Large Entity Declaration

PATENT
IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Application Number: 12/656872
Current Date

Filing Date: 18 FEB 2010
Patent Number: 8347314
Issue Date: 01 JAN 2013

CHANGE OF ENTITY STATUS PURSUANT TO 37 C.F.R. §1.27 (g)(2)

Commissioner for Patents
Mail Stop M Correspondence
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This communication hereby notifies the United States Patent and Trademark Office

that small entity status is no longer applicable for the above-identified patent.

COMPANY or FIRM
NAME AND ADDRESS:

SK planet
The Planet, Pangyo-ro 264,
Bundang-gu, Seongnam-si
Gyeonggi-do, 463-400, Korea

Respectfully submitted,

Signature
Printed Name
Title
OR
Reg. # if US Attorney
It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 9, claim 1, line 65, change “a-execution” to --execution--.
IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of:

Wonjang BAEK et al.

ATTENTION

Application No.: 12/656,872

CERTIFICATE OF CORRECTION

Confirmation No.: 2380

BRANCH

Filed: February 18, 2010

U.S. Patent No.: 8,347,314

Issue Date: January 1, 2013

For: METHOD FOR MANAGING JAVA APPLICATIONS

REQUEST FOR CERTIFICATE OF CORRECTION

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Patentee(s) respectfully request(s) that a Certificate of Correction be issued in the subject patent, pursuant to 35 U.S.C. §254 and 37 C.F.R. §1.322, to correct the mistake(s) shown on the attached Certificate of Correction form.

Since the mistakes are Patent Office mistakes, it is believed that no fee is required.

Respectfully submitted,

STEIN MCEWEN, LLP

Date: January 14, 2013

By:

Sungyeob Chung
Registration No. 64,130

1400 Eye St., N.W.
Suite 300
Washington, D.C. 20005
Telephone: (202) 216-9505
Facsimile: (202) 216-9510
UNITED STATES PATENT AND TRADEMARK OFFICE
CERTIFICATE OF CORRECTION

U.S. Patent No.: 8,347,314

Issue Date: January 1, 2013

Wonjang BAEK et al.

Column 9, claim 1, line 65, change "a-execution" to --execution--.

MAILING ADDRESS OF SENDER: STEIN MCEWEN, LLP
1400 Eye St., N.W.
Suite 300
Washington, DC 20005

PATENT NO. 8,347,314
No. of add'l copies @ 50¢ per page
Electronic Acknowledgement Receipt

<table>
<thead>
<tr>
<th>EFS ID:</th>
<th>14693050</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Number:</td>
<td>12656872</td>
</tr>
<tr>
<td>International Application Number:</td>
<td></td>
</tr>
<tr>
<td>Confirmation Number:</td>
<td>2380</td>
</tr>
<tr>
<td>Title of Invention:</td>
<td>METHOD FOR MANAGING JAVA APPLICATIONS</td>
</tr>
<tr>
<td>First Named Inventor/Applicant Name:</td>
<td>Wonjang Baek</td>
</tr>
<tr>
<td>Customer Number:</td>
<td>49455</td>
</tr>
<tr>
<td>Filer:</td>
<td>Sungyeop Chung/Christine Grimes</td>
</tr>
<tr>
<td>Filer Authorized By:</td>
<td>Sungyeop Chung</td>
</tr>
<tr>
<td>Attorney Docket Number:</td>
<td>0366.1006</td>
</tr>
<tr>
<td>Receipt Date:</td>
<td>14-JAN-2013</td>
</tr>
<tr>
<td>Filing Date:</td>
<td>18-FEB-2010</td>
</tr>
<tr>
<td>Time Stamp:</td>
<td>16:03:06</td>
</tr>
<tr>
<td>Application Type:</td>
<td>Utility under 35 USC 111(a)</td>
</tr>
</tbody>
</table>

Payment information:

Submitted with Payment: no

File Listing:

<table>
<thead>
<tr>
<th>Document Number</th>
<th>Document Description</th>
<th>File Name</th>
<th>File Size(Bytes)/Message Digest</th>
<th>Multi Part / .zip</th>
<th>Pages (if appl.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Request for Certificate of Correction</td>
<td>requestforCOC_011413.pdf</td>
<td>124480</td>
<td>no</td>
<td>2</td>
</tr>
</tbody>
</table>

Warnings:

Information:
This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/D0/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.
The projected patent number and issue date are specified above.

**Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)**

(application filed on or after May 29, 2000)

The Patent Term Adjustment is 416 day(s). Any patent to issue from the above-identified application will include an indication of the adjustment on the front page.

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Application Assistance Unit (AAU) of the Office of Data Management (ODM) at (571)-272-4200.

APPLICANT(s) (Please see PAIR WEB site http://pair.uspto.gov for additional applicants):

Wonjang Baek, Seongnam-si, KOREA, REPUBLIC OF;
John Kim, Seoul, KOREA, REPUBLIC OF;

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The USA offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to encourage and facilitate business investment. To learn more about why the USA is the best country in the world to develop technology, manufacture products, and grow your business, visit SelectUSA.gov.
PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fee will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

49455 7590 11/02/2012
STEIN MCEWEN, LLP
1400 EYE STREET, NW
SUITE 300
WASHINGTON, DC 20005

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission
I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571)-273-885, on the date indicated below.

(Date)

APPLICATION NO. 12/656,872
FILING DATE 02/18/2010
FIRST NAMED INVENTOR Wonjung Baek
ATTORNEY DOCKET NO. 0366.1006
CONFIRMATION NO. 2380

TITLE OF INVENTION: METHOD FOR MANAGING JAVA APPLICATIONS

<table>
<thead>
<tr>
<th>APPL. TYPE</th>
<th>SMALL ENTITY</th>
<th>ISSUE FEE DUE</th>
<th>PUBLICATION FEE DUE</th>
<th>REV. PAID ISSUE FEE</th>
<th>TOTAL FEE(S) DUE</th>
<th>DATE DUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Provisional</td>
<td>Yes</td>
<td>$885</td>
<td>$300</td>
<td>$0</td>
<td>$1185</td>
<td>02/04/2013</td>
</tr>
</tbody>
</table>

EXAMINER NGUYEN, VAN H
ART UNIT 2199
CLASS-SUBCLASS 718-100000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
   - Change of correspondence address or (Change of Correspondence Address form PTO/SB/122) attached.
   - Fee Address" indication or "Fee Address" Indication form PTO/SB/47, Rev 03-02 or more recent added. Use of a Customer Number is required.

