Java CGI HOWTO

Table of Contents

Table of Contents

Java CGI HOWTO

by David H. Silber `javacgi-document@orbits.com`

v0.5, 1 December 1998

This HOWTO document explains how to set up your server to allow CGI programs written in Java and how to use Java to write CGI programs. Although HOWTO documents are targetted towards use with the Linux operating system, this particular one is not dependant on the particular version of unix used.

5.Using the Java CGI Classes.

6.Future Plans

7.Changes

1.Introduction

Because of the way that Java is designed the programmer does not have easy access to the system's environment variables. Because of the way that the Java Development Kit (JDK) is set up, it is necessary to use multiple tokens to invoke a program, which does not mesh very well with the standard HTML forms/CGI manner of operations. There are ways around these limitations, and I have implemented one of them. Read further for details.

Since I wrote the previous paragraph in 1996, there have been many changes in the Java technology. It is likely that a better solution to running server−side Java programs is now available −− perhaps you should take a look at servlets.

1.1 Prior Knowledge

I am assuming that you have a general knowledge of HTML and CGI concepts and at least a minimal knowledge of your HTTP server. You should also know how to program in Java, or a lot of this will not make sense.

1.2 This Document

The latest version of this document can be read at http://www.orbits.com/software/Java_CGI.html.

1.3 The Package

The latest version of the package described here can be accessed via anonymous FTP at ftp://ftp.orbits.com/pub/software/java_cgi−0.5.tgz. The package distribution includes SGML source for this document.

The package is distributed under the terms of the GNU Library General Public License. This document can be distributed under the terms of the Linux HOWTO copyright notice.

If you use this software, please make some reference to http://www.orbits.com/software/Java_CGI.html, so that others will be able to find the Java CGI classes.

I have run out of time to maintain and support this package, so this will probably be its final release. If anyone out there is sufficiently enamoured of this software that they wish to take over the maintenace of it, please contact me at `javacgi-document@orbits.com`.

1.4 The Mailing List

I have created a majordomo list to allow people to help each−other work through their mutual problems in installing and using this software. Send a message to `javacgi−request@orbits.com`, containing the word *subscribe*.

2.Setting Up Your Server to Run Java CGI Programs (With Explanations)

This section will lead you through installing my *Java CGI* package with copious explanations so that you know what the effects of your actions will be. If you just want to install the programs and don't care about the whys & wherefores, skip to Setting Up Your Server to Run Java CGI Programs (The Short Form).

2.1 System Requirements

This software should work on any unix−like web server that has the Java Development Kit installed. I am using it on a *Debian Linux* system running *apache* as the HTTP daemon. If you find that it does not run on your server, please contact the mailing list. See The Mailing List for details.

Unfortunatly, the Java run−time interpreter seems to be something of a memory hog −− you may want to throw another few megabytes of RAM onto your server if you will be using Java CGI programs a lot.

2.2 Java CGI Add−On Software

The software that I wrote to aid in this is called *Java CGI*. You can get it from ftp://ftp.orbits.com/pub/software/java_cgi−0.5.tgz. (The version number may have changed.)

2.3 Unpacking the Source

Find a convenient directory to unpack this package into. (If you don't already have a standard place to put packages, I suggest that you use `/usr/local/src`.) Unpack the distribution with this command:

```
gzip −dc java_cgi−0.5.tgz | tar −xvf −
```

This will create a directory called `java_cgi−0.5`. In there you will find the files referenced in the rest of this document. (If the version number has changed, use the instructions from within that distribution from this point on.)

2.4 Decide On Your Local Path Policies

You need to decide where you want your Java CGI programs to live. Generally, you will want to put them in a directory in parallel with your `cgi−bin` directory. My *apache* server came configured to use `/var/www/cgi−bin` as the `cgi−bin` directory, so I use `/var/www/javacgi` as the directory to put Java CGI programs in. You probably do not want to put your Java CGI programs into one of the existing `CLASSPATH` directories. Edit the Makefile to reflect your system configuration. Make sure that you are logged in as the root user and run `make install`. This will compile the Java programs, modify the `java.cgi` script to fit in with your system and install the programs in the appropriate places. If you want the HTML version of this documentation and an HTML test document in addition, run `make all` instead.

2.5 Testing your installation.

Installed from the distribution are HTML documents called `javacgitest.html`, `javaemailtest.html` and `javahtmltest.html`. If you installed `all` in the previous section, it will be in the directory you specified for WEBDIR in the `Makefile`. If you didn't, you can run `make test` to build them from `javacgitest.html-dist`, `javaemailtest.html-dist` and `javahtmltest.html-dist`.

