



IBM Java™ 2 SDK HeapAnalyzer for Windows Version 1.1

Jinwoo Hwang
jinwoo@us.ibm.com

Welcome to IBM Java 2 SDK HeapAnalyzer. This tool is for internal use only.
Please do not distribute the tool externally since International Patent disclosures were submitted with this tool.

Introduction

The Heapdump contains a list of all the objects that are in the heap.

Heapdumps can be very large with millions of items in them.

It's not always easy to analyze a large file. This tool analyzes heapdumps of Java SDK 1.3.1 and 1.4.1.

HeapAnalyzer is an unofficial tool and is provided “as-is”.

Prerequisite

Java 2 SDK 1.4.1 or higher (Java 2 runtime 1.4.1 is provided with this tool)

Definitions

root object An object for which no (different) object holds a reference.

parent object An object (for example, A) that holds at least one reference to some (different) object (for example, B). In this case, A is said to be the parent of B.

Owner object If an object has more than one parent object, a parent object is chosen as owner object.

Total size is calculated only with owner objects.

child object An object (for example, B) for which at least one (different) object (for example, A) holds a reference. In this case B is said to be the child of A.

type Collection of same objects

size The size of an object is the amount of memory that is required to hold that object in memory.

total size The subtree size of an object is the sum of its size and the sizes of all the objects that it reached from its children. Note that each object is assigned a unique parent and root during processing.

Features

Creates a tree from heapdump

Calculates size of each objects

Calculates total size of each subtree

Finds size drop in a subtree

Shows free heap space by size

Shows object by size

Shows types by size

Shows types by count

Shows types alphabetical order

Shows free heap space distribution

Shows detailed information of an object

Finds type with regular expression

Drag and drop support in input fields and text

How to run this tool

Java runtime environment is provided with this tool.

Usage `jre\bin\java -Xmx[heapsize] -jar ha.jar`

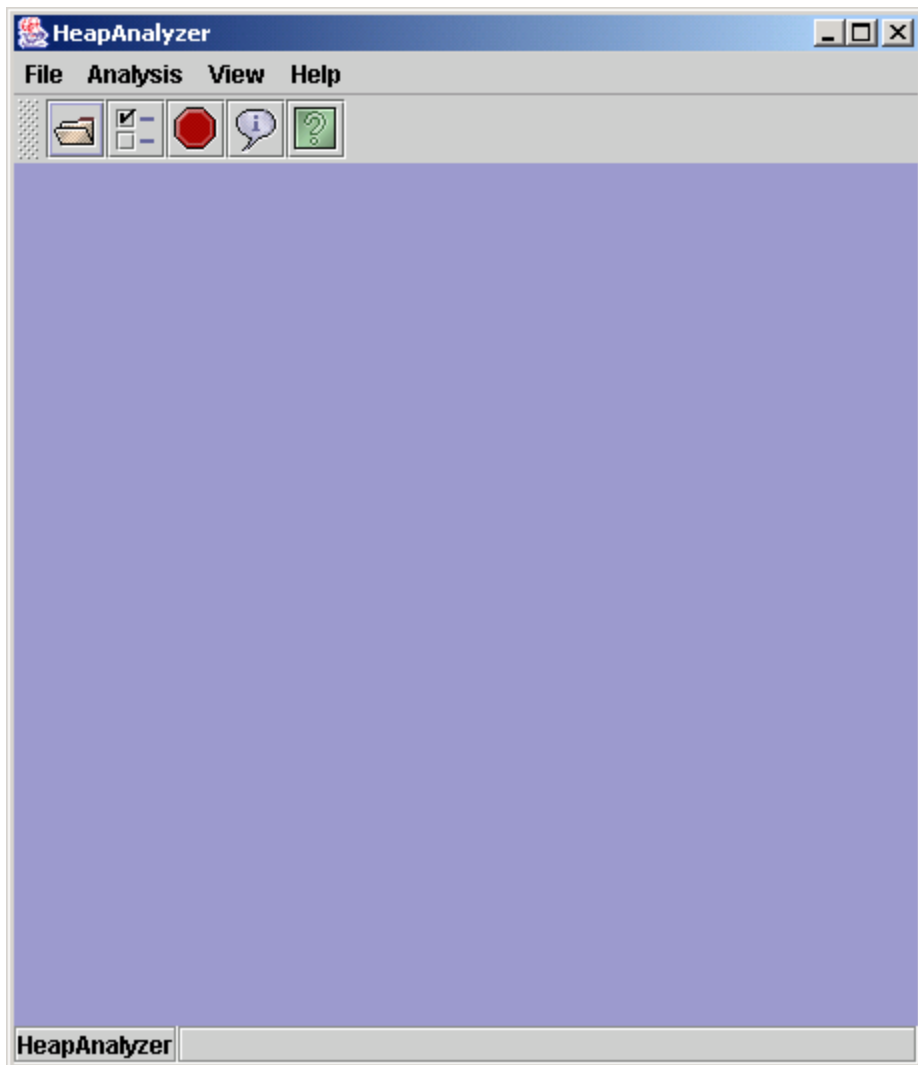
For example, `jre\bin\java -Xmx1000m -jar ha.jar`

If there's `java.lang.OutOfMemoryError`, please try increasing the maximum heap size (`-Xmx`) value to give the JVM more memory.

Maximum heap size should not be larger than the size of available physical memory size for this tool due to performance issue.

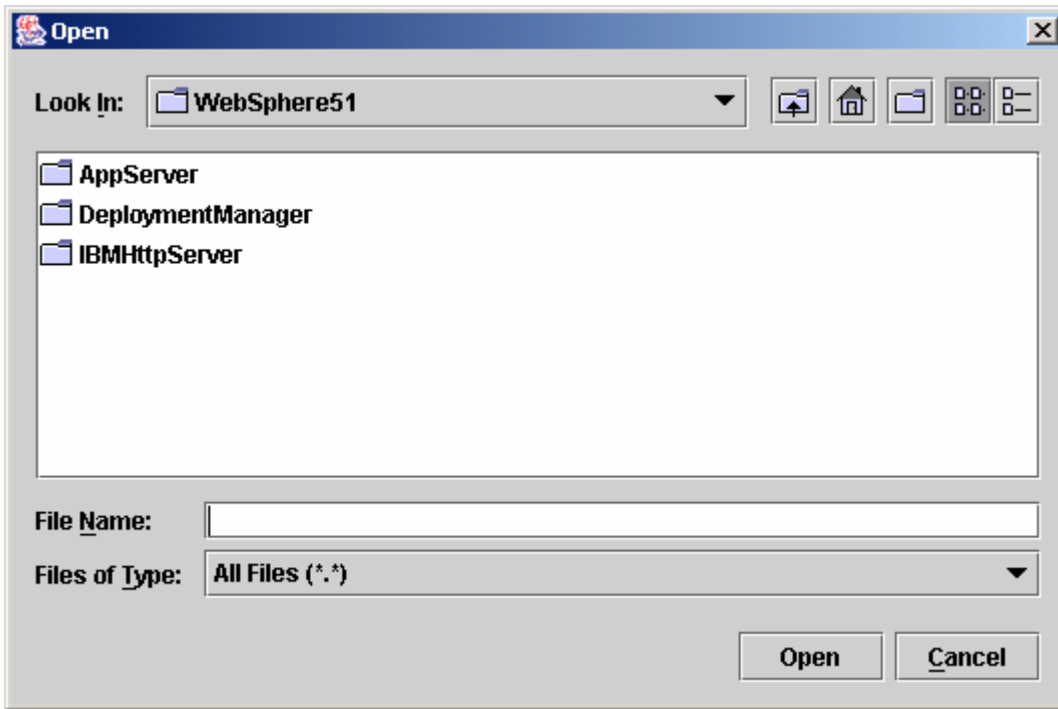
Feel free to contact me if you have any comments or suggestions.