2. For printing on the patent front page, list
   - (1) the names of up to 3 registered patent attorneys or agents OR, alternatively,
   - (2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

   Stein McEwen, LLP

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

   PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing as assignment.

   (A) NAME OF ASSIGNEE
   SK Planet Co., Ltd.

   (B) RESIDENCE: (CITY and STATE OR COUNTRY)
   Seoul, Republic of Korea

4a. The following fee(s) are submitted:
   - Issue Fee
   - Publication Fee (No small entity discount permitted)
   - Advance Order - # of Copies

4b. Payment of Fee(s) (Please first reapply any previously paid issue fee shown above)
   - A check is enclosed.
   - Payment by credit card. Form PTO-2038 is attached.
   - The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account No. 50333333 (enclose an extra copy of this form).

5. Change in Entity Status (from status indicated above)
   a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.
   b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature

Typed or printed name

Date

Registration No.

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PTOL-85 (Rev. 02/11) Approved for use through 08/31/2013.

OMB 0651-0033 U.S. Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE
## Electronic Patent Application Fee Transmittal

<table>
<thead>
<tr>
<th>Application Number:</th>
<th>12656872</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filing Date:</td>
<td>18-Feb-2010</td>
</tr>
<tr>
<td>Title of Invention:</td>
<td>METHOD FOR MANAGING JAVA APPLICATIONS</td>
</tr>
<tr>
<td>First Named Inventor/Applicant Name:</td>
<td>Wonjang Baek</td>
</tr>
<tr>
<td>Filer:</td>
<td>Sungyeop Chung/Sarah Brogi</td>
</tr>
<tr>
<td>Attorney Docket Number:</td>
<td>0366.1006</td>
</tr>
<tr>
<td>Filed as Small Entity</td>
<td></td>
</tr>
</tbody>
</table>

### Utility under 35 USC 111(a) Filing Fees

<table>
<thead>
<tr>
<th>Description</th>
<th>Fee Code</th>
<th>Quantity</th>
<th>Amount</th>
<th>Sub-Total in USD($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Filing:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pages:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Claims:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Miscellaneous-Filing:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Petition:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Patent-Appeals-and-Interference:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Post-Allowance-and-Post-Issuance:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| Utility Appl issue fee          | 2501     | 1        | 885    | 885                  |
| Publ. Fee- early, voluntary, or normal | 1504     | 1        | 300    | 300                  |</p>
<table>
<thead>
<tr>
<th>Description</th>
<th>Fee Code</th>
<th>Quantity</th>
<th>Amount</th>
<th>Sub-Total in USD($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extension-of-Time:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Miscellaneous:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Total in USD ($)</strong></td>
<td></td>
<td></td>
<td></td>
<td><strong>1185</strong></td>
</tr>
</tbody>
</table>
## Electronic Acknowledgement Receipt

<table>
<thead>
<tr>
<th>Field</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>EFS ID:</td>
<td>14336012</td>
</tr>
<tr>
<td>Application Number:</td>
<td>12656872</td>
</tr>
<tr>
<td>International Application Number:</td>
<td></td>
</tr>
<tr>
<td>Confirmation Number:</td>
<td>2380</td>
</tr>
<tr>
<td>Title of Invention:</td>
<td>METHOD FOR MANAGING JAVA APPLICATIONS</td>
</tr>
<tr>
<td>First Named Inventor/Applicant Name:</td>
<td>Wonjang Baek</td>
</tr>
<tr>
<td>Customer Number:</td>
<td>49455</td>
</tr>
<tr>
<td>Filer:</td>
<td>Sungyeop Chung/Sarah Brogi</td>
</tr>
<tr>
<td>Filer Authorized By:</td>
<td>Sungyeop Chung</td>
</tr>
<tr>
<td>Attorney Docket Number:</td>
<td>0366.1006</td>
</tr>
<tr>
<td>Receipt Date:</td>
<td>29-NOV-2012</td>
</tr>
<tr>
<td>Filing Date:</td>
<td>18-FEB-2010</td>
</tr>
<tr>
<td>Time Stamp:</td>
<td>11:19:45</td>
</tr>
<tr>
<td>Application Type:</td>
<td>Utility under 35 USC 111(a)</td>
</tr>
</tbody>
</table>

### Payment Information:

- **Submitted with Payment**: Yes
- **Payment Type**: Credit Card
- **Payment was successfully received in RAM**: $1185
- **RAM confirmation Number**: 13471
- **Deposit Account**
- **Authorized User**

### File Listing:

<table>
<thead>
<tr>
<th>Document Number</th>
<th>Document Description</th>
<th>File Name</th>
<th>File Size(Bytes)/Message Digest</th>
<th>Multi Part/.zip</th>
<th>Pages (if appl.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Issue Fee Payment (PTO-85B)</td>
<td>Issue_Fee_112912.pdf</td>
<td>184135</td>
<td>no</td>
<td>1</td>
</tr>
<tr>
<td>---</td>
<td>---------------------------</td>
<td>----------------------</td>
<td>--------</td>
<td>----</td>
<td>---</td>
</tr>
</tbody>
</table>

**Warnings:**

**Information:**

<table>
<thead>
<tr>
<th></th>
<th>Fee Worksheet (S806)</th>
<th>fee-info.pdf</th>
<th>32100</th>
<th>no</th>
<th>2</th>
</tr>
</thead>
</table>

**Warnings:**

**Information:**

**Total Files Size (in bytes):** 216235

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

**New Applications Under 35 U.S.C. 111**
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

**National Stage of an International Application under 35 U.S.C. 371**
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

**New International Application Filed with the USPTO as a Receiving Office**
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.
NOTICE OF ALLOWANCE AND FEE(S) DUE

EXAMINER
EXAMINER

NGUYEN, VAN H
NGUYEN, VAN H

ART UNIT
PAPER NUMBER
ART UNIT
PAPER NUMBER

2199
2199

DATE MAILED: 11/02/2012
DATE MAILED: 11/02/2012

APPLICATION NO.
FILING DATE
FIRST NAMED INVENTOR
ATTORNEY DOCKET NO.
CONFIRMATION NO.
APPLICATION NO.
FILING DATE
FIRST NAMED INVENTOR
ATTORNEY DOCKET NO.
CONFIRMATION NO.