When you are sure that your installation is working correctly, you may wish to remove `CGI_Test.class`, `Email_Test.class` and `HTML_Test.class` from your JAVACGI directory and `javacgitest.html`, `javaemailtest.html` and `javahtmltest.html` from your WEBDIR directory as they show the user information that is normally only available to the server.

3.Setting Up Your Server to Run Java CGI Programs (The Short Form)

- Get the *Java CGI* package from ftp://ftp.orbits.com/pub/software/java_cgi−0.5.tgz. (The version number may have changed.)
- Unpack the distribution with this command:

  ```
  gzip -dc java_cgi-0.5.tgz | tar -xvf -
  ```
  (If the version number has changed, use the instructions from within that distribution from this point on.)
- Edit the `Makefile` you will find in the newly created directory `java_cgi-0.5` as appropriate to your system.
- As root, run `make install`. This will compile the Java programs, apply your system−specific information and install the various files. If you want the HTML version of this documentation and an HTML test document, run `make all` instead.
- You should be ready to go.

4.Executing a Java CGI Program

4.1 Obstacles to Running Java Programs Under the CGI Model

There are two main problems in running a Java program from a web server:

You can't run Java programs like ordinary executables.

You need to run the Java run−time interpreter and provide the initial class (program to run) on the command−line. With an HTML form, there is no provision for sending a command−line to the web server.

Java does not have general access to the environment.

Every environment variable that will be needed by the Java program must be explicitly passed in. There is no method similar to the **C**`getenv()` function.

4.2 Overcoming Problems in Running Java CGI Programs

To deal with these obstacles, I wrote a shell CGI program that provides the information needed by the Java interpreter.

The java.cgi script.

This shell script manages the interaction between the HTTP daemon and the Java CGI program that you wish to use. It extracts the name of the program that you want to run from the server−provided data. It collects all of the environment data into a temporary file. Then, it runs the Java run−time interpreter with the name of the file of environment information and the program name added to the command−line.

The `java.cgi` script was configured and installed in Decide On Your Local Path Policies.

Invoking java.cgi from an HTML form.

My forms that use Java CGI programs specify a form action as follows:

```
<form action="/cgi-bin/java.cgi/CGI_Test" method="POST">
```

Where `/cgi-bin/` is your local CGI binary directory, `java.cgi` is the Java front−end that allows us to run Java programs over the web and `CGI_Test` is an example of the name of the Java program to run.

5.Using the Java CGI Classes.

There are currently three main classes supported −− CGI, Email and HTML. I am considering adding classes to deal with MIME−formatted input and output −− MIMEin & MIMEout, respectively.

There are also a few support and test classes. CGI_Test, Email_Test and HTML_Test are intended to be used to test your installation. They can also be used as a starting−point for your own Java programs which use this class library. The Text class is the superclass for both the `Email` and the `HTML` classes.

5.1 CGI

Class Syntax

```
public class CGI
```

Class Description

The CGI class holds the ``CGI Information'' −− Environment variables set by the web server and the name/value sent from a form when its **submit** action is selected. All information is stored in a `Properties` class object.

This class is in the ``Orbits.net'' package.

Member Summary

```
CGI()          //  Constructor.
getNames()     //  Get the list of names.
getValue()     //  Get form value by specifying name.
```

See Also

CGI_Test.

CGI()

Purpose

Constructs an object which contains the available CGI data.

Syntax

`public CGI()`

Description

When a CGI object is constructed, all available CGI information is sucked−up into storage local to the new object.

getNames()

Purpose

List the names which are defined to have corresponding values.

Syntax

`public Enumeration getKeys ()`

Description

Provides the full list of names for which coresponding values are defined.

Returns

An `Enumeration` of all the names defined.

getValue()

Purpose

Retrieves the **value** associated with the **name** specified.

Syntax

```
public String getValue ( String name )
```

Description

This method provides the corespondence between the `names` and `values` sent from an HTML form.

Parameter

name

The key by which values are selected.

Returns

A `String` containing the value.

5.2 CGI_Test

This class provides both an example of how to use the `CGI` class and a test program which can be used to confirm that the *Java CGI* package is functioning correctly.

Member Summary

```
main()      //  Program main().
```

See Also

`CGI`.

main()

Purpose

Provide a `main()` method.

Syntax

```
public static void main( String argv[] )
```

Description

This is the entry point for a CGI program which does nothing but return a list of the available name/value pairs and their current values.

Parameter

argv[]

Arguments passed to the program by the `java.cgi` script. Currently unused.

