

CS 580 Client-Server Programming

Fall Semester, 2000

Doc 15 WWW & Client-Server

Contents

WWW as a Client-Server System	2
Special Web Servers	5
Special API for Web servers.....	6
Servlets	7
Common Gateway Interface (CGI)	10
Finding and running the CGI program.....	12
Sending a Response to the server.....	12
Information from the Web server to the CGI Program	24
CGI & Java.....	32

References

Apache Modules <http://modules.apache.org/>

Servlets <http://java.sun.com/products/servlet/2.2/javadoc/index.html>

Tomcat <http://jakarta.apache.org/>

JRun <http://www.allaire.com/products/jrun/index.cfm>

FastCGI <http://www.fastcgi.com/>

The Common Gateway Interface

<http://hoofoo.ncsa.uiuc.edu/cgi/overview.html>

Copyright ©, All rights reserved.

2000 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700
USA.

OpenContent (<http://www.opencontent.org/opl.shtml>) license defines the copyright on
this document.

WWW as a Client-Server System

Many people use WWW to implement client-server systems

Advantages:

- WWW browsers are everywhere
- People know how to use WWW
- No need to distribute updates to your client
- Provides links to your services

Disadvantages:

- Browsers have limited functionality
- http is stateless
- http is slow for some applications
- HTML is static

Generating HTML pages dynamically

WWW client-server systems need more interaction than static web pages

- Dynamically generate HTML pages
- Augment the functionality of browsers

Web server applications

- Code/programs used to dynamically generate HTML

Ways to Dynamically generate HTML

- Special Web servers
- Special API for Web servers
- Servlets
- CGI
- FastCGI
- Active Web pages

Special Web Servers

Basic HTTP is not hard to implement

Write your own server to produce dynamic responses

Not common

Examples

- Assignment 2
- Class Wiki

Special API for Web servers

Web servers allow you to link code to extend their functionality

Examples

- Netscape Server API (NSAPI)
- Internet Server API (ISAPI) - Microsoft
- Apache Modules <http://modules.apache.org/>
- Servlets

Pros:

- Fast response time

Cons

- Steep learning curve
- Proprietary
- Language dependence
- No process isolation

Servlets Background

Started by Sun

Sun controls the Servlet API

(<http://java.sun.com/products/servlet/2.2/javadoc/index.html>)

Tomcat - Apache module that supports servlets & Java Server pages

JRun - commercial product supporting servlets & Java Server pages

What are Servlets?

Java programs that are
Run as part of a web server
Mapped to URLs

Servlet API contains 40 classes & interfaces

Sample Servlet Program

```
import java.io.*;
import java.util.Date;
import javax.servlet.*;

public class SimpleDateServlet extends GenericServlet
{
    public void service(ServletRequest request,
        ServletResponse response)
        throws ServletException, IOException
    {
        InputStream rawIn;
        DataInputStream parsedInput;

        rawIn = request.getInputStream();
        parsedInput = new DataInputStream(rawIn);

        ServletOutputStream out = response.getOutputStream();

        String inputLine = parsedInput.readLine();

        if (inputLine.startsWith("date"))
        {
            Date now = new Date();
            out.println( now.toString() );
        }
        else
            out.println( "Invalid Input" );
    }
}
```

Servlet that Displays Form Data

```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class SnoopServlet extends HttpServlet
{
    public void doGet (HttpServletRequest request,
        HttpServletResponse response)
        throws ServletException, IOException
    {
        response.setContentType("text/html");

        ServletOutputStream out = response.getOutputStream();

        Enumeration names = request.getParameterNames();
        if (names.hasMoreElements())
        {
            out.println("<pre>");
            while (names.hasMoreElements())
            {
                String formItemName = (String)names.nextElement();
                String formItemValues[] =
                    (String []) request.getParameterValues(formItemName);
                if (formItemValues != null)
                {
                    out.print("<b> " + formItemName + " = </b>");
                    out.println(formItemValues[0]);
                }
                out.println("<p>");
            }
            out.println("</pre>");
        }
        out.println("</body></html>");
    }
}
```

Common Gateway Interface (CGI)

Standard for calling external programs

The basic idea

Web server receives a CGI request

Web server starts the requested CGI program

Web server passes request data to CGI program

CGI program processes request

CGI program return data to Web server

Web server returns response to web client

Pros

- Simplicity
- Language independence
- Process isolation

Cons

- Start up costs for CGI program per request

The basic questions

- How does the Web server find and run the CGI program?
- How do the Web server and CGI program communicate?
- What information can be sent from Web Server to the CGI program?
- What information can be sent from the CGI program to the Web Server?

