

## Processing User Character Inputs Having Whitespace

### Field of the Invention

5 The present invention relates to user interface software and, more particularly, to a method, computer program product and computer system for processing whitespace in user character inputs.

### Background

10 In today's global economy, computer software is designed, developed and marketed to meet the demands of an international customer base. Software solutions (whether client-server or web-based) must be capable of providing a consistent and functional user interface across a wide range of languages.

However, the production of internationalized software is not without its challenges. For example, users, in general, enter information in different and inconsistent ways.  
15 Developing a 'write once, run everywhere' solution that processes multi-lingual information in a similar way, regardless of language or platform has certain disadvantages.

For example, in a conventional web-based environment, a web-form may be used to  
20 implement a "Security question – Security answer" scenario. The web-form is used to prompt a user to submit information at one time and to then prompt that user to re-submit the same information at a later time as a means of identification.

Referring to **Fig. 1**, at step **100**, a user wishes to register for a service of an organization  
25 and is presented with a web-form and asked to enter certain information. At step **110** and as part of this registration information, the user enters an information pair known as a "Security question – Security answer" set. For example, the security question could be: "What are the names of your cats?". The security answer might be: "Kuching and Mao Mao".

30 At step **120**, this information pair is stored in a database for later identity verification in case the user loses a password.

At step 130, and at some point in time after registration, the user may mislay a password and consequently request a new one. At step 140, a short web-form is presented with the security question from the database. The user is prompted to enter the answer supplied previously at step 110. At step 150, the software detects that the entered answer is  
5 semantically correct, and the characters are in the same order, but the nature of the whitespace surrounding the glyphs (that is, the visual characters) is different. This typically occurs because the user is careless with the character input and adds or omits whitespace characters before and/or after the glyphs. Accordingly, at step 160, the software incorrectly refuses the request for a new password. Thus the result may be an  
10 irritated user and a potential loss of business for the organization.

It is known that functionality exists in certain programming languages to trim leading and trailing whitespace characters from a string of characters. For example, the "trim" function in the Java™ (Sun Microsystems) programming language, which can be found at  
15 [http://java.sun.com/j2se/1.4.2/docs/api/java/lang/String.html#trim\(\)](http://java.sun.com/j2se/1.4.2/docs/api/java/lang/String.html#trim()) is such a function. However, this function is restricted to the American Standard Code for Information Interchange (ASCII) 7-bit character set and thus other defined whitespace characters in other character encodings, for other languages, are not considered.

20 Therefore, a need exists to provide processing of whitespace in multi-lingual user input that overcomes or substantially ameliorates one or more disadvantages and shortcomings of existing arrangements.

### **Summary**

Accordingly, in an aspect of the present invention there is provided a method for  
25 processing a user character input. The method comprises the steps of converting a user input character string to a Universal Character Set (UCS) string, such as a UNICODE character string; removing any leading and any trailing whitespace characters in the UCS string; and converting all remaining whitespace characters to a single value.

30 In another aspect of the present invention, there is provided a computer program product comprising a computer readable medium having a computer program recorded therein for processing a user character input. The computer program includes code means for converting a user input character string to a UCS string, such as a UNICODE character

string; code means for removing any leading and any trailing whitespace characters in the UCS string and code means for converting all remaining whitespace characters to a single value.

- 5 In another aspect of the present invention, there is provided a computer system for processing a user character input. The system comprises a memory unit for storing data and instructions to be performed by a processing unit; and a processing unit coupled to the memory unit. The processing unit is programmed to convert a user input character string to a UCS string, such as a UNICODE character string; remove any leading and any  
10 trailing whitespace characters in the UCS string; and convert all remaining whitespace characters to a single value.

It is preferable that the single value is hexadecimal 20. It can also be determined whether a converted UCS string has been previously stored in memory and, if so, the converted  
15 UCS string is compared with the previously stored converted UCS string. If the converted UCS string has not been previously stored, the converted UCS string is stored in memory. The computer program can be written in the Java language.