1. Start the tool.

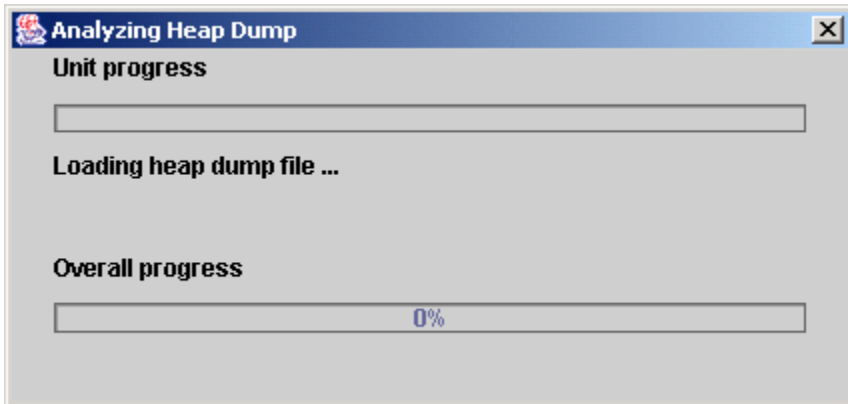


[Initial screen]

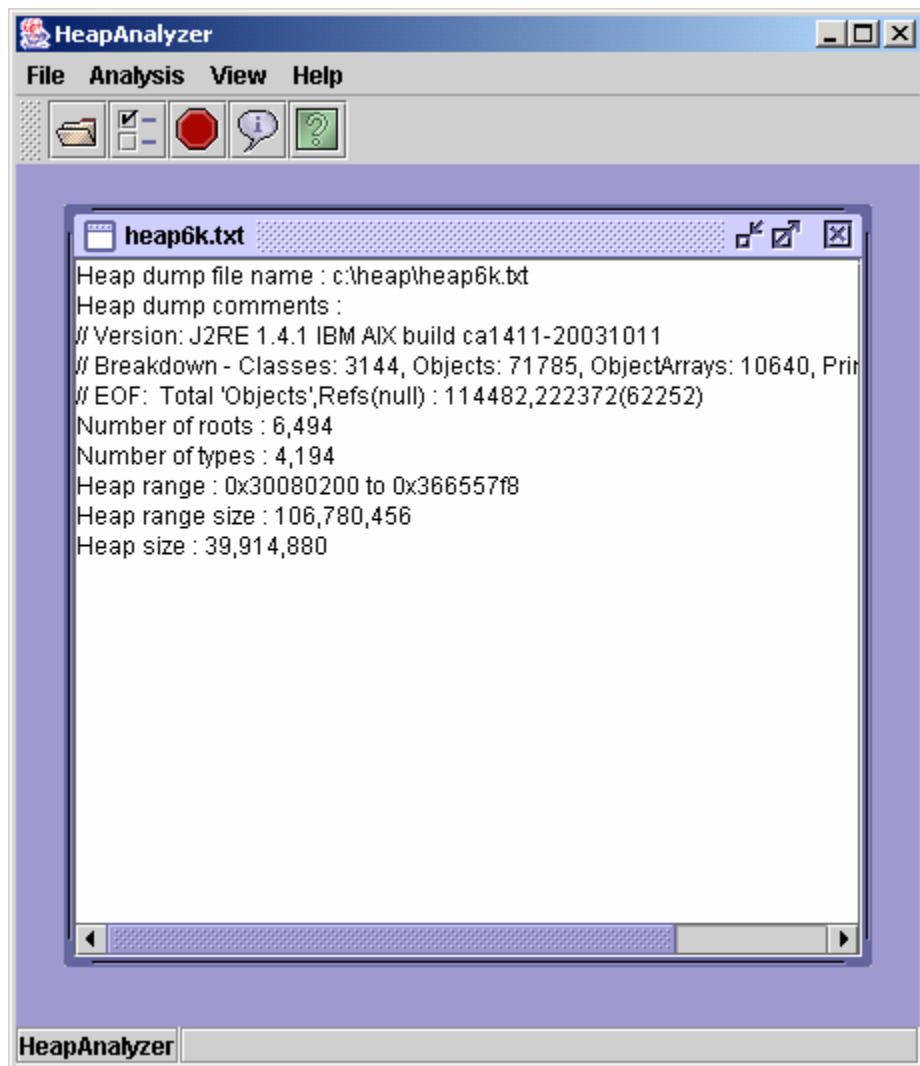
2. Select File -> Open and select a heapdump file



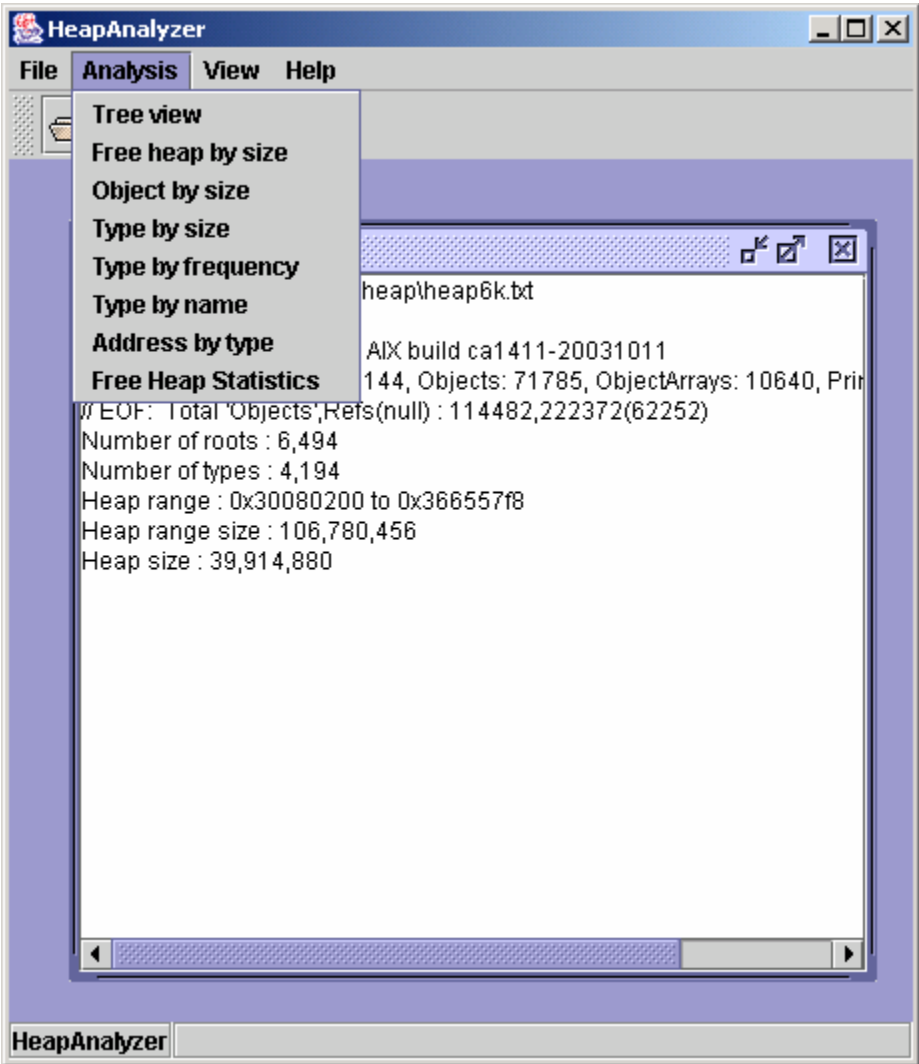
3. Progress is shown during processing heapdump.




4. It would take lots of time if you are processing large heapdump. The following is the screen when processing is complete. Please do not close this window until you do not need this heapdump.



Click on Analysis menu and select a menu item for further analysis.