5.3 Email

Class Syntax

`public class Email extends Text`

Class Description

Messages are built up with the `Text` class `add*()` methods and the e−mail−specific methods added by this class. When complete, the message is sent to its destination.
This class is in the ``Orbits.net'' package.

Member Summary

```
Email()     //  Constructor.
send()      //  Send the e-mail message.
sendTo()    //  Add a destination for message.
subject()   //  Set the Subject: for message.
```

See Also

`Email_Test`, `Text`.

Email()

Purpose

Constructs an object which will contain an email message.

Syntax

```
public Email()
```

Description

Sets up an empty message to be completed by the Email methods.

See Also

`Text`.

send()

Purpose

Send the e−mail message.

Syntax

```
public void send ()
```

Description

This formats and sends the message. If no destination address has been set, there is no action taken.

sendTo()

Purpose

Add a destination for this message.

Syntax

```
public String sendTo ( String address )
```

Description

Add `address` to the list of destinations for this method. There is no set limit to the number of destinations an e−mail message may have. I'm sure that if you build up the list large enough, you can exceed the size of the parameter list that the *Mail Transport Agent* can accept or use up your memory.

Parameter/

address

A destination to send this message to.

subject()

Purpose

Set the subject for this message.

Syntax

```
public void subject ( String subject )
```

Description

This method sets the text for the e−mail's `Subject:` line. If called more than once, the latest subject set is the one that is used.

Parameter

subject

The text of this message's `Subject:` line.

5.4 Email_Test

This class provides both an example of how to use the `Email` class and a test program which can be used to confirm that the *Java CGI* package is functioning correctly.

Member Summary

```
main()      //  Program main().
```

See Also

`Email`.

main()

Purpose

Provide a `main()` method.

Syntax

```
public static void main( String argv[] )
```

Description

This is the entry point for a CGI program which returns a list of the available name/value pairs and their current values. It will also send this list to the address specified in the `Email` variable.

Parameter

argv[]

Arguments passed to the program by the `java.cgi` script. Currently unused.

5.5 HTML

Class Syntax

```
public class HTML extends Text
```

Class Description

Messages are built up with the `Text` class `add*()` methods and the HTML−specific methods added by this class. When complete, the message is sent to its destination.
Currently, there is no error checking to confirm that the list−building methods are being used in a correct order, so the programmer must take pains not to violate HTML syntax.
This class is in the ``Orbits.net'' package.

Member Summary

```
HTML()                //  Constructor.
author()              //  Set the name of the document author.
definitionList()      //  Start a definition list.
definitionListTerm()  //  Add a term to a definition list.
endList()             //  End a list.
listItem()            //  Add an entry to a list.
send()                //  Send the HTML message.
title()               //  Set the text for the document title.
```

See Also

`HTML_Test, Text.`

HTML()

Purpose

Constructs an object which will contain an HTML message.

Syntax

```
public HTML()
```

Description

Sets up an empty message to be completed by the HTML methods.

See Also

`Text`.

author()

Purpose

Set the name of the document author.

Syntax

```
public void author ( String author )
```

Description

Set the name of the document author to `author`.

Parameter/

author

The text to use as the author of this message.

See Also

`title()`.

definitionList()

Purpose

Start a definition list.

Syntax

```
public void definitionList ()
```

Description

Start a definition list. A *definition list* is a list specialized so that each entry in the list is a *term* followed by the definition *text* for that term. The start of a definition list should be followed by the creation of (at least) one term/text pair and a call to the `endList()` method. *Note that,* *currently, lists cannot be nested.*

See Also

`definitionListTerm()`, `endList()`, `listItem()`.

definitionListTerm()

Purpose

Add a term to a definition list.

Syntax

```
public void definitionListTerm ()
```

Description

Add a term to a definition list. The text for the term part of the current list entry should be appended to the message after this method is called and before a corresponding `listItem` method is called.

See Also

`definitionList()`, `listItem()`.

endList()

Purpose

End a list.

Syntax

```
public void endList ()
```

Description

End a list. This method closes out a list. *Note that, currently, lists cannot be nested.*

See Also

`definitionList()`.

listItem()

Purpose

Add an entry to a list.

Syntax

```
public void listItem ()
```

```
public void listItem ( String item )
```

```
public boolean listItem ( String term, String item )
```

Description

Add an entry to a list. If the first form is used, the text for the current list item should be appended to the message after this method is called and before any other list methods are called. In the second and third forms, the `item` text is specified as a parameter to the method instead of (or in addition to) being appended to the message. The third form is specific to definition lists and provides both the term and the definition of the list entry.

Parameters

item

The text of this list entry.

term

The text of this definition list entry's term part.