12/656,872
02/18/2010
Wonjang Baek
0366.1006
2380
12/656,872
02/18/2010
Wonjang Baek
0366.1006
2380

TITLE OF INVENTION: METHOD FOR MANAGING JAVA APPLICATIONS
TITLE OF INVENTION: METHOD FOR MANAGING JAVA APPLICATIONS

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT.
THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT.
PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS.
PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS.
THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON
THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON
PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.
PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE
THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE
MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS
MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS
STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES
STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES
NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS
NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS
PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM
PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM
WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW
WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW
DUE.
DUE.

HOW TO REPLY TO THIS NOTICE:
HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.
I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current
If the SMALL ENTITY is shown as YES, verify your current
SMALL ENTITY status:
SMALL ENTITY status:
A. If the status is the same, pay the TOTAL FEE(S) DUE shown
A. If the status is the same, pay the TOTAL FEE(S) DUE shown
above.
above.
B. If the status above is to be removed, check box 5b on Part B -
B. If the status above is to be removed, check box 5b on Part B -
Fee(s) Transmittal and pay the PUBLICATION FEE (if required)
Fee(s) Transmittal and pay the PUBLICATION FEE (if required)
and twice the amount of the ISSUE FEE shown above, or
and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:
If the SMALL ENTITY is shown as NO:
A. Pay TOTAL FEE(S) DUE shown above, or
A. Pay TOTAL FEE(S) DUE shown above, or
B. If applicant claimed SMALL ENTITY status before, or is now
B. If applicant claimed SMALL ENTITY status before, or is now
claiming SMALL ENTITY status, check box 5a on Part B - Fee(s)
claiming SMALL ENTITY status, check box 5a on Part B - Fee(s)
Transmittal and pay the PUBLICATION FEE (if required) and 1/2
Transmittal and pay the PUBLICATION FEE (if required) and 1/2
the ISSUE FEE shown above.
the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office
II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office
(USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" (ii
(USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b"
of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a
of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a
request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing
request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing
the paper as an equivalent of Part B.
the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please directly all communications prior to issuance to
III. All communications regarding this application must give the application number. Please directly all communications prior to issuance to
Mail Stop ISSUE FEE unless advised to the contrary.
Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of
IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of
maintenance fees. It is patentee’s responsibility to ensure timely payment of maintenance fees when due.
maintenance fees. It is patentee’s responsibility to ensure timely payment of maintenance fees when due.

Page 1 of 3
## PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: **Mail**

Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

or **Fax**

(571)-273-2885

**INSTRUCTIONS:** This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

**CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)**

49145 7590 11/2/2012

STEIN MCEWEN, LLP
1400 EYE STREET, NW
SUITE 300
WASHINGTON, DC 20005

**Note:** A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

**Certificate of Mailing or Transmission**

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

<table>
<thead>
<tr>
<th>(Deposit's Name)</th>
<th>(Signature)</th>
<th>(Date)</th>
</tr>
</thead>
</table>

**APPLICATION NO.**

12/656,872

**FILING DATE**

02/18/2010

**FIRST NAMED INVENTOR**

Wonjung Baek

**ATTORNEY DOCKET NO.**

0366.1006

**CONFIRMATION NO.**

2380

**TITLE OF INVENTION:** METHOD FOR MANAGING JAVA APPLICATIONS

<table>
<thead>
<tr>
<th>APPLN. TYPE</th>
<th>SMALL ENTITY</th>
<th>ISSUE FEE DUE</th>
<th>PUBLICATION FEE DUE</th>
<th>PREV. PAID ISSUE FEE</th>
<th>TOTAL FEES DUE</th>
<th>DATE DUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>nonprovisional</td>
<td>YES</td>
<td>$885</td>
<td>$300</td>
<td>$0</td>
<td>$1185</td>
<td>02/04/2013</td>
</tr>
</tbody>
</table>

**EXAMINER**

NGUYEN, VAN H

2199

718-1009000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).  
   - Change of correspondence address or Change of Correspondence Address form PTOB/122 attached.
   - "Fee Address" indication or "Fee Address" Indication form PTOB/847; Rev 03-02 or more recent attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
   1. the names of up to 3 registered patent attorneys or agents OR, alternatively,
   2. the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

3. **ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT** (print or type)

   PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is **NOT** a substitute for filing an assignment.

   (A) NAME OF ASSIGNEE

   (B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent):  
- Individual  
- Corporation or other private group entity  
- Government

4a. The following fee(s) are submitted:
   - Issue Fee
   - Publication Fee (No small entity discount permitted)
   - Advance Order - # of Copies

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**
   - A check is enclosed.
   - Payment by credit card. Form PTO-2038 is attached.
   - The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)
   - a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.  
   - b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

**NOTE:** The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

**Authorized Signature**

Date

**Typed or printed name**

Registration No.

This collection of information is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.
<table>
<thead>
<tr>
<th>APPLICATION NO.</th>
<th>FILING DATE</th>
<th>FIRST NAMED INVENTOR</th>
<th>ATTORNEY DOCKET NO.</th>
<th>CONFIRMATION NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/656,872</td>
<td>02/18/2010</td>
<td>Wonjang Baek</td>
<td>0366.1006</td>
<td>2380</td>
</tr>
<tr>
<td>49458</td>
<td>7590</td>
<td>11/02/2012</td>
<td>EXAMINER</td>
<td>NGUYEN, VAN H</td>
</tr>
<tr>
<td>STEIN MCEWEN, LLP</td>
<td>1400 EYE STREET, NW</td>
<td>2199</td>
<td>ART UNIT</td>
<td>PAPER NUMBER</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)  
(application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 416 day(s). If the issue fee is paid on the date that is three months after the mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a half months) after the mailing date of this notice, the Patent Term Adjustment will be 416 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.
Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.
Notice of Allowability

--- The MAILING DATE of this communication appears on the cover sheet with the correspondence address---

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS. This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to the amendment and supporting arguments filed 09/05/2012.

2. ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

3. ☒ The allowed claim(s) is/are 1-22.

4. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All  b) ☐ Some*  c) ☐ None  of the:
   1. ☐ Certified copies of the priority documents have been received.
   2. ☐ Certified copies of the priority documents have been received in Application No. _____.
   3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE MAILING DATE of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.

THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER’S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

6. ☐ CORRECTED DRAWINGS (as “replacement sheets”) must be submitted.
   a) ☐ including changes required by the Notice of Draftsperson’s Patent Drawing Review (PTO-948) attached
      1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   b) ☐ including changes required by the attached Examiner’s Amendment / Comment or in the Office action of Paper No./Mail Date _____.

Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

7. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner’s comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

---

Attachment(s)

1. ☒ Notice of References Cited (PTO-892)
3. ☐ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date _____
4. ☐ Examiner’s Comment Regarding Requirement for Deposit of Biological Material
6. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____
7. ☐ Examiner’s Amendment/Comment
8. ☐ Examiner’s Statement of Reasons for Allowance
9. ☐ Other ______.

/VAN H NGUYEN/
Primary Examiner, Art Unit 2199
<table>
<thead>
<tr>
<th>*</th>
<th>Document Number</th>
<th>Date MM-YYYY</th>
<th>Name</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>US-8,015,548</td>
<td>09-2011</td>
<td>Chen et al.</td>
<td>717/118</td>
</tr>
</tbody>
</table>

**FOREIGN PATENT DOCUMENTS**

<table>
<thead>
<tr>
<th>*</th>
<th>Document Number</th>
<th>Date MM-YYYY</th>
<th>Country</th>
<th>Name</th>
</tr>
</thead>
</table>

**NON-PATENT DOCUMENTS**

<table>
<thead>
<tr>
<th>*</th>
<th>Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages</th>
</tr>
</thead>
</table>

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)

Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.
**Search Notes**

<table>
<thead>
<tr>
<th>Class</th>
<th>Subclass</th>
<th>Date</th>
<th>Examiner</th>
</tr>
</thead>
<tbody>
<tr>
<td>719</td>
<td>310, 313</td>
<td>10/22/2012</td>
<td>VN</td>
</tr>
<tr>
<td>718</td>
<td>100</td>
<td>10/22/2012</td>
<td>VN</td>
</tr>
</tbody>
</table>

**SEARCHED**

<table>
<thead>
<tr>
<th>Class</th>
<th>Subclass</th>
<th>Date</th>
<th>Examiner</th>
</tr>
</thead>
<tbody>
<tr>
<td>719</td>
<td>310, 313</td>
<td>10/22/2012</td>
<td>VN</td>
</tr>
<tr>
<td>718</td>
<td>100</td>
<td>10/22/2012</td>
<td>VN</td>
</tr>
</tbody>
</table>

**SEARCH NOTES**

<table>
<thead>
<tr>
<th>Search Notes</th>
<th>Date</th>
<th>Examiner</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVENTOR NAME SEARCH IN PALM AND WEST</td>
<td>10/22/2012</td>
<td>VN</td>
</tr>
<tr>
<td>INTERFERENCE SEARCH</td>
<td>10/22/2012</td>
<td>VN</td>
</tr>
<tr>
<td>WEST UPDATED SEARCH --- SEE SEARCH HISTORY</td>
<td>10/22/2012</td>
<td>VN</td>
</tr>
<tr>
<td>NPL (GOOGLE SCHOLAR) SEARCH</td>
<td>10/22/2012</td>
<td>VN</td>
</tr>
</tbody>
</table>

**INTERFERENCE SEARCH**

<table>
<thead>
<tr>
<th>Class</th>
<th>Subclass</th>
<th>Date</th>
<th>Examiner</th>
</tr>
</thead>
<tbody>
<tr>
<td>718</td>
<td>100</td>
<td>10/22/2012</td>
<td>VN</td>
</tr>
<tr>
<td>719</td>
<td>313</td>
<td>10/22/2012</td>
<td>VN</td>
</tr>
</tbody>
</table>
(pauseXlet or destroyXlet) and java

[CITATION] Design and Implementation of an MHP Demonstration System
W. Wu, H. Han
Related articles All 2 versions Import into BibTeX More

(PDF) Digital television application manager
C. Peng, E. Kuppuswamy - IEEE International Conference on Multimedia and ... - 2001 - Ieee
... The Xlet and XletContext were designed and implemented as interfaces as specified in
DVB-MHP [3] and Java TV API [4]. The interfaces defined the protocols of behavior that can be
implemented by ... The methods include initXlet(), startXlet(), pauseXlet(), and destroyXlet(). ...
Cited by 25 Related articles All 3 versions Import into BibTeX More

Method and system for providing services
... At any time, the application manager may deactivate the Xlet by invoking the pauseXlet() method,
at which the manager may also destroy the Xlet at any time by invoking the destroyXlet() method,
Which ... of the Xlet application model, the model has been included in Java 2 Platform ...
Related articles All 5 versions Import into BibTeX More

Method for managing java applications
... is received, the application manager pauses the root java application using pauseXlet() function ... 
the second child java application is received from the second child java application, the application 
manager kills the second child java application using destroyXlet() function ...
All 2 versions Import into BibTeX More

Department of information and computer science Waseda University
C. Peng, E. Kuppuswamy - 2001 - dl.ac.ir/computersoociety.org
... The Xlet and XletContext were designed and implemented as interfaces as specified in
DVB-MHP [3] and Java TV API [4]. The interfaces defined the protocols of behavior that can be
implemented by ... The methods include initXlet(), startXlet(), pauseXlet(), and destroyXlet() ...
Related articles All 3 versions Import into BibTeX More

Image forming apparatus, information processing apparatus, information processing method,
information processing program and storage medium
... Sheet 10 of 21 US 2007/0008583 A1 FIG.11 T install () ( INSTALLED update uninstall resolve ( 
UNINSTALLED) ) (uninstall ( STARTING ) ( STOPPING ) ( ACTIVE ) FIG.12 c T lev 1 new Loaded >
HntXlet: destroyXlet Paused +) destroyXlet startXlet pauseXlet ( Destroyed J ... 
Related articles All 3 versions Import into BibTeX More

[doc] STORAGE MANAGER SYSTEM FOR DVB TERMINALS
J. Coomans, A. Lucas, K. Klimotschiev, M. Akhtar - 2001 - cs.fsrnu.ac.uk
... The state of the Java Xlet may be transformed to any of the flowing states. initXlet;
StartXlet; PauseXlet; DestroyXlet. The Application Manager switches between Java

http://scholar.google.com/scholar?q=%28pauseXlet%29+or+%28destroyXlet%29+and++java+&hl=en&as_sdt=0%2C0&btnG=10/22/2012 10:38:16 AM
Xlets by pausing and resuming execution of Java Xlets. The ...
Cited by 1  Related articles  Import into BibTeX  More »

Praktikum für Nachrichtentechnik
MHPA mit Java - 81.G.81.66.de
... ausgeführt) wird die Methode pauseXlet() des Xlets aufgerufen. In dieser Methode soll das Xlet ...
Wenn das Xlet endgültig beendet werden soll, wird die Methode destroyXlet() aufgerufen. Das Xlet ...
Zustandswechsel reagieren, und die beschriebenen Java-Methoden bereitstellen. ...
Related articles  Import into BibTeX  More »