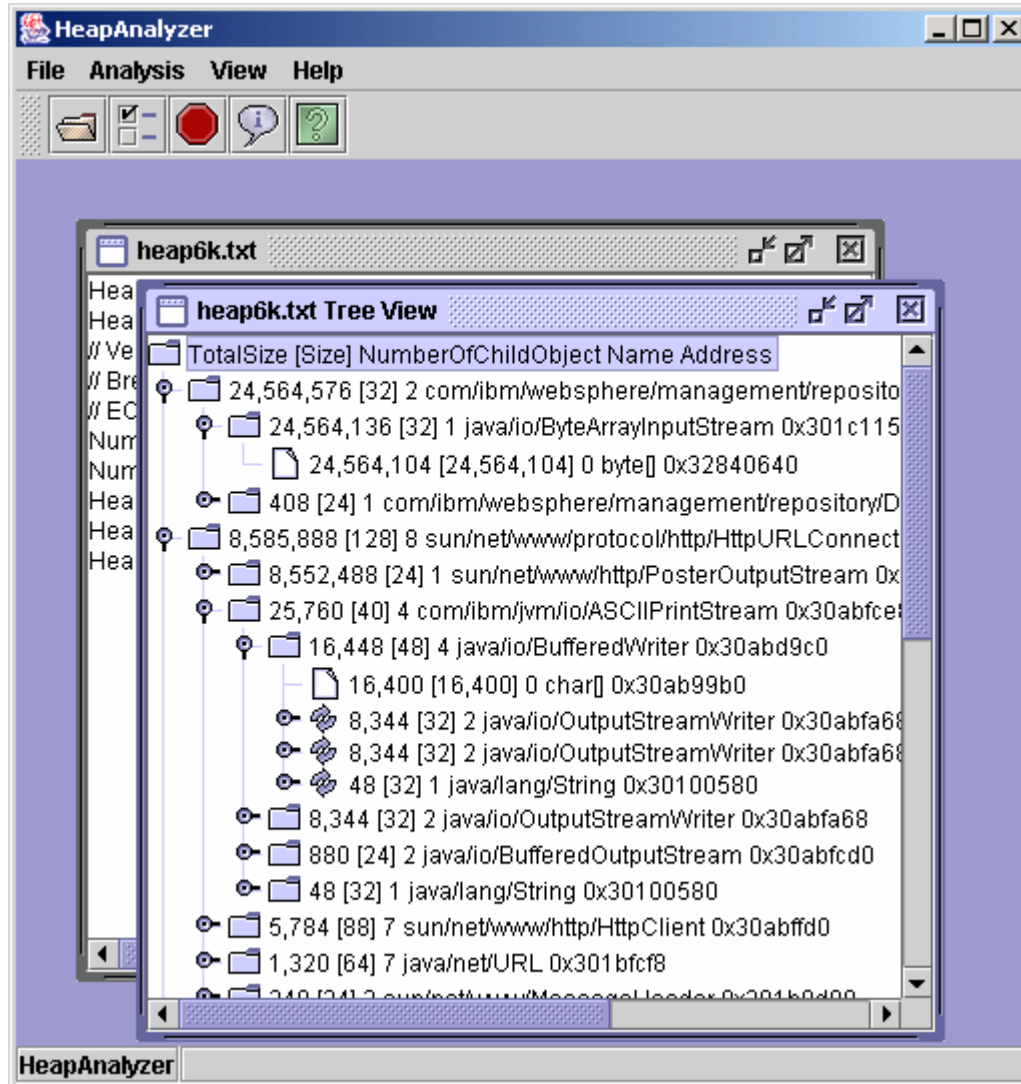


5. The following is tree view of the heapdump.

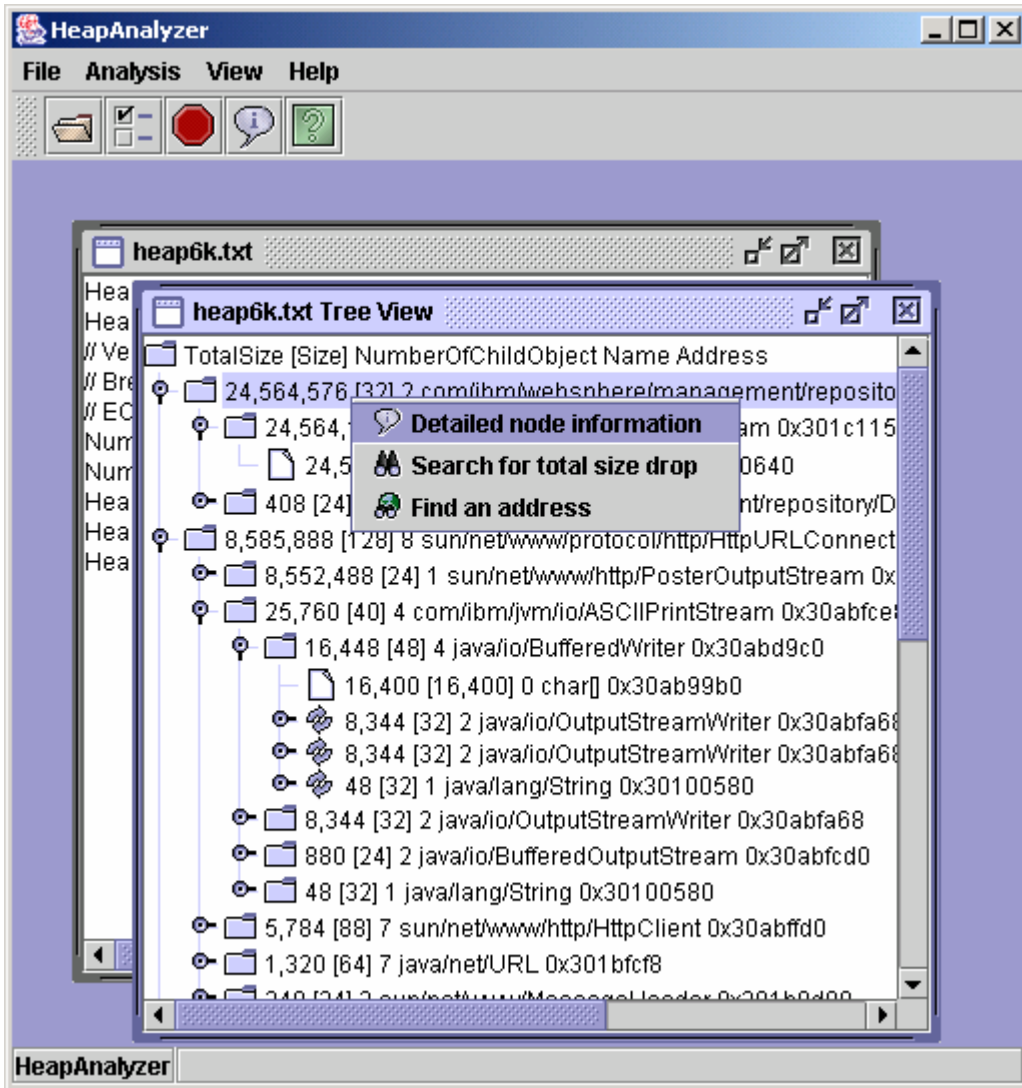
The icon, , indicates that it has already been included as a child object of owner object in tree view

Each tree node as in the following format:

TotalSize[Size] NumberOfChildObject Name Address



6. In tree view, you can see detailed information of a node , you can search for total size drop between parent and child
or you can find an address by selecting a node and click on right mouse button.

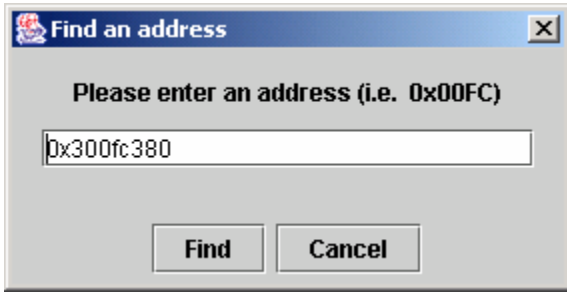


“Search for total size drop” will find a size drop between the total size of a parent and the biggest total size of child of the parent.

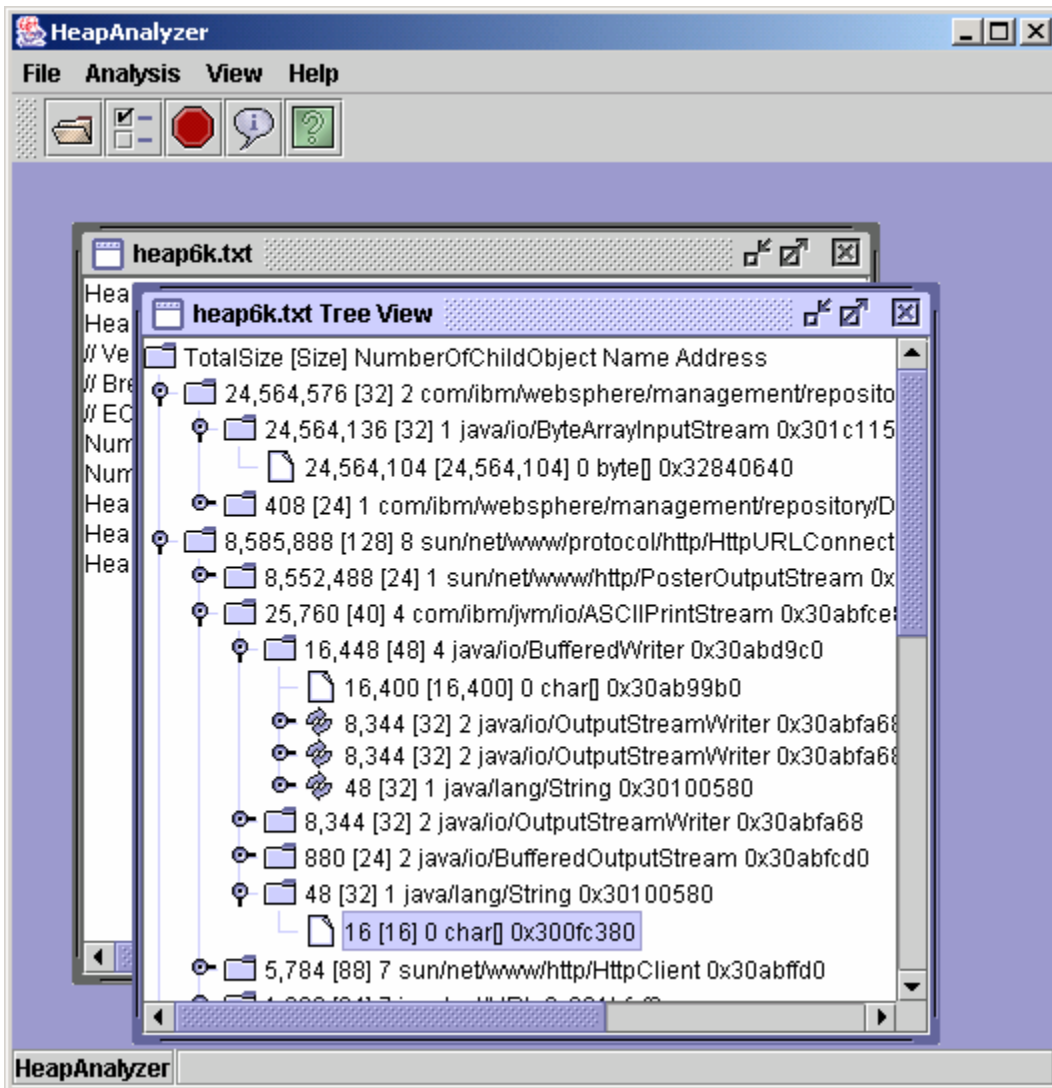
If you cannot find any size drop from the menu “Search for total size drop”, you need to decrease Minimum total size drop for search in options.