Embodiments of the invention provide fault tolerance in terms of users using variable  
20 numbers of whitespace characters at different instances of time. Additionally, the use of UCS and a single value replacement of whitespace character allows multiple users using different language sets to be accommodated at the same time.

### **Brief Description of the Drawings**

Aspects of the present invention will now be described, by way of example only, with  
25 reference to the accompanying drawings, in which:

**Fig. 1** is a flowchart representation of a conventional "Security question – Security answer" scenario.

30 **Fig. 2** is a flowchart representation of a "Security question – Security answer" scenario according to an embodiment of the present invention.

**Fig. 3** is a schematic representation of the known UNICODE whitespace code-points showing the subset of space characters.

**Fig. 4** is a schematic representation of a trimmed UNICODE character string according to  
5 an embodiment of the present invention.

**Fig. 5** is a schematic representation of a character string containing ASCII space (0x20) according to an embodiment of the present invention.

10 **Fig. 6** is a schematic representation of a computer system on which an embodiment of the invention can operate as computer software.

### **Detailed Description**

Where reference is made in any one or more of the accompanying drawings to steps and/or features, which have the same reference numerals, those steps and/or features have  
15 for the purposes of this description the same function(s) or operation(s), unless the contrary intention appears.

**Figs 2 to 5** show a method, computer program product and computer system for fault-tolerant pre-processing of whitespace in multi-lingual user input according to aspects of  
20 the present invention.

### *Overview*

**Fig. 2** shows a flow chart of a “Security question – Security answer” scenario according to an aspect of the present invention. At step **200**, a software application (for example a  
25 web browser) prompts the user for a “security question” and “security answer” pair. Accordingly, the user enters the information pair. The character strings as entered by the user will conform to the native character set and encoding of the browser for the language as selected by the user. For example, the encoding could be “Windows simplified Chinese”.

30

At step **210**, a character string (for example the “security answer” character string) is converted to a Universal Character Set (UCS), for example UNICODE. At step **220**, all whitespace characters before the first non-whitespace character and all whitespace

characters after the last non-whitespace character are trimmed from the character string. At step 230, all whitespace characters (including spaces) between words/diagram characters are replaced with a single instance of the ASCII space character 0x20 (hexadecimal 20). This space character is also known as a displayable placeholder. At  
5 step 240, the character string is stored in the database or is compared with the corresponding value previously stored in the database.

#### *Step 210 - Conversion to UCS*

Referring again to **Fig. 2**, at step 210, the user input character string is converted to a  
10 UCS string, such as a UNICODE character string. UNICODE (see [www.unicode.org](http://www.unicode.org)) enables a single software product to be deployed across multiple platforms, languages and countries without re-engineering. UNICODE does this by providing a unique number (called a code-point) for every character in all known character sets, regardless of platform, computer program or language.

15

Thus characters that could be one of many numbers in other encodings which map to one character, say a double dash with space, are entered as a single known UNICODE code-point that represents it. Accordingly, the characters are processed in a consistent manner.

20 Referring now to **Table 1**, and assuming that the byte array contains simplified Chinese GBK (Guo Biao Kuozhan) -encoded characters, a Java™ code function which performs the conversion to UNICODE is shown.

```
// A byte array of non-UNICODE characters
byte[] gbk_bytes = getGBKBytes();

//convert to a String of Java 16-bit UNICODE characters
String unicodeStr = new String(gbk_bytes, "GBK");
```

**Table 1 – Java™ code for conversion to UNICODE**

25

#### *Step 220 – Trim leading and trailing whitespace*

Referring again to **Fig. 2**, at step 220, any leading and any trailing whitespace characters are trimmed from the character string. In the context of this specification, the term

“trimming” (or removal) of any leading and any trailing whitespace is understood to include the range of possibilities for removal of whitespace characters from a character string. Accordingly, trimming is understood to include removal of any leading whitespace characters (if such characters exist); or removal of any trailing whitespace characters (if such characters exist); or removal of any leading and any trailing whitespace characters (if such characters exist in the character string). There also can be instances where no leading or trailing whitespace characters exist, and the trimming process is still performed but on a null data set.