Finding and running the CGI program

Web server is configured to treat files in particular directories as programs

Often cgi-bin indicates a such a CGI directory

All files in a CGI directory and subdirectories are treated as programs

A URL that references a program in a CGI directory will cause the Web server to run that program as a CGI program

Sending a Response to the server

CGI program writes the response on the standard output

The server reads the output and forms http response

CGI defines headers for communication from CGI program to server

The CGI Return Headers (CGI/1.1)

The output of scripts begins with a small header. This header consists of text lines, in the same format as an HTTP header, terminated by a blank line (a line with only a linefeed or CR/LF).

Any headers that are not server directives are sent directly back to the client. Currently, this specification defines three server directives:

Content-type

This is the MIME type of the document you are returning.

Location

Returns to the server a reference to a document

If is a URL, the server will issue a redirect to the client.

If is a virtual path, the server will retrieve the document specified

Status

An HTTP/1.0 status line for the server to send to the client.

CGI Examples of Sending Data to a Sever Content-type Example 1 Simple Text

File: hiMomTextPlain

Location: /net/www/www-eli/cgi-bin/cgiExamples

CGI directory: /net/www/www-eli/cgi-bin

File Contents:

```
#!/bin/sh
```

```
echo Content-type: text/plain  
echo
```

```
echo Hi Mom,  
echo  
echo  How is Dad?  
echo I am fine.
```

URL:

<http://www.eli.sdsu.edu/cgi-bin/cgiExamples/hiMomTextPlain>

Result in Netscape when open above URL:

Hi Mom,

How is Dad?

I am fine.

Content-type Example 1 Simple Text

Telnet Session:

```
/net/www/www-eli-> telnet www.eli.sdsu.edu 80
Trying 130.191.226.80...
Connected to www.eli.sdsu.edu.
Escape character is '^]'.
GET /cgi-bin/cgiExamples/hiMomTextPlain HTTP/1.0
```

```
HTTP/1.0 200 OK
Server: Netscape-Commerce/1.12
Date: Saturday, 29-Mar-97 05:04:17 GMT
Content-type: text/plain
```

Hi Mom,

How is Dad?

I am fine.

Connection closed by foreign host.

Content-type Example 2 HTML

File: hiMomTextHtml

Location: /net/www/www-eli/cgi-bin/cgiExamples

CGI directory: /net/www/www-eli/cgi-bin

File Contents:

```
#!/bin/sh
```

```
echo Content-type: text/html  
echo
```

```
echo Hi Mom,  
echo  
echo How is Dad?  
echo I am fine.
```

URL:

<http://www.eli.sdsu.edu/cgi-bin/cgiExamples/hiMomTextHtml>

Result in Netscape when open above URL:

Hi Mom, How is Dad? I am fine.

Content-type Example 3 Some Real HTML

File: hiMomFancy

Location: /net/www/www-eli/cgi-bin/cgiExamples

CGI directory: /net/www/www-eli/cgi-bin

File Contents:

```
#!/bin/sh
```

```
echo Content-type: text/html  
echo
```

```
echo "<B>Hi Mom</B>",  
echo  
echo  "<BIG>How is Dad?</BIG>"  
echo "<CENTER>I am fine</CENTER>".
```

URL:

<http://www.eli.sdsu.edu/cgi-bin/cgiExamples/hiMomFancy>

Redirection Example 1 URL

File: redirectURL

Location: /net/www/www-eli/cgi-bin/cgiExamples

CGI directory: /net/www/www-eli/cgi-bin

File Contents:

```
#!/bin/sh
```

```
echo Location: http://www.eli.sdsu.edu/index.html
```

```
echo
```

```
# All the stuff after this is does no good
```

```
# Client is working on the redirection
```

```
echo "<B>Hi Mom</B>",
```

```
echo
```

```
echo  "<BIG>How is Dad?</BIG>"
```

```
echo "<CENTER>I am fine</CENTER>".
```

URL:

<http://www.eli.sdsu.edu/cgi-bin/cgiExamples/redirectURL>

Redirection Example 1 URL

Telnet Session:

```
/net/www/www-eli-> telnet www.eli.sdsu.edu 80
Trying 130.191.226.80...
Connected to www.eli.sdsu.edu.
Escape character is '^]'.
GET /cgi-bin/cgiExamples/redirectURL HTTP/1.0
```

```
HTTP/1.0 302 Found
Server: Netscape-Commerce/1.12
Date: Saturday, 29-Mar-97 06:25:26 GMT
Location: http://www.eli.sdsu.edu/index.html
```

```
<B>Hi Mom</B>,
```

```
<BIG>How is Dad?</BIG>
<CENTER>I am fine</CENTER>.
Connection closed by foreign host.
```

Redirection Example 2 Path

File: redirectPath

Location: /net/www/www-eli/cgi-bin/cgiExamples

CGI directory: /net/www/www-eli/cgi-bin

File Contents:

```
#!/bin/sh
```

```
echo Location: /cgi-bin/cgiExamples/hiMomFancy  
echo
```

URL:

<http://www.eli.sdsu.edu/cgi-bin/cgiExamples/redirectPath>

Redirection Example 2 Path

Telnet Session:

```
/net/www/www-eli-> telnet www.eli.sdsu.edu 80
Trying 130.191.226.80...
Connected to www.eli.sdsu.edu.
Escape character is '^]'.
GET /cgi-bin/cgiExamples/redirectPath HTTP/1.0
```

```
HTTP/1.0 200 OK
Server: Netscape-Commerce/1.12
Date: Saturday, 29-Mar-97 06:38:56 GMT
Content-type: text/html
```

```
<B>Hi Mom</B>,
```

```
<BIG>How is Dad?</BIG>
<CENTER>I am fine</CENTER>.
Connection closed by foreign host.
```

Status Example 1

File: status

Location: /net/www/www-eli/cgi-bin/cgiExamples

CGI directory: /net/www/www-eli/cgi-bin

File Contents:

```
#!/bin/sh
```

```
echo Content-type: text/plain  
echo Status: 298 Okey Dokey with Me  
echo
```

```
echo Hi Mom,  
echo  
echo How is Dad?  
echo I am fine.
```

URL:

<http://www.eli.sdsu.edu/cgi-bin/cgiExamples/status>

Status Example 1

Telnet Session:

```
/net/www/www-eli -> telnet www.eli.sdsu.edu 80
```

```
Trying 130.191.226.80...
```

```
Connected to www.eli.sdsu.edu.
```

```
Escape character is '^]'.
```

```
GET /cgi-bin/cgiExamples/status HTTP/1.0
```

```
HTTP/1.0 298 Okey Dokey with Me
```

```
Server: Netscape-Commerce/1.12
```

```
Date: Saturday, 29-Mar-97 19:49:50 GMT
```

```
Content-type: text/plain
```

```
Hi Mom,
```

```
How is Dad?
```

```
I am fine.
```

```
Connection closed by foreign host.
```

Information from the Web server to the CGI Program

Web server sends data to the CGI Program via:

- Environment variables
- Command line
- Standard input

Environment Variables

AUTH_TYPE	CONTENT_LENGTH	CONTENT_TYPE
DOCUMENT_ROOT	GATEWAY_INTERFACE	HTTP_ACCEPT
HTTP_USER_AGENT	PATH_INFO	PATH_TRANSLATED
QUERY_STRING	REMOTE_ADDR	REMOTE_HOST
REMOTE_IDENT	REMOTE_USER	REQUEST_METHOD
SCRIPT_NAME	SERVER_NAME	SERVER_PORT
SERVER_PROTOCOL	SERVER_SOFTWARE	

Not all variables are set for each request

See <http://hoohoo.ncsa.uiuc.edu/cgi/env.html> for more information.