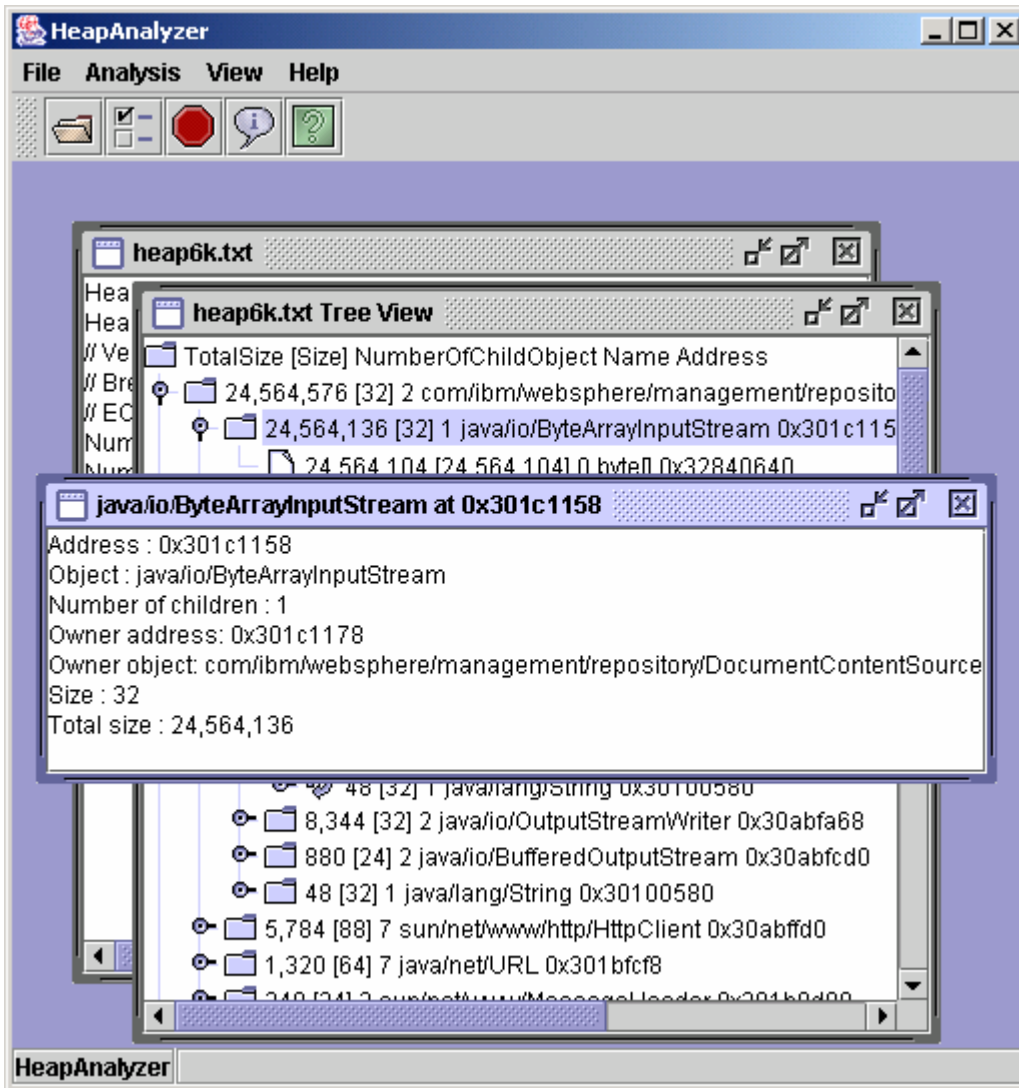
You can find an address in the tree view by selecting the menu “Find an address”



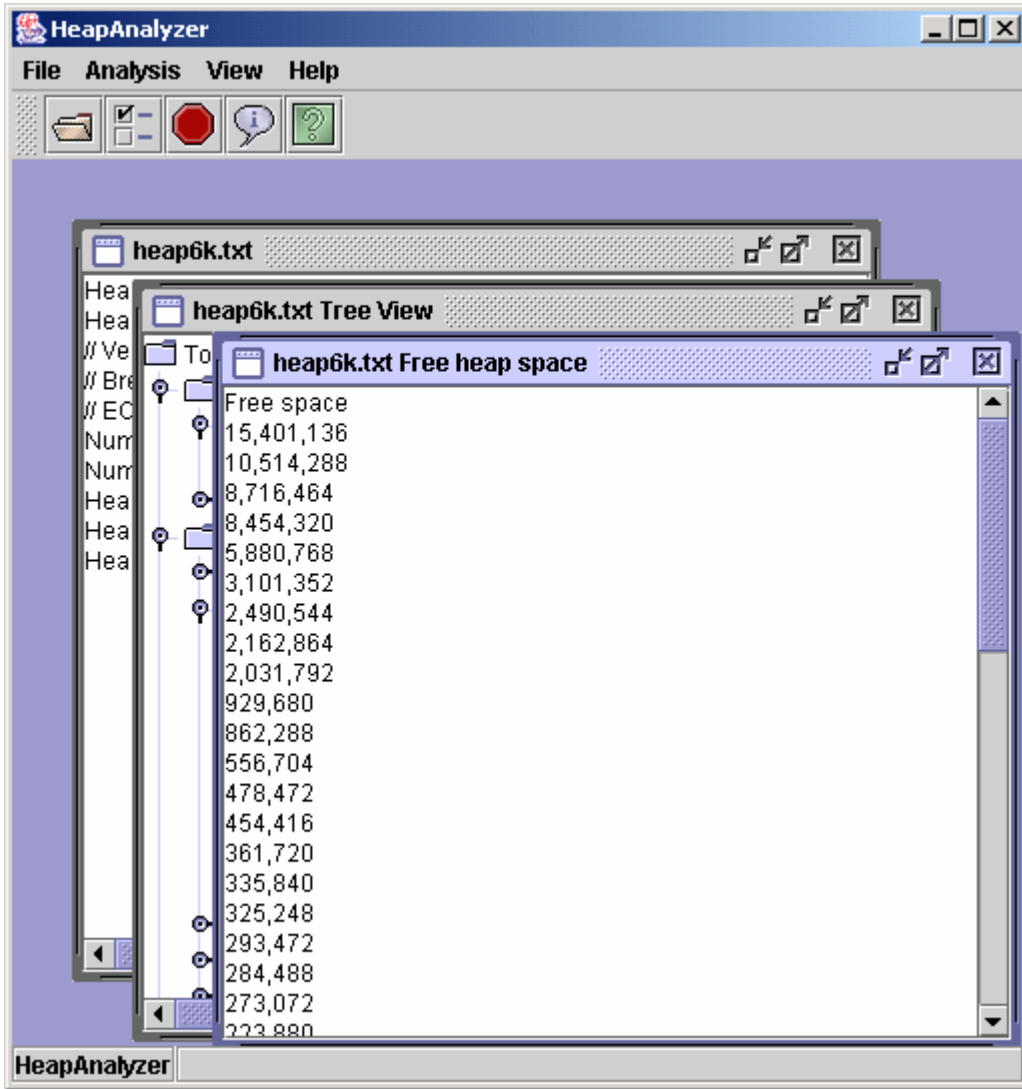
The following is the result of address search :



