



A programming language for novice users to easily generate HTML forms

LANGUAGE REFERENCE MANUAL

Vinay Ahuja
va2199@columbia.edu

COMS W4115 – Programming Languages and Translators

March 10, 2009

Table of Contents

1. Introduction	3
2. Execution	3
3. Lexical Conventions	3
3.1 ASCII	3
3.2 Whitespace	3
3.3 Identifiers	4
3.4 Keywords	4
3.5 Types	4
3.5.1 Number	4
3.5.2 Boolean	5
3.5.3 String	5
3.5.4 create	5
3.5.5 page	5
3.5.6 form	5
3.5.7 set	5
3.5.8 title	6
3.5.9 method	6
3.5.10 action	6
3.5.11 submit	6
3.5.12 reset	6
3.5.13 textbox	6
3.5.14 label	6
3.5.15 name	7
3.5.16 size	7
3.5.17 if then else	7

1. Introduction

The Bridge programming language provides an ability for novice users to develop web pages without necessarily being familiar with any HTML syntax. As implied by the name, the language provides novice users a *bridge* to web based applications. It does so by providing users an intuitive and easy to learn syntax which is simple enough to read and understand, but at the same time powerful enough to create complex HTML pages. This document formally defines the Bridge programming language, its syntax and semantics so that users and developers can understand the basic elements and start to write programs in this language.

2. Execution

A file representing a Bridge program has a `.br` extension and each such file represents a complete program. The Bridge language interpreter scans this file in the lexical analysis stage, which results in tokens explained later in this document. The tokens are then parsed to verify if the input is a valid, syntactically correct program. This stage is followed by a semantic analysis and the generation of a `.html` file as an output which contains HTML code generated for the input program. The HTML file can then be viewed in any browser.

3. Lexical Conventions

A Bridge program is comprised of a single file written in ASCII character set. Following the convention of other common languages – comments, tabs and newlines as defined below are ignored. Whitespace is required to separate tokens and is used by the parser to identify the beginning and the end of a token. At the lexical analysis stage a Bridge program can be thought of as a single line program containing the tokens described below separated by whitespace.

3.1 ASCII

Includes all of the 128 characters defined in the USASCII code chart. Every single element of a Bridge program is made up of some combination of these characters which the parser determines to be a part of the language.

3.2 Whitespace

Whitespace is used to identify one token apart from another.

3.3 Identifiers

An identifier is defined as a combination of alphanumeric characters [a-z] [A-Z] [0-9] and must start with a lower case alphabet [a-z].

3.4 Keywords

The following identifiers are reserved keywords and therefore may not be used in any other context:

```
Number
Boolean
String
create
set
textbox
label
name
size
page
form
method
action
title
submit
reset
if then else
```

3.5 Types

3.5.1 Number

A Number is used to define an integer value [0-9].

Example Usage:

```
Number x = 10;
```

3.5.2 Boolean

A **Boolean** holds a value of `true` or `false`.

Example Usage:

```
Boolean b = true;
if b == true
    then { ... }
else { ... }
```

3.5.3 String

A **String** is any combination of ASCII characters `[a-z]` `[A-Z]` `[0-9]` `[.]` and may use whitespace to separate words just like in regular English language. A String must start with an alphabet `[a-z]` `[A-Z]`.

Example Usage:

```
String s = "Hello World."
```

3.5.4 create

The `create` keyword is used to create either a regular html page or a html form and their properties.

Example Usage:

```
create page { ... }
create form { ... }
```

3.5.5 page

The `page` keyword is used in conjunction with the `create` keyword for creating a regular HTML page with no form.

Example Usage:

```
create page { ... }
```

3.5.6 form

The `form` keyword is used in conjunction with the `create` keyword for creating a HTML page which contains a form.

Example Usage:

```
create form { ... }
```

3.5.7 set

The `set` keyword is used to set a property of HTML page or form, where a property may be a title, label, value, name, size, rows, cols, method or an action.

Example Usage:

```
set method = "POST";
```

3.5.8 title

The `title` keyword is used to set a page or form's title and is used in conjunction with the `set` keyword.

Example Usage:

```
set title = "Bridge Demo";
```

3.5.9 method

The `method` keyword is used to set a html form's method of sending information to the server and taken in either a "POST" or "GET" values.

Example Usage:

```
set method = "POST";
```

3.5.10 action

The `action` keyword is used to set a html form's action, the target that receives the information entered by a user in the html form.

Example Usage:

```
set action = "/webServiceMethod";
```

3.5.11 submit

The `action` keyword is used to create a submit button for a html form.

Example Usage:

```
create submit;
```

3.5.12 reset

The `action` keyword is used to create a reset button for a html form.

Example Usage:

```
create reset;
```

3.5.13 textbox

The `text` keyword is used to create a HTML text box and must be used inside a HTML page or form.

Example Usage:

```
create page  
{ create textbox }
```

3.5.14 label

The `label` keyword is used to display the text associated a `textbox`.

3.5.15 name

The `name` keyword is used to provide a name to a an input field, like a textbox on the html form and is used in conjunction with the `set` keyword.

Example Usage:

```
set name = "in1";
```

3.5.16 size

The `size` keyword is used to set the size of an input field, like a textbox on the html form and is used in conjunction with the `set` keyword.

Example Usage:

```
set size = 20;
```

3.5.17 if then else

The `if ([expr]) then {...} else {...}` keywords are used to execute code based on simple arithmetic expressions.

Example Usage:

```
if (x == 3)
    then { ... }
else {...}
```

3.6 Punctuations

Bridge programs may include the following punctuations:

```
{
}
;
(
)
=
==
+
*
-
/
"
"
//
```

3.7 Sample Programs

The following are some sample programs written in Bridge

3.7.1 Hello World

```
create page {
  set title = "Hello World";
  set label = "This is a sample Hello World program";
}
```

3.7.2 User Info Form

```
create form {
  set title = "User Info Form";
  set method = "POST";
  set action = "/userInfoWebService";
  create textbox {
    set label = "First Name ";
    set size = 20;
    set name = "fName";
  }
  create textbox {
    set label = "Last Name ";
    set size = 25;
    set name = "lName";
  }
  create submit {
    set label = "Submit";
    set name = "button1";
  }
  create reset {
    set label = "Reset";
    set name = "button2";
  }
}
```