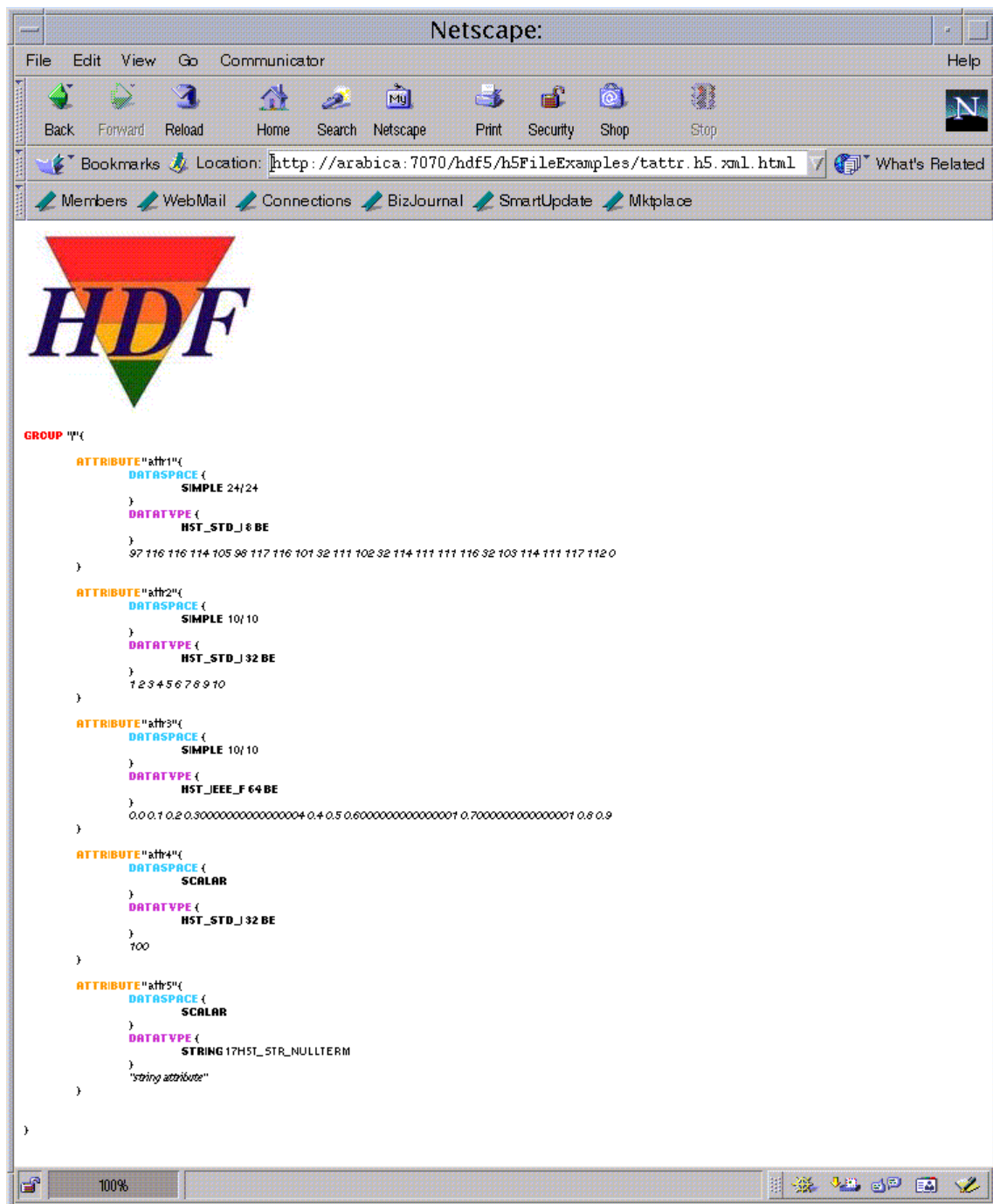


Figure 10: *tattr.h5.xml.html*

6. SUMMARY

This experiment investigate many new technologies, especially JSP and XML associated techniques, These technologies could be used to design and implement an application of remote scientific data access.

There are some well done applications to support web based data access for HDF , such as DIAL [7] and SDB [1]. These applications use CGI. We could also use JSP and JavaBeans to implement an application package with similar functionalities. Also, we should consider how to combine XML technology with Java and JSP technology to design our own Client/Server architecture.

There are many ways to use JSP and XML together. We could generate XML with JSP, as in this experiment, generate beans from XML, and transform XML into JSP [4, 5]. These ideas need to be investigated further.

Also there are other issues essential to our design. Since scientific data files usually are large and user may not want to acquire entire data at one time, we have to make the application efficient. Using XML technology, Xlink, Xpointer, and XPath could allow partial data access [8, 9]. Also we may need to use some sort of cache techniques to store XML objects in memory.

Based on this experiment, we have developed a sequence JSP pages that allow loading hdf files from URL and then convert the file to other possible files with single click (see <http://arabica.ncsa.uiuc.edu:7070/hdf5/PathList.jsp>) . Currently the following conversions can be accomplished:

- hdf4 to hdf5, xml and html
- hdf5 to xml and html
- xml to hdf5 and html

7. ACKNOWLEDGMENT

This report is based upon work supported in part by a Cooperative Agreement with NASA under NASA grant NAG 5-2040. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Aeronautics and Space Administration.

Other support provided by NCSA and other sponsors and agencies [6].

8. REFERENCES

1. Scientific Data Browser (SDB)
<http://hdf.ncsa.uiuc.edu/sdb/sdb.html>
2. JavaServer Pages™ (JSP™)
<http://java.sun.com/products/jsp/>
3. Alex Chaffee, “Using XML and JSP together”
http://www.javaworld.com/javaworld/jw-03-2000/jw-0331-ssj-jspxml_p.html
4. Kammie Kayl, “Creating web services with Java technology and Xml”
<http://www.sun.com/software/cover/2001-0530/>
5. Timothy Eden and Ed Ludke, “Introducing JavaServer Pages”
<http://www.devx.com/upload/free/features/javapro/2000/04apr00/te0004/te0004-1.asp>
6. <http://hdf.ncsa.uiuc.edu/acknowledge.html>
7. DIAL (Data and Information Access Link)
<http://dial.gsfc.nasa.gov/>
8. XML Pointers, XML Base and XML Linking
<http://www.w3.org/XML/Linking>
9. XML Path Language
<http://www.w3.org/TR/xpath>