7. The following is the screen of detailed node information in heapdump tree



8. The following is free heap space view

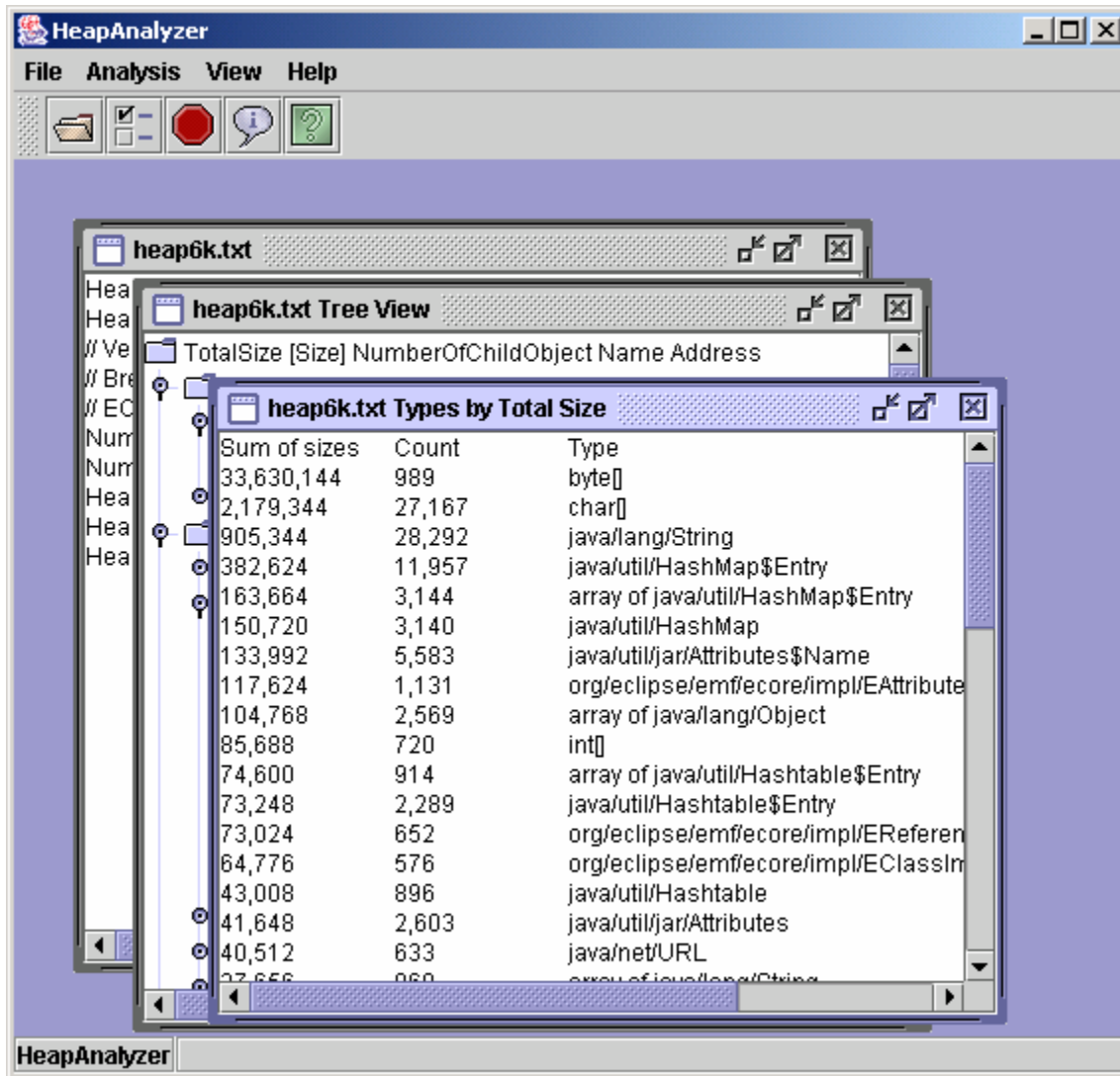


9. Object view by size

The screenshot displays the HeapAnalyzer application interface. The main window is titled "heap6k.txt Objects" and contains a table with the following columns: Size, Total, Address, Object, and NumberOfChildObject. The table lists various heap objects, including arrays of java/lang/Object and byte arrays, with their respective sizes and addresses.

| Size | Total | Address | Object | NumberOfChildObject |
|------------|------------|------------|---------------------------|---------------------|
| 24,564,104 | 24,564,104 | 0x32840640 | byte[] | 0 |
| 8,552,464 | 8,552,464 | 0x31d23388 | byte[] | 0 |
| 65,552 | 65,552 | 0x305d84b8 | byte[] | 0 |
| 65,552 | 65,552 | 0x305e84c8 | byte[] | 0 |
| 63,352 | 63,352 | 0x3012fef8 | byte[] | 0 |
| 36,696 | 36,696 | 0x30447060 | byte[] | 0 |
| 26,424 | 26,424 | 0x301c2950 | byte[] | 0 |
| 16,400 | 16,400 | 0x3010a040 | char[] | 0 |
| 16,400 | 16,400 | 0x3010e050 | char[] | 0 |
| 16,400 | 16,400 | 0x305915d0 | char[] | 0 |
| 16,400 | 16,400 | 0x30ab99b0 | char[] | 0 |
| 12,096 | 12,096 | 0x30444120 | byte[] | 0 |
| 10,256 | 2,620,632 | 0x30904730 | array of java/lang/Object | 1,750 |
| 8,304 | 8,304 | 0x3018ad30 | byte[] | 0 |
| 8,216 | 8,528 | 0x31aef710 | array of java/lang/Object | 7 |
| 8,208 | 8,208 | 0x300fd9c8 | byte[] | 0 |
| 8,208 | 8,208 | 0x30106658 | byte[] | 0 |
| 8,208 | 8,208 | 0x301f2bc0 | byte[] | 0 |
| 8,208 | 8,208 | 0x3057f648 | byte[] | 0 |

10. Type view by total size view



11. Types view by frequency

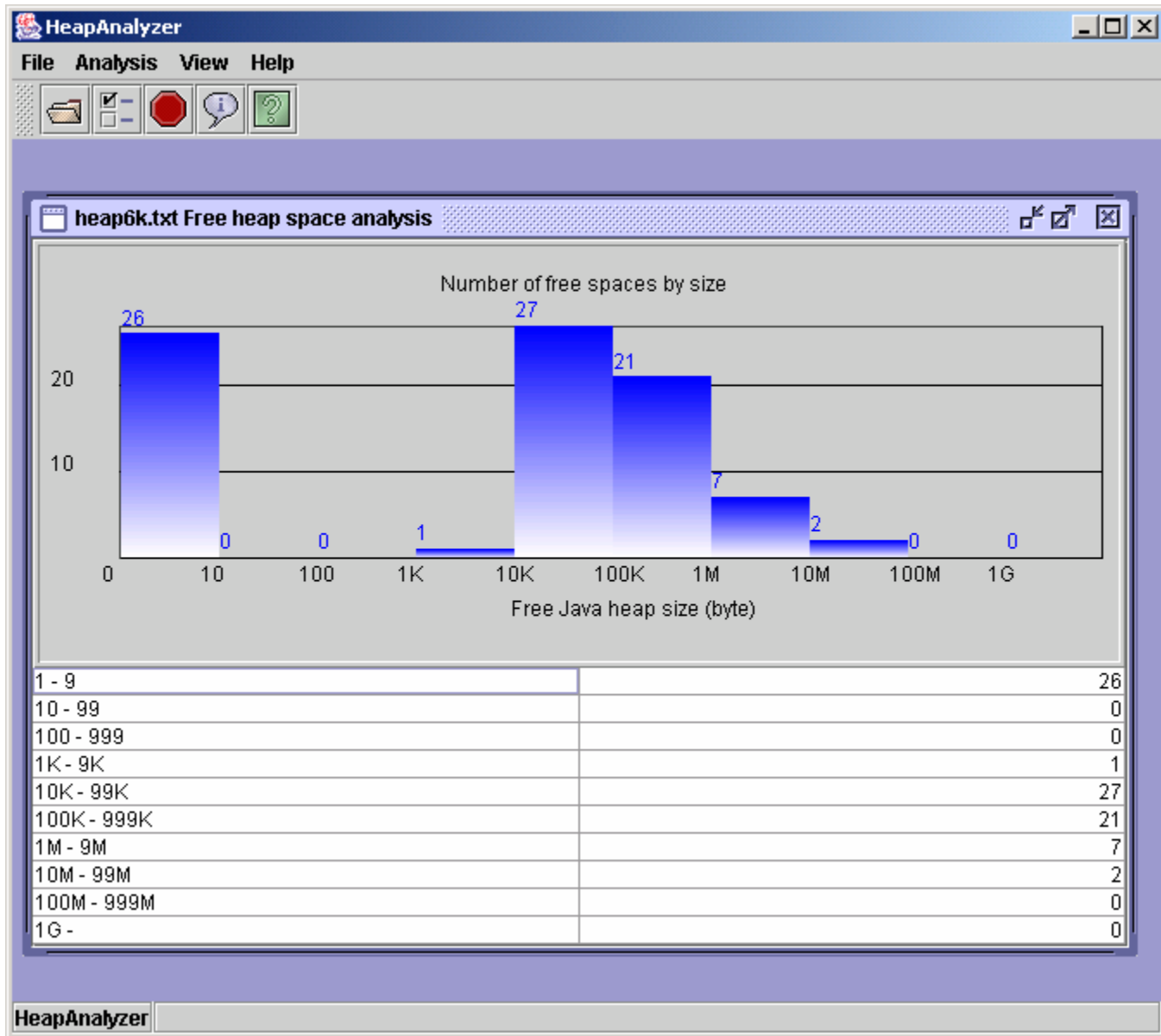
The screenshot displays the HeapAnalyzer application interface. The main window is titled 'HeapAnalyzer' and has a menu bar with 'File', 'Analysis', 'View', and 'Help'. Below the menu bar is a toolbar with icons for file operations and help. The main workspace contains several overlapping windows:

- 'heap6k.txt' (background)
- 'heap6k.txt Tree View' (middle background)
- 'heap6k.txt Types by Frequency' (foreground, active window)