Example of CGI Environment Variables

File: variables

Location: /net/www/www-eli/cgi-bin/cgiExamples

URL:

<http://www.eli.sdsu.edu/cgi-bin/cgiExamples/variables>

File Contents:

```
#!/bin/sh
# Code from http://hoohoo.ncsa.uiuc.edu/cgi/test-cgi.txt
echo Content-type: text/plain
echo

echo CGI/1.0 test script report:
echo

echo argc is $#. argv is "$*".
echo

echo CONTENT_LENGTH = $CONTENT_LE
echo CONTENT_TYPE = $CONTENT_TYPE
echo GATEWAY_INTERFACE = $GATEWAY_INTERFACE
echo HTTP_ACCEPT = "$HTTP_ACCEPT"
echo PATH_INFO = $PATH_INFO
echo PATH_TRANSLATED = $PATH_TRANSLATED
echo QUERY_STRING = $QUERY_STRING
echo REMOTE_ADDR = $REMOTE_ADDR
echo REMOTE_HOST = $REMOTE_HOST
echo REMOTE_USER = $REMOTE_USER
echo REQUEST_METHOD = $REQUEST_METHOD
echo SCRIPT_NAME = $SCRIPT_NAME
echo SERVER_NAME = $SERVER_NAME
echo SERVER_PORT = $SERVER_PORT
echo SERVER_PROTOCOL = $SERVER_PROTOCOL
echo SERVER_SOFTWARE = $SERVER_SOFTWARE
```

Other Data sent to the CGI Program

The following http request methods can be used to send other data to the CGI program:

GET
POST
PUT
ISINDEX query

GET

GET sends data via the QUERY_STRING environment variable

GET data is sent in one of two ways:

Via a html form
By placing name-value pairs at the end of a URL

The data is encoded in x-www-form-urlencoded format when send via html form

A space is encoded as a '+'

All characters other than a-z, A-Z, 0-9, '_' and space are encoded as a % followed by two hex digits that make up the character.

GET Method Example 1. URL

URL:

<http://www.eli.sdsu.edu/cgi-bin/cgiExamples/variables?name=Roger&class=596>

Result in Netscape when open above URL:

(I removed some variables to save space)

CGI/1.0 test script report:

CONTENT_LENGTH =

CONTENT_TYPE =

QUERY_STRING = name=Roger&class=cs596

REQUEST_METHOD = GET

SCRIPT_NAME = /cgi-bin/cgiExamples/variables

SERVER_PORT = 80

SERVER_PROTOCOL = HTTP/1.0

SERVER_SOFTWARE = Netscape-Commerce/1.12

GET Method Example 2. URL bad form

URL:

<http://www.eli.sdsu.edu/cgi-bin/cgiExamples/variables?name<Roger;class@cs596>

Result in Netscape when open above URL:

CGI/1.0 test script report:

argc is 1. argv is name\<Roger\;class@cs596.

CONTENT_LENGTH =

CONTENT_TYPE =

GATEWAY_INTERFACE = CGI/1.1

HTTP_ACCEPT = image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*

PATH_INFO =

PATH_TRANSLATED =

QUERY_STRING = name<Roger;class@cs596

REMOTE_ADDR = 24.177.53.118

REMOTE_HOST =

REMOTE_USER =

REQUEST_METHOD = GET

SCRIPT_NAME = /cgi-bin/cgiExamples/variables

SERVER_NAME = www.eli.sdsu.edu

SERVER_PORT = 80

SERVER_PROTOCOL = HTTP/1.0

SERVER_SOFTWARE = Apache/1.3.9 (Unix) PHP/3.0.12

GET Method Example 3. A Form

The Form, HTML

```
<P>
<FORM ACTION=
"http://www.eli.sdsu.edu/cgi-bin/cgiExamples/variables"
METHOD="GET">

<INPUT TYPE="checkbox" NAME="top" VALUE="on">
<INPUT TYPE="checkbox" NAME="left" VALUE="on">
<INPUT TYPE="submit" VALUE="Submit"> <P>

</FORM>
```