As shown in Fig. 3, in the first 64,000 UNICODE code-points, there are defined 25 whitespace characters, 17 of which are space characters. For example, 0x0020, 0x00A0, 0x3000 are all examples of UNICODE space characters.

A Java™ code function to trim a UNICODE character string is shown in Table 2.

15

```
public static String getTrimmedUnicodeString(String s) {
    if (s==null || "".equals(s))
        return s;

    int len = s.length();
    int st = 0;
    char[] val = s.toCharArray();

    while ((st<len) && Character.isWhitespace(val[st])) {
        st++;
    }
    while ((st<len) && Character.isWhitespace(val[len-1])) {
        len--;
    }

    return s.substring(st, len);
}
```

**Table 2 – Java™ code to trim whitespace characters**

Fig. 4 shows an example of the output 400 from the Java™ function as shown in Table 2. The character string (i.e. semantic unit #1, UNICODE whitespace, semantic unit #2,..., semantic unit #n) has been trimmed of all whitespace at the beginning and the end of the string.

20

*Step 230 - Replace whitespace with code-point 0x20*

Referring again to **Fig. 2**, at step **230**, pure whitespace character sequences between the words/diagram characters of the character string are now replaced with a single ASCII space (code-point 0x20). **Table 3** shows an example Java™ function that replaces all  
s pure whitespace character sequences with the code-point 0x20. The example Java™ function shown in **Table 3** also accounts for UNICODE control characters for completeness.

```
public static String slimdown(String s) {
    if (s==null || "".equals(s))
        Return s;
    s = getTrimmedUNICODEString(s); // see above
    if ("".equals(s))
        Return s;

    StringBuffer sbuff = new StringBuffer();
    StringTokenizer st = new StringTokenizer(s,
Constants.UNICODE_WHITESPACE_CHARACTERS_LIST); // see above
    Vector toks = new Vector();
    while (st.hasMoreTokens()) {
        toks.add(st.nextToken().trim());
    }
    int len = toks.size();
    int lastIndex = len-1;
    final String SPACE_0020 = "\u0020";
    for (int n=0; n<len; n++) {
        sbuff.append((String) toks.get(n));
        if (n<lastIndex)
            sbuff.append(SPACE_0020);
    }
    return new String(sbuff);
}
```

**Table 3 – Java™ function to replace whitespace with code point 0x20**

Referring now to **Fig. 5**, the output **500** from the Java™ function of **Table 3** shows all UNICODE whitespace has been replaced with ASCII space “0x20” **520**. ASCII space “0x20” is also mapped into most major character set encodings (such as Windows simplified Chinese). Accordingly, it is anticipated that the modified character string will  
5 display as originally intended in most situations.

#### *Step 240 – Store or Compare*

Referring again to **Fig. 2**, at step **240**, if the character string is new to the database, then it is stored in the database. If this is a subsequent entry of the character string, then it is  
10 compared with the previously stored version of the character string. If the user entered the same sequences of characters, ignoring whitespace, then the string comparison will match and the user will be provided a new password, and not an error message.

#### *Computer hardware*

**Fig. 6** is a schematic representation of a computer system **600** of a type that is suitable for processing whitespace in a user input. Computer software executes under a suitable operating system installed on the computer system **600**, and may be thought of as comprising various software code means for achieving particular steps. Such software can support a client/server architecture, in the sense that computer software embodying  
15 the invention can reside on a server machine and will serve user character inputs from client machines.

The components of the computer system **600** include a computer **620**, a keyboard **610** and mouse **615**, and a video display **690**. The computer **620** includes a processor **640**, a  
25 memory **650**, input/output (I/O) interfaces **660**, **665**, a video interface **645**, and a storage device **655**.

The processor **640** is a central processing unit (CPU) that executes the operating system and the computer software executing under the operating system. The memory **650**  
30 includes random access memory (RAM) and read-only memory (ROM), and is used under direction of the processor **640**.