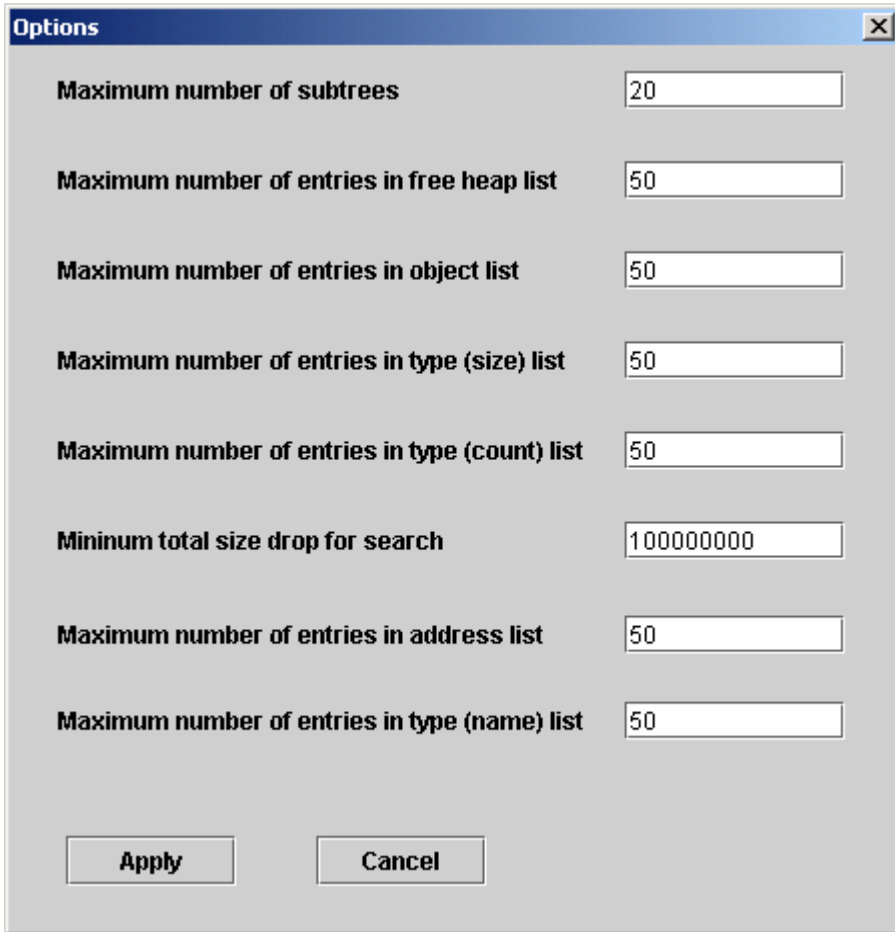
The 'Types by Frequency' window displays a table with the following data:

| Count | Sum of sizes | Type |
|--------|--------------|---------------------------------------|
| 28,292 | 905,344 | java/lang/String |
| 27,167 | 2,179,344 | char[] |
| 11,957 | 382,624 | java/util/HashMap\$Entry |
| 5,583 | 133,992 | java/util/jar/Attributes\$Name |
| 3,144 | 163,664 | array of java/util/HashMap\$Entry |
| 3,140 | 150,720 | java/util/HashMap |
| 2,603 | 41,648 | java/util/jar/Attributes |
| 2,569 | 104,768 | array of java/lang/Object |
| 2,289 | 73,248 | java/util/Hashtable\$Entry |
| 1,131 | 117,624 | org/eclipse/emf/ecore/impl/EAttribute |
| 989 | 33,630,144 | byte[] |
| 968 | 37,656 | array of java/lang/String |
| 914 | 74,600 | array of java/util/Hashtable\$Entry |
| 896 | 43,008 | java/util/Hashtable |
| 852 | 34,080 | org/eclipse/emf/ecore/util/EObjectCo |
| 811 | 32,440 | org/eclipse/emf/ecore/impl/EObjectTr |
| 739 | 17,736 | org/eclipse/emf/ecore/util/EContents |

12. Free heap space distribution view



13. You can configure setting in File-> Options menu

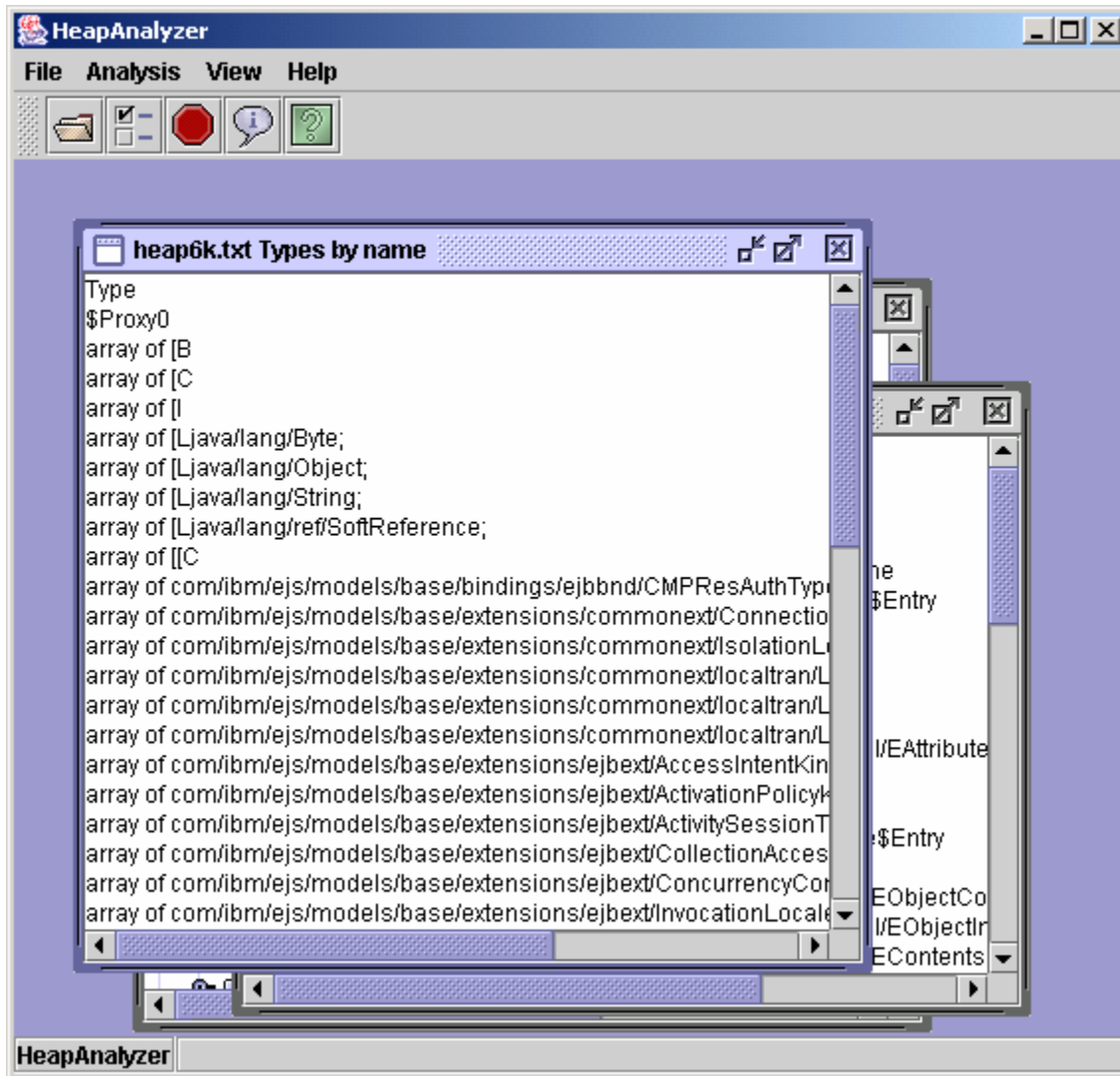


The image shows a dialog box titled "Options" with a close button (X) in the top right corner. The dialog contains several configuration options, each with a text input field:

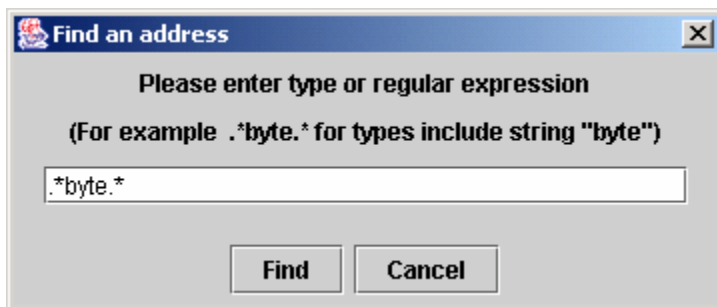
- Maximum number of subtrees: 20
- Maximum number of entries in free heap list: 50
- Maximum number of entries in object list: 50
- Maximum number of entries in type (size) list: 50
- Maximum number of entries in type (count) list: 50
- Minimum total size drop for search: 100000000
- Maximum number of entries in address list: 50
- Maximum number of entries in type (name) list: 50

At the bottom of the dialog, there are two buttons: "Apply" and "Cancel".