See Also

`definitionList()`, `definitionListTerm()`, `endList()`.

send()

Purpose

Send the HTML message.

Syntax

```
public void send ()
```

Description

Send the HTML message.

title()

Purpose

Set the text for the document title.

Syntax

```
public void title ( String title )
```

Description

Set the text for the document title.

Parameter

title

The text of this message's title.

See Also

`author()`.

5.6 HTML_Test

This class provides both an example of how to use the `HTML` class and a test program which can be used to confirm that the *Java CGI* package is functioning correctly.

Member Summary

```
main()      //  Program main().
```

See Also

`HTML`.

main()

Purpose

Provide a `main()` method.

Syntax

```
public static void main( String argv[] )
```

Description

This is the entry point for a CGI program which returns a list of the available name/value pairs in an HTML document, with each name/value pair displayed in a definition list element.

Parameter

argv[]

Arguments passed to the program by the `java.cgi` script. Currently unused.

5.7 Text

Class Syntax

```
public abstract class Text
```

Class Description

This class is the superclass of the `Email` and `HTML` classes. Messages are built up with the methods in this class and completed and formatted with the methods in subclasses.

This class is in the ``Orbits.text'' package.

Member Summary

```
Text()          //  Constructor.        add()           //  Add text to this object.        addLineBreak()   //  Add a line break.        addParagraph()   //  Add a paragraph break.
```

See Also

`Email`, `HTML`.

add()

Purpose

Add text to this item.

Syntax

```
public void add ( char addition )
```

```
public void add ( String addition )
```

```
public void add ( StringBuffer addition )
```

Description

Add `addition` to the contents of this text item.

Parameter

addition

Text to be added to the text item.

See Also

`addLineBreak()`, `addParagraph()`.

addLineBreak()

Purpose

Force a line break at this point in the text.

Syntax

```
public void addLineBreak ()
```

Description

Add a line break to the text at the current point.

See Also

`add(), addParagraph().`

addParagraph()

Purpose

Start a new paragaph.

Syntax

```
public void add ()
```

Description

Start a new paragraph at this point in the text flow.

See Also

`add(), addLineBreak().`

6.Future Plans

Add to the Email class:

Email(int capacity)

Used when we know how much space the message will need to have allocated.

sendTo(String [] address)

Add a list of primary destinations to the e−mail message.

sendCc(String address)

Add a Carbon−Copy destination to the e−mail message.

sendCc(String [] address)

Add a list of Carbon−Copy destinations to the e−mail message.

sendBcc(String address)

Add a Blind Carbon−Copy destination to the e−mail message.

sendBcc(String [] address)

Add a list of Blind Carbon−Copy destinations to the e−mail message.
Add to the HTML class:

HTML(int capacity)

Used when we know how much space the message will need to have allocated.

public void unorderedList()

Start an unordered list.

public void orderedList()

Start an ordered list.

public void directoryList()

Start a directory list.

public void menuList()

Start a menu list.

void anchor(String anchorName)

Specify an anchor.

void link(String url, String text)

Specify a link.

void applet(String url, String altText)

Specify an applet link.
Allow HTML lists to be nested.
Add error checking code to enforce correct ordering of HTML list formatting codes.
The location of the file of environment data should be configurable from the `Makefile`.
Get rid of the spurious empty name/value pair that appears in the list when we are dealing with the GET method of data transfer.
Consider having CGI implement the java.util.Enumeration interface to successively provide variable names.
Add a `Test` class, which would use every method in this package.
Document how `CGI_Test`, `Email_Test` and `HTML_Test` build on each other to provide incremental tests for debugging purposes.
Document how Test uses every feature available in this package.

7.Changes

7.1 Changes from 0.4 to 0.5

Changed documentation and comments to reflect the final nature of this release.

7.2 Changes from 0.3 to 0.4

Fleshed out the HTML class to provide minimal functionality.
Wrote the HTML_Test class and javahtmltest.html−dist.
Added the HTML methods to deal with a definition list.

7.3 Changes from 0.2 to 0.3

Added the Text and Email classes. HTML was also added, but it is merely a stub at this point.
Put the various classes into packages. The main classes are in `Orbits.net.*`, the support class `Text` is in `Orbits.text.Text`.

Changed `CGItest` to `CGI_Test`.
Added the `Email_Test` class.

7.4 Changes from 0.1 to 0.2

The environment variables are put into a temportary file instead of being crammed into the Java inperpreter command−line. The `CGI` class and `java.cgi` had to be modified.
The `javacgitest.html` document is made part of the distribution.
The text files which are modified by `make` upon installation are provided with names that end with *−dist*.