Aplicações para TV Digital Interativa
FPGR São Paulo, AHE de Senna, G Lemos ... - dis.sebrae.com.br
... Uma aplicação para TV Digital é construída utilizando aplicações Java chamadas Xlets ... é ...
chamado apenas uma vez; o startXlet: o estado do Xlet é modificado para Ativo e sua execução ...
force; o pauseXlet: o estado ... o destroyXlet: muda o estado do ciclo de vida para Destruido ...
All 7 versions  Import into BibTeX  More »

OddCi: on-demand distributed computing infrastructure
R Costa, P Brasilheiro, D Marques ... - dl.acm.org
... been introduced by Sun in the JavaTV specification and adopted as the Java application format ...
the execution of the Xlet, the application manager may call the pauseXlet() and startXlet ... At the ...
end of the Xlet’s life time, the application manager calls the destroyXlet() method, which ...
Cited by 7  Related articles  All 6 versions  Import into BibTeX  More »

ELECTRONIC APPARATUS, INFORMATION PROCESSING METHOD, AND RECORDING MEDIUM STORING INFORMATION PROCESSING PROGRAM
... 15, 2011 Sheet 7 of 11 US 2011/0226600 A1 WSDL FILE <?xml version="1.0"?> 231d <definitions > ...
SDK APPLICATION < > initXlet() t ."_"_es starXlet() <types> pauseXlet() if ... <message> g "finalpdf " <portType> - > portType> <binding> ...
All 2 versions  Import into BibTeX  More »

Java TV: Visão Geral
R da Silva Ogilvie, JR Santos ... - infovis.ucpea.che.br
... public void pauseXlet(); public void destroyXlet(boolean unconditional) throws XletStateChangeException; ) Listagem 1 - Construtor da classe JTable Em relação à interface ...
gráfica, a Java TV utiliza componentes AWT definidos pelo perfil J2ME onde está sendo ...
Related articles  All 13 versions  Import into BibTeX  More »

Method for obtaining context of corresponding Xlet while playing BD- -1 title
... (4) During the execution of the Xlet, the application manager may call the pauseXlet() method ...
(5) At the end of the Xlet’s life, the application manager will call the destroyXlet() method, which ...
will cause ... It’s important to remember that an Xlet is not a standard Java application ...
Related articles  All 5 versions  Import into BibTeX  More »

CITATION Beginning Java ME Platform
R Rushby ... - 2008 - Apress.
Cited by 11  Related articles  All 14 versions  Import into BibTeX  More »

MHP: OSGi convergence: a new model for open residential gateways
AC Viana, RP Diaz Pascorbo, MF Cabra ... - Software: Practice ... - Wiley Online Library
... ACTIVE Xlet.intXlet Xlet destroyXlet Xlet Xlet destroyXlet Xlet startXlet Xlet pauseXlet ...
INSTALLED ... When the initial Java class of an Xlet is loaded and instantiated (either from the ...
transport stream or locally from the STB), it enters the LOADED state. ...

http://scholar.google.com/scholar?q=%28pauseXlet%20or%20destroyXlet%29%20and%20java%20&hl=en&as_sdt=0%2C47[10/22/2012 10:38:16 AM]
... JAVA TV - Codificando public void startXlet() throws XletStateException{ scene
setVisible(true); X le t)] scene.setVisible(true); } public void pauseXlet() { } public void
destroyXlet(boolean condicoes) throws XletStateException{ contexto.notifyDestroyed(); } ...
Related articles All 5 versions Import into BibTeX More »

Datacasting e Desenvolvimento de Serviços e Aplicações para TV Digital Interativa
V. Becker, C. Pacheco, C. Morais, ... - MINAS, P. Web e ..., 2005 - tvdi.inf.br
... Dentre estas, destacam-se HAVI e DAVIC. Além disso, simultaneamente houve uma iniciativa
da SUN em propor uma biblioteca Java que facilitasse a adoção desse padrão em ambientes
de TV digital. ... Figura 1.5: Xlets, Gerente de Aplicações e máquina virtual Java. ...
Cited by 2 Related articles All 4 versions Import into BibTeX More »

Experiências no desenvolvimento de aplicações para Televisão Digital Interativa
P. Jucá, U de Lixenda - III Fórum de Oportunidades em Televisão Digital ..., 2005 - cesar.org.br
... Aplicações para televisão digital são desenvolvidas geralmente utilizando a linguagem de
programação Java com bibliotecas de componentes ... possa controlar o ciclo de vida, cada
aplicação deve implementar quatro métodos: initXlet, startXlet, pauseXlet e destroyXlet. ...
Cited by 3 Related articles All 4 versions Import into BibTeX More »

Sviluppo di un’applicazione per un palinsesto televisivo in tecnologia Java TV
C. Pinho, L. D’Barsa
Import into BibTeX More »

Digital-TV genom DVB-T: Översikt över lämpliga överföringstekniker för att kunna mottaga och
styr strömmande medija från en javaapplikation på en specifik.
H. Norström, B. Bergström, H. i Gåve - norstrom.info
... En Xlet innehåller ingen metod som i traditionell java utan liknar mer en java applet i den
mening att klassen innehåller metoder för att kontrollera applet livscykel. ... Andra metoder är
pauseXlet() och destroyXlet och syftet med dessa hör inte till namnet [REF03]. ...
Related articles All 2 versions Import into BibTeX More »

Comparação entre o Desenvolvimento de Aplicações MHP e Open TV
P. Jucá, A. SOUZA, F. DIAImports, C. FERREZ - SBRCG009, Curitiba, 2005 - libdoc.ufmg.br
... das encontradas nas aplicações Java para celular os midlets. Todavia, foram inseridas
peculiaridades inerentes apenas ao ambiente de TV Digital. Assim tem-se o xlet, o qual é
simplesmente uma interface que define os métodos: initXlet, startXlet, pauseXlet, destroyXlet. ...
Cited by 2 Related articles All 4 versions Import into BibTeX More »

A distribuição de aplicativos em dispositivos móveis: O caso Java-RMI
A BORGES PONTES,... de Computação em ..., 2010 - lseam-pa.edu.br
... ANTONIO PEREIRA DA SILVA JÚNIOR A distribuição de aplicativos em dispositivos móveis:
O caso Java-RMI BELÉM 2006 Page 2. ... ALLAN BORGES PONTES ANTONIO PEREIRA DA SILVA
JÚNIOR A distribuição de aplicativos em dispositivos móveis: O caso Java-RMI ...
Import into BibTeX More »

Reengineering a PC-based system into the mobile device product line
W Zhang, S Jansebaek, N Loueugan,... Workshop on Principles ..., 2003 - heesnongeiee.org
... The main class in an applet extends java.applet.Applet and main class defines four life-cycle
notification methods: init ... implements the javax.microedition.xlet.Xlet interface, which declares
four life-cycle notification methods: initXlet(), startXlet(), pauseXlet(), and destroyXlet(). ...
Cited by 23 Related articles All 23 versions Import into BibTeX More »
System and method for reducing the start-up time of Mhp applications
... A destroyXlet signals the Xlet to terminate and enter the destroyed state. ... The pauseXlet signals the Xlet to stop providing service and enter the paused state. ... This execution environment is based on the use of a Java™ virtual machine and the definition of generic APIs that provide ...