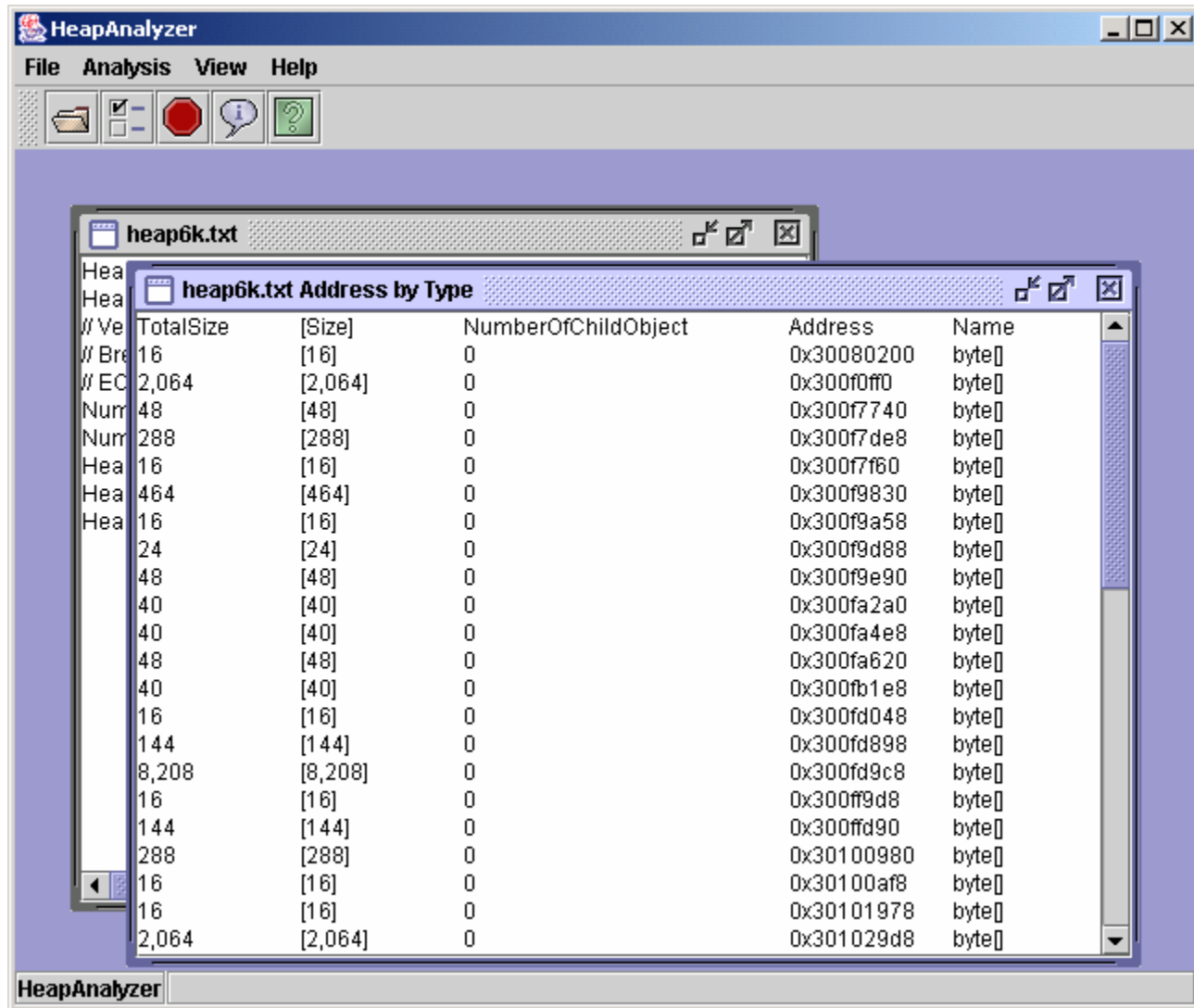
14. Type by alphabetical order



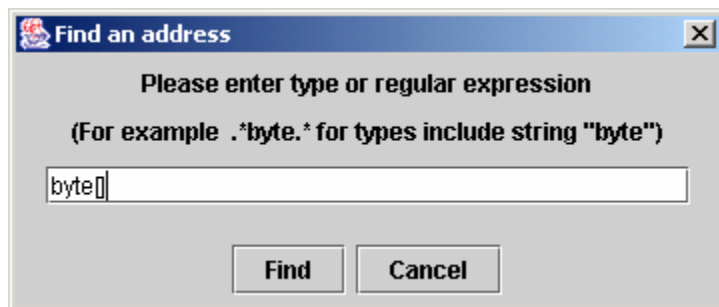
15. Address by type to find types include string "byte"



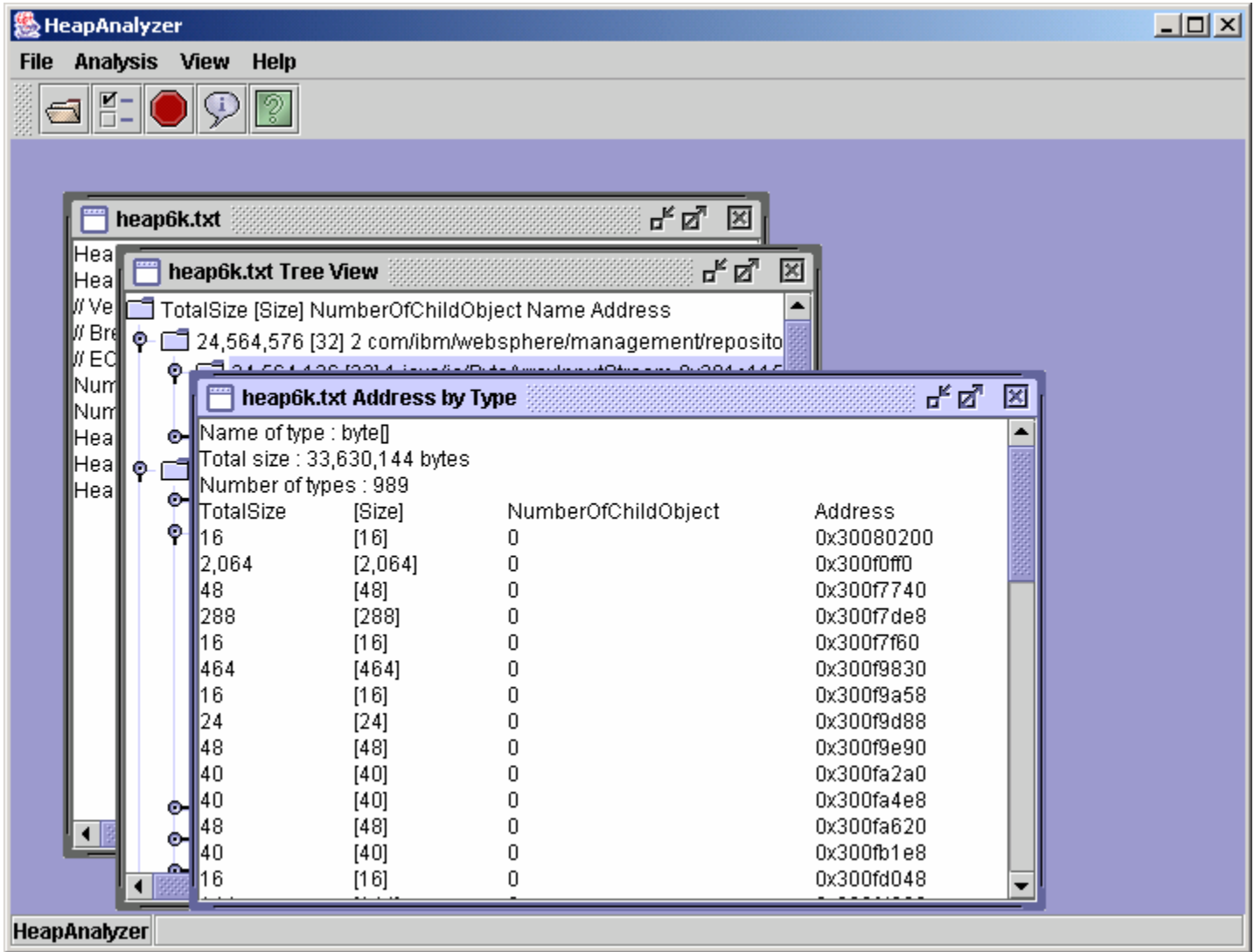
The following is the list of types which have “byte” in their names.