The Form, Rendered

```
<FORM METHOD="GET" ACTION="http://www.eli.sdsu.edu/cgi-
bin/cgiExamples/variables">
<INPUT TYPE="CHECKBOX" NAME="top" VALUE="on">
<INPUT TYPE="CHECKBOX" NAME="left" VALUE="on">
<INPUT TYPE="SUBMIT" VALUE="Submit">
</FORM>
```

Result in Netscape upon submit, both boxes checked:

CGI/1.0 test script report:

argc is 0. argv is .

```
CONTENT_LENGTH =
GATEWAY_INTERFACE = CGI/1.1
REQUEST_METHOD = GET
QUERY_STRING = top=on&left=on
```

POST and PUT

When data is sent via POST or PUT,

the CONTENT_LENGTH environment variable contains the number of bytes of data sent to the CGI program via the standard input

POST Method Example 1. A Form

The Form, HTML

```
<P>
<FORM ACTION=
"http://www.eli.sdsu.edu/cgi-bin/cgiExamples/variables"
METHOD="POST">

<INPUT TYPE="checkbox" NAME="top" VALUE="on">
<INPUT TYPE="checkbox" NAME="left" VALUE="on">
<INPUT TYPE="submit" VALUE="Submit"> <P>
</FORM>
```

The Form, Rendered

```
<FORM METHOD="POST"
ACTION="http://www.eli.sdsu.edu/cgi-
bin/cgiExamples/variables">
<INPUT TYPE="CHECKBOX" NAME="top" VALUE="on">
<INPUT TYPE="CHECKBOX" NAME="left" VALUE="on">
<INPUT TYPE="SUBMIT" VALUE="Submit">
</FORM>
```

Result in Netscape upon submit, both boxes checked:

CGI/1.0 test script report:

argc is 0. argv is .

CONTENT_LENGTH =

CONTENT_TYPE = application/x-www-form-urlencoded

GATEWAY_INTERFACE = CGI/1.1

HTTP_ACCEPT = image/gif, image/x-xbitmap, image/jpeg,
image/pjpeg, */*

PATH_INFO =

PATH_TRANSLATED =

QUERY_STRING =

REMOTE_ADDR = 130.191.4.123

REMOTE_HOST = ebbsrv1p22.sdsu.edu

REMOTE_USER =

REQUEST_METHOD = POST

SCRIPT_NAME = /cgi-bin/cgiExamples/variables

SERVER_NAME = www.eli.sdsu.edu

SERVER_PORT = 80

SERVER_PROTOCOL = HTTP/1.0

SERVER_SOFTWARE = Netscape-Commerce/1.12

CGI & Java

Problems using Java program as CGI

- Java can not read environment variables

CGI requires access to environment variables

- Java programs are started with a two word command:

"java ClassName"

All methods of calling CGI programs (in Unix) only allow a one-word command to start the program

- Java uses the CLASSPATH environment variable to find classes

The web server, not you, runs your CGI programs.

Your CGI programs do not use your environment

Problems with Java & CGI on rohan

- CGI programs on rohan do not have write permission on rohan

Solution to Java & CGI problems

Use a wrapper program

A sample CGI Java program

```
class HiMom {  
    public static void main(String args[]) {  
        System.out.print("Content-type: text/plain\n\n");  
        System.out.println("Hi Mom");  
    }  
}
```

A simple wrapper for Hi Mom

```
#!/bin/sh
```

```
JAVA="/opt/java/bin/java"  
$JAVA HiMom
```

Make the URL request refer to the wrapper program

See <http://www-rohan.sdsu.edu/faculty/vinge/courses/spring00/cs580/notes/cgiJava/cgiJava.html> for a complete solution