All 2 versions: Import into BBTeX  More »

[PDF] JavaTV™
MVH Arndtö, E Santos - tvid.int.br
... Discontinued em 1999 Page 55. JavaTV™ - Java TV é uma API desenvolvida pela Sun Microsystems para prover acesso e ... throws XletStateChangeException; public void pauseXlet();
public void destroyXlet(boolean unconditional) throws XletStateChangeException; } ...
All 2 versions: Import into BBTeX  More »

Utöverking av tjänsteportal för MHP-plattformen
O Bergström - 2007 - fag.diva-portal.org
... det avbryts för att ligga i bakgrunden körs pauseXlet och för när programmet avslutas körs destroyXlet som förstärker trädar och Page 7. 7 liknande. Figur 1. 1. En Xlets livscykel. Modifierad utifrån diagram Sid. 92 [1]. 2.3.1 Översikt över klasserna i MHP • java. ’ Javas standardklasser ...
Related articles: All 3 versions: Import into BBTeX  More »

[PDF] from unideb.hu
... public void startXlet()); throws XletStateChangeException; public void pauseXlet(); public void destroyXlet(boolean unconditional) throws XletStateChangeException; } Ha ezt összehasonlítsuk a java applet. Applet osztály új feladatot néhány hasonlóságot. ...
All 4 versions: Import into BBTeX  More »

Technology
M Ponzzi, H Länsimäki - Developing Services for the Wireless Internet, 2006 - Springer
... An interesting approach has been taken in Savaje OS C which is a full Java OS that is fully Java based. ... This way, however, there are fewer requirements on the client. On the other side, the API (Java applications) support is enabled by default see below and [13]. 3.4.3.3 Future ...
Related articles: All 4 versions: Import into BBTeX  More »

Xlet
... 2013 ... 2013 - dcpia.co.kr
... television receiver's software operating environment that manages Java applications. ... returns SUCCESSfully, 01 - From the ACvce State after the Xlet.pauseXlet() method ... This state is entered: When the destroyXlet() method for the Xlet returns rSeULCeGaessedtawul. ...
Import into BBTeX  More »

Controlling the smart home from TV
MR Cabrera, RPD Feldendo, AEVias ... - Consumer ... - 2006 - ieeeexplore.ieee.org
... both OSGi and MHP specifications define and implement functionality as a Java application with ... Similarly to the activation order, whatever deactivation order received (Xlet.pauseXlet or bundle.stop ... It has received the finish order, Xlet.destroyXlet or bundle.uninstall, so it frees its ...
Cited by 40. Related articles: All 5 versions: Import into BBTeX  More »

[PDF] Problemi di sicurezza in DVB-T: applicazioni MHP e soluzioni
[PDF] from xp-dev.com
Interoperabilitet mellan DVB-HTML och DVB-J

Interoperability between DVB-HTML and DVB-J

... 6 version av Java, baserad på Personale Java 1.2, som är specialt framtagen för datorlac
sändningsmedier. ... En Xlet implementerar därför ett gränssnitt som definierar fyra metoder som
är det möjligt att kontrollera dess livscykeln: • initXlet(): • startXlet(): • pauseXlet(): • destroyXlet(): ...

Related articles All 5 versions Import into BibTeX More »

Ginga Game: A Framework for Game Development for the Interactive Digital Television

Ginga Game: A Framework for Game Development for the Interactive Digital Television

... for the Digital TV Applications for the Digital TV are called Xlets, just like Java applications for ...
The following are the public methods of this interface: • destroyXlet(): signals the Xlet must be ... the
paused state, which means it's ready to start providing a service: • pauseXlet(): signal the ...

Related articles All 10 versions Import into BibTeX More »

Exploiting OSGI capabilities from MHP applications

Exploiting OSGI capabilities from MHP applications

... To sum up, both OSGI and MHP specifications define an implemented functionality as a Java
application... Similarly to the activation order, whatever deactivation order received(Xlet.pauselXlet
or bundle.stop), the other one is... It has received the finish order, Xlet.destroyXlet or...

Related articles All 6 versions Import into BibTeX More »

マルチメディアという，方とコンテンツフォーマット

Multimedia--Electronic Media学会誌, 2006 - J-STAGE

... import java.awt:*; import org.havu.*; import java.awt.event:*; ...
public void destroyXlet(Boolean unconditional) throws XletStateChangeException {
    pauseXlet(); HSceneFactory.getinstance().dispose(scene); }
}

Import into BibTeX More »

Tecnologías Para O Desenvolvimento De Aplicações Educacionais Para Tv Digital

Tecnologías Para O Desenvolvimento De Aplicações Educacionais Para Tv Digital

... do método startXlet(); Que por sua vez pode voltar ao estado paused utilizando o método
pauseXlet(); E por fim vai para o estado destroyed, usando o método destroyXlet(); ... em qualquer
nível de educação, necessitando apenas ter familiaridade com a linguagem Java e pode ...

Related articles Import into BibTeX More »

Emulaciones para TV Digital--OpenMHP y Xletview

Emulaciones para TV Digital--OpenMHP y Xletview

... Digital. A Xlet equivale a um applet Java em um PC, no que diz respeito a sua estrutura. ...
Iniciado: • pauseXlet(): o estado do ciclo de vida da Xlet é modificado para Paralisado: •
destroyXlet(): muda o estado do ciclo de vida para Desstruído. ...