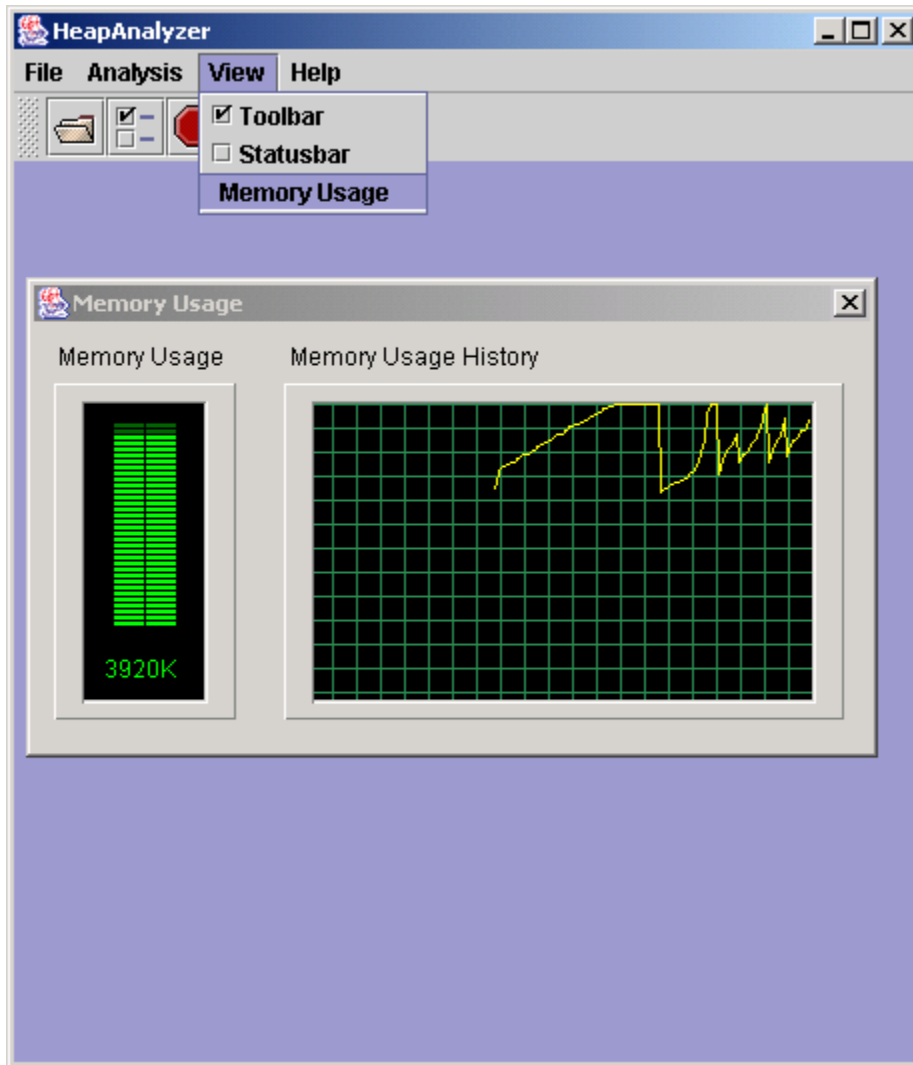
You can also enter exact name of a type:



The following is the list of types of byte[]



16. You can monitor memory usage in “Memory Usage” menu.



Summary of regular-expression constructs

| Construct | Matches |
|--------------------|--|
| Characters | |
| x | The character x |
| \backslash | The backslash character |
| $\backslash 0n$ | The character with octal value $0n$ ($0 \leq n \leq 7$) |
| $\backslash 0nn$ | The character with octal value $0nn$ ($0 \leq n \leq 7$) |
| $\backslash 0mnn$ | The character with octal value $0mnn$ ($0 \leq m \leq 3, 0 \leq n \leq 7$) |
| $\backslash xhh$ | The character with hexadecimal value $0xhh$ |
| $\backslash uhhhh$ | The character with hexadecimal value $0xhhhh$ |
| $\backslash t$ | The tab character (' $\backslash u0009$ ') |
| $\backslash n$ | The newline (line feed) character (' $\backslash u000A$ ') |
| $\backslash r$ | The carriage-return character (' $\backslash u000D$ ') |
| $\backslash f$ | The form-feed character (' $\backslash u000C$ ') |
| $\backslash a$ | The alert (bell) character (' $\backslash u0007$ ') |

`\e` The escape character ('`\u001B`')
`\cx` The control character corresponding to *x*

Character classes

`[abc]` a, b, or c (simple class)
`[^abc]` Any character except a, b, or c (negation)
`[a-zA-Z]` a through z or A through Z, inclusive (range)
`[a-d[m-p]]` a through d, or m through p: `[a-dm-p]` (union)
`[a-z&&[def]]` d, e, or f (intersection)
`[a-z&&[^bc]]` a through z, except for b and c: `[ad-z]` (subtraction)
`[a-z&&[^m-p]]` a through z, and not m through p: `[a-lq-z]` (subtraction)

Predefined character classes

`.` Any character (may or may not match line terminators)
`\d` A digit: `[0-9]`
`\D` A non-digit: `[^0-9]`
`\s` A whitespace character: `[\t\n\x0B\f\r]`
`\S` A non-whitespace character: `[^\s]`
`\w` A word character: `[a-zA-Z_0-9]`
`\W` A non-word character: `[^\w]`

POSIX character classes (US-ASCII only)

`\p{Lower}` A lower-case alphabetic character: `[a-z]`
`\p{Upper}` An upper-case alphabetic character: `[A-Z]`
`\p{ASCII}` All ASCII: `[\x00-\x7F]`
`\p{Alpha}` An alphabetic character: `[\p{Lower}\p{Upper}]`
`\p{Digit}` A decimal digit: `[0-9]`
`\p{Alnum}` An alphanumeric character: `[\p{Alpha}\p{Digit}]`
`\p{Punct}` Punctuation: One of `!"#$%&'()*+,-./:;<=>?@[\\]^_`{|}~`
`\p{Graph}` A visible character: `[\p{Alnum}\p{Punct}]`
`\p{Print}` A printable character: `[\p{Graph}]`
`\p{Blank}` A space or a tab: `[\t]`
`\p{Cntrl}` A control character: `[\x00-\x1F\x7F]`
`\p{XDigit}` A hexadecimal digit: `[0-9a-fA-F]`
`\p{Space}` A whitespace character: `[\t\n\x0B\f\r]`

Classes for Unicode blocks and categories

`\p{InGreek}` A character in the Greek block (simple block)
`\p{Lu}` An uppercase letter (simple category)
`\p{Sc}` A currency symbol
`\P{InGreek}` Any character except one in the Greek block (negation)
`[\p{L}&&[^p{Lu}]]` Any letter except an uppercase letter (subtraction)

Boundary matchers

`^` The beginning of a line
`$` The end of a line

| | |
|-----------------|---|
| <code>\b</code> | A word boundary |
| <code>\B</code> | A non-word boundary |
| <code>\A</code> | The beginning of the input |
| <code>\G</code> | The end of the previous match |
| <code>\Z</code> | The end of the input but for the final terminator, if any |
| <code>\z</code> | The end of the input |

Greedy quantifiers

| | |
|-------------|--|
| $X?$ | X , once or not at all |
| X^* | X , zero or more times |
| X^+ | X , one or more times |
| $X\{n\}$ | X , exactly n times |
| $X\{n, \}$ | X , at least n times |
| $X\{n, m\}$ | X , at least n but not more than m times |

Reluctant quantifiers

| | |
|--------------|--|
| $X??$ | X , once or not at all |
| $X*?$ | X , zero or more times |
| $X+?$ | X , one or more times |
| $X\{n\}?$ | X , exactly n times |
| $X\{n, \}?$ | X , at least n times |
| $X\{n, m\}?$ | X , at least n but not more than m times |

Possessive quantifiers

| | |
|--------------|--|
| $X?+$ | X , once or not at all |
| $X*+$ | X , zero or more times |
| $X++$ | X , one or more times |
| $X\{n\}+$ | X , exactly n times |
| $X\{n, \}+$ | X , at least n times |
| $X\{n, m\}+$ | X , at least n but not more than m times |

Logical operators

| | |
|-------|----------------------------|
| XY | X followed by Y |
| $X Y$ | Either X or Y |
| (X) | X , as a capturing group |

Back references

| | |
|-----------------|--|
| <code>\n</code> | Whatever the n^{th} capturing group matched |
|-----------------|--|

Quotation

| | |
|-----------------|--|
| <code>\</code> | Nothing, but quotes the following character |
| <code>\Q</code> | Nothing, but quotes all characters until <code>\E</code> |
| <code>\E</code> | Nothing, but ends quoting started by <code>\Q</code> |

Special constructs (non-capturing)

| | |
|---------|--------------------------------|
| $(?:X)$ | X , as a non-capturing group |
|---------|--------------------------------|

| | |
|---------------------------------|---|
| (? idmsux-idmsux) | Nothing, but turns match flags on - off |
| (? idmsux-idmsux:X) | <i>X</i> , as a non-capturing group with the given flags on - off |
| (? =<i>X</i>) | <i>X</i> , via zero-width positive lookahead |
| (? !<i>X</i>) | <i>X</i> , via zero-width negative lookahead |
| (? <=<i>X</i>) | <i>X</i> , via zero-width positive lookbehind |
| (? <!<i>X</i>) | <i>X</i> , via zero-width negative lookbehind |
| (? ><i>X</i>) | <i>X</i> , as an independent, non-capturing group |

Copyrights and notices

(C) Copyright IBM Corporation 2004. All Rights Reserved. Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

AIX, Cloudscape, DB2, IBM, iSeries, OS/390, VisualAge, WebSphere, and z/OS are trademarks or registered trademarks of IBM Corporation in the United States and/or other countries.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation in the United States and/or other countries.

UNIX is a registered trademark of The Open Group.

Other company, product, and service names may be trademarks or service marks of others.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS TOOLS "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you. This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the tool. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this tools at any time without notices.