The video interface 645 is connected to video display 690 and provides video signals for display on the video display 690. User input to operate the computer 620 is provided from the keyboard 610 and mouse 615. The storage device 655 can include a disk drive or any other suitable storage medium.

5

Each of the components of the computer 620 is connected to an internal bus 630 that includes data, address, and control buses, to allow components of the computer 620 to communicate with each other via the bus 630.

10

The computer system 600 can be connected to one or more other similar computers via an input/output (I/O) interface 665 using a communication channel 685 to a network, represented as the Internet 680. The computer system can function as either a client or server in a networked architecture.

15

The computer software may be recorded on a portable storage medium, in which case, the computer software program is accessed by the computer system 600 from the storage device 655. Alternatively, the computer software can be accessed directly from the Internet 680 by the computer 620. In either case, a user can interact with the computer system 600 using the keyboard 610 and mouse 615 to operate the programmed computer software executing on the computer 620.

20

Other configurations or types of computer systems can be equally well used to execute computer software that assists in implementing the techniques described herein.

25

#### *Further Embodiments*

Although the invention has been described in what is conceived to be the most practical and preferred embodiments, it is understood that further embodiments are within the scope of the invention. For example, the invention may equally be deployed in a client/server application environment; a distributed software application environment; or in a stand-alone application environment.

30

*Conclusion*

Various alterations and modifications can be made to the techniques and arrangements described herein, as would be apparent to one skilled in the relevant art.

**Claims:**

1. A method for processing a user character input, said method comprising the steps of:
  - 5 converting a user input character string to a Universal Character Set (UCS) string;
  - removing any leading and any trailing whitespace characters in said UCS string;
  - and
  - converting all remaining whitespace characters to a single value.
- 10 2. The method of claim 1, wherein said single value is hexadecimal 20.
3. The method of claim 1, further comprising the step of storing said converted UCS string in memory.
- 15 4. The method of claim 1, further comprising the step of determining if a converted UCS string has been previously stored in memory and, if so, comparing said converted UCS string with said previously stored converted UCS string.
- 20 5. The method of any one of claims 1, 2, 3 or 4, wherein said UCS is UNICODE.
6. A computer program product comprising a computer readable medium having a computer program recorded therein for processing a user character input, said computer program including:
  - 25 code means for converting a user input character string to a Universal Character Set (UCS) string;
  - code means for removing any leading and any trailing whitespace characters in said UCS string; and
  - code means for converting all remaining whitespace characters to
  - 30 a single value.
7. The computer program product of claim 6, wherein said single value is hexadecimal 20.

8. The computer program product of claim 6, further comprising code means for storing said converted UCS string in a database.

9. The computer program product of claim 6, further comprising code means for  
5 determining if a converted UCS string has been previously stored in the database and if  
so, comparing said converted UCS string with said previously stored converted UCS  
string.

10. The computer program product of claim 6, wherein said computer program is  
10 written in the Java language.

11. The computer program product of any one of claims 6, 7, 8, 9 or 10, wherein  
said UCS is UNICODE.

12. A computer system for processing a user character input, said system  
15 comprising:

a memory unit for storing data and instructions to be performed by a processing  
unit; and

a processing unit coupled to said memory unit, said processing unit being  
20 programmed to:

convert a user input character string to a Universal Character Set (UCS) string;

remove any leading and any trailing whitespace characters in said UCS string;

and

convert all remaining whitespace characters to a single value.

25

13. The computer system of claim 12, wherein said single value is hexadecimal 20.

14. The computer system of claim 12, wherein said processing unit is further  
programmed to store said converted UCS string in a database.

30

15. The computer system of claim 12, wherein said processing unit is further  
programmed to determine if a converted UCS string has been previously stored in the  
database and if so, said processing unit compares said converted UCS string with said  
previously stored converted UCS string.

16. The computer system of claim 12, wherein the processing unit is programmed in the Java language.

5 17. The computer system of any one of claims 12, 13, 14, 15 or 16, wherein said UCS is UNICODE.