Related articles All 5 versions Import into BibTeX More »
VIDEO INFORMATION PLAYBACK METHOD, VIDEO INFORMATION PLAYBACK DEVICE, RECORDING MEDIUM AND VIDEO CONTENT


... || java.awt.Graphics, M ... f, T java.awt.image. I org. ci so. net. ... |...] catch (NOp|aye|EXception e) { } catch (IOEception e) { } T4 catch (inval idiatorException e) { } timer_tsk = new
TimerTaskO; public void pauseXletO() {} public void destroyXletO(boolean arg0 ...
All 2 versions Import into BibTeX More»

Software evolution with XCVL

W Zhang, S Jarzabek, K Zhang, ... - ... Evolution with UML, ... 2005 - books.google.com

... Applet Model. The main class in an applet extends java. applet, ... microedition. xlet.
Xlet interface, which also declares four life-cycle notification methods as the Applet
model: initXletO(), startXletO(), pauseXletO(), and destroyXletO() ...
Cited by 1 Related articles All 10 versions Import into BibTeX More»

Ambiente declarativo para sistemas que implementam o GEM

RF Rodrigues - Rio de Janeiro, 2007 - telemidia.puc-rio.br

... 2. Gem. 3. NCL. 4. Java 5. Ginga 6. TV Interativa. I. Soares, Luiz Fernando ... conteúdo declarativo
produzido para o Sistema Brasileiro de TV Digital Palavras-chave GEM, NCL, JAVA, Ginga, TV
Digital Interativa PUC-Rio - Certificação Digital Nº 0511036/CA Page 7. Abstract ...
Cited by 4 Related articles All 8 versions Import into BibTeX More»

Broadcast signal processing apparatus and control method therefor

C Lee - US Patent 6,015,578, 2001 - Google Patents

... Such an application program can be provided as a Java application defined in a specification
associated with ... to the Destroyed state 54 at any time by calling, for example, a destroyXlet function. ...
from the Active state 53 to the Paused state 52 by calling the pauseXlet function so ...
Related articles All 5 versions Import into BibTeX More»

SWF2MHP – Konvertierung des Streamingformates auf die Multimedia Home Platform

T Parnell - webserver.de.fh-furtwangen.de

... verwendet. Bei der Umsetzung der MHP Applikation wird also der Flash Prototyp
in Java „nachgebaut“. Die Entwicklung ... des Entstehungsprozesses. Da die eigentliche
MHP Anwendung komplett neu in Java erstellt werden muss ...
Related articles All 7 versions Import into BibTeX More»

Xlet Resource Estimation

C Köbert, MGB Celotti-Trokser - 2004 - Citeseer

... they implement a specific Java interface, ... interface Xlet { 2 3 public void initXlet(XletContext ctx)
4 throws XletStateChangeException; 5 6 public void startXletO() throws XletStateChangeException;
7 8 public void pauseXletO(); 9 10 public void destroyXletO (boolean unconditional ...
Related articles All 2 versions Import into BibTeX More»

T-MAESTRO and its authoring tool: using adaptation to integrate entertainment into personalized t-
learning

M Ray-López, RP Díaz-Redondo... - Multimedia Tools and ..., 2008 - Springer

into personalized t-learning Marta Ray-López-Rebeca P. Díaz ...

http://scholar.google.com/scholar?q=%28pauseXlet%29+or+%28destroyXlet%29+and+java+&btnG=&hl=en&as_sdt=0%2C0[10/22/2012 10:38:16 AM]
As for the SDK for exclusive use, there are provided "CSDK", which is used for developing the application 131 according to C language and "JSDK", which is used for developing the applications according to Java (registered trademark). ... Related articles Import into BibTeX More →

Programación en srt para Blu-ray

R. Perea Beiland - 2011 - upcomms.upc.edu

... de televisión. No siguen el modelo convencional de las aplicaciones Java en el que cada ... public void pauseXlet() { public void destroyXlet(boolean unconditional) throws XletStateChangeException; { ... All 3 versions Import into BibTeX More →

IntegraTV: um portal para aplicações colaborativas em TV Digital Interativa utilizando a plataforma MHP

J.A. Andrade - 2006 - dais.ufsc.br

... 31.7.2 Interfaces com o usuário produzidas em Java .... 31 ... VSB – Vestigial Side Band, método de modulação de sinal. Widgets – Objeto gráfico genérico existente em interface gráficas com o usuário em Java (botão, listas, etc...). ... Cited by 9 Related articles All 7 versions Import into BibTeX More →

Ontwerp van een intelligente EPG voor het MHP-platform

S. Hennesbe - ibu.ugent.be

... IMDb Internet Movie Database IP Internet Protocol JMF Java Multimedia Framework LE Live Events Page 13 ... en andere zetelen, MHP is een volledig open Java gebaseerd middleware platform. De specificaties en de APIs kunnen gratis van het internet gedownload worden. ... Related articles All 2 versions Import into BibTeX More →

Digital broadcast system receiving apparatus and transmitting apparatus


... 31: 32; 33: 34: 35: import java.tv.xlet.; import java.awt.; import ... public class App-abl extends Component implements Xlet { public void destroyXlet (boolean unconditional ... scene.requestFocus(); scene.repaint(); } -> catch(InterruptedException e) { } } > public void pauseXlet () {} public void ... Related articles All 5 versions Import into BibTeX More →

REALIZZAZIONE DI UN FRAMEWORK PER LO SVILUPPO DI APPLICAZIONI INTERATTIVE NEL DIGITALE TERRESTRE

G. Ventre, G. Impressa, G. Sforzi - terraeuils.org

... II CAPITOLO II - 54 - 2 LE TECNOLOGIE UTILIZZATE - 54 - 2.1 JAVA - 55 - 2.1.1 Altri aspetti di interesse - 57 - 2.1.2 JXME - 58 - 24 - TABELLA 2.1 - CORRISPONDENZA TRA I TIPI DI SOAP E GLI OSSETTI JAVA - 67 - TABELLA 2.2 - QUANTIFICATORI NELLE REGEXP - 93 - ... Related articles Import into BibTeX More →

Concepción e desenvolvimento de aplicações interativas para televisão digital

V. Becker - 2006 - repositório.ufsc.br

Page 1. UNIVERSIDADE FEDERAL DE SANTA CATARINA PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA E GESTÃO DO CONHECIMENTO VALDEOIR BECKER CONCEPÇÃO E DESENVOLVIMENTO DE APlicações INTERATIVAS PARA TELEVISÃO DIGITAL ...
InteractTV
JA Andrade - 2006 - repositorio.ufscar.br
... 31 3.7.2 Interfaces com o usuário produzidas em Java, ... 31 ... VSB - Vestigial Side Band, método de modulação de sinal. Widgets – Objeto gráfico genérico existente em interface gráficas com o usuário em Java (botão, listas, etc...). ...
Related articles Import into BibTeX More →

[F] Fachhochschule München
CIM Probst - 2014 - it.de
... Die Implementierung des Protokolls wurde in Java realisiert. ... Die Umwandlung der Metadaten und die Übertragung zum PocketPC sollen auf der Seite der Settopbox realisiert und in JAVA implementiert werden. Die Implementierung wird in Kapitel 5 erläutert. ...
Related articles Import into BibTeX More →

[F] RTXlet: Uma Abordagem de Tempo Real para Aplicações de TV Digital baseadas em Xlets
FHs Neto - das.ufscar.br
... Co-orientador: Rômulo Silva de Oliveira, Dr. ‘Área de Concentração: Automação e Sistemas. Palavras-chave: Xlet, Real-Time Specification for Java (RTSJ), Escalonamento Adapt- ... real-time) através da utilização da Real-Time Specification for Java (RTSJ) na implode. ...
Related articles All 2 versions Import into BibTeX More →

RTXlet
R. Harke, Neto - 2006 - repositorio.ufscar.br
... Co-orientador: Rômulo Silva de Oliveira, Dr. ‘Área de Concentração: Automação e Sistemas. Palavras-chave: Xlet, Real-Time Specification for Java (RTSJ), Escalonamento Adapt- ... real-time) através da utilização da Real-Time Specification for Java (RTSJ) na implode. ...
Related articles Import into BibTeX More →

[F] Análise de Middleware no Contexto de Sistemas Terrestres de Televisão Digital
LDeS Stempel - 2006 - infoicencia.info
... No am- biente procedural a linguagem utilizada é Java [9]. ... O Ambiente Procedural O ambiente procedural do DASE faz uso de variáveis APLs em Java, tais como HAVI UI, Java TV e pJava, para a implementação de aplicações de TV interativa. Neste contexto o com: ...
Cited by 2 Related articles Import into BibTeX More →

Projeto de um Framework de Desenvolvimento de Interfaces Gráficas para TV Digital
A.F. Silva, N.P.M de Oliveira - CEP - monografias.cef.unb.br
... 45 4.1.1 Bibliotecas Gráficas para Java Desktop. . . . 46 ... GEM (continuação) [5]. . . . 40 4.1.1 Classes do pacote java.awt disponíveis para interoperabil- dade entre aplicações HAVI [8], . . . 52 Page 12. Capítulo 1 Introdução ...
Related articles Import into BibTeX More →

[F] HAVI-käyttöliittymäkomponenttien käyttö digitaalisen television MHP-sovellusten käyttöliittymissä
T. Järvinen - cc usta.fi
... 2.5.1. Nätiviivihelmi esto.... 12 2.5.2. Java-virtuaallitekoon.... 13 2.5.3. DAVIC.... 13 2.5.4. JMF (Java Media Framework) .... ...
All 3 versions Import into BibTeX More →

[F] Fakultät für informatik
J Heß - uni-siegen.de
... HTPC Home Theatre PC HW Hardware IDR Integrated Digital Receiver IPTV Internet Protocol TV IRT Institut für Rundfunktechnik ISO International Organization for Standardization ITU International Telecommunication Union IPTV Interaktives TV JMF Java Media Framework ...

Detachable java applets
... new frame 720 Add components back to application 730 Close 'kill' detached window ... it must decompose Java code into processor/platform native code, the Java applet and ... Thus, the application retains control of concurrently terminating the applet when terminating itself even ...
Cited by 36 Related articles All 2 versions Import into BibTeX More +

psj] JavaThreads
DJ Berg - Whitepapers Sun Microsystems Computer Corporation, 1996 - matheso.emory.edu
... The system won't bite you. You can't destroy anything by working on the system ... ] ] Match any characters enclosed, jobs. Show active jobs. ] ] Execute in subshell, kill, Terminate running jobs. " "
Substitute output of enclosed command, newgrp, Change to a new group ...
Related articles: All 2 versions Import into BibTeX More +

pm] Vanguard Assemblages: New Media and the Enthrumme
PJ McHenry - 2005 - usimages.uml.edu
... triggered simultaneously. Tearing the piece, even if the avant-gardist attempts to 'kill' ...
a new object to take its place—do not need to destroy their objects; the logic of ...
putting—only the meta-avant-garde is without a terminal stopping point ...
Cited by 1 Related articles All 6 versions Import into BibTeX More +

Reverse engineering of mobile application lifecycles
D Franke, C Ekstrom, S Kozwicki, .. (ICCRE), 2011 ; Bh... 2011 - icemobile.pace.org
... in the paused or stopped stale, the Android system is able to kill it along with ... not running has either not been launched or has been running and was terminated ... Active AMS. Start / startApp() AMS. Pausa / pauseApp() AMS. Destroy / destoryApp() MIDlet. notifyDestroyed() MIDlet ...
Cited by 3 Related articles All 4 versions Import into BibTeX More +

pm] Angry Birds Solving Agent
Z Li; Hastan - cs.anu.edu.au
... game consists of different levels or stages in which the Aim is to use different birds as the tool to destroy the structure and kill the pigs. ... Although it can be proved that the procedure will always terminate, the k-means algorithm does not necessarily find the most optimal ...
Related articles: All 2 versions Import into BibTeX More +

Risk assessment methodology for AIX-based computer systems
... the `kill` command, which is used to terminate processes. The `kill` command can be used to terminate processes by sending them a signal. The `kill` command can be used in conjunction with the `kill -9` command to force a process to terminate, even if it is not responding.

Cited by 4 Related articles: All 2 versions Import into BibTeX More ▼

[PDF] A Short Introduction to HTK – Graphical User Interfaces for Haskell

... window manager does not provide means to close a running application, we will have to use the `kill` or `kill -9` command to terminate the application gracefully. All GUI elements are instances of this class, and calling the `destroy` function will remove them from the application.

Cited by 1 Related articles: Import into BibTeX More ▼

[Citation] Android Processes and Threads made easy

C: Stange - Android made easy
All 3 versions Import into BibTeX More ▼

Automated risk assessment tool for AIX-based computer systems


... this patent is extended or adjusted under 35 USC 151(b) by 0 days. This patent is subject to a terminal disclaimer. (21) Appl. No.: 09/399,192 (22) Filed: Sep. 2, 1999 (51) Int. Cl. G06F 11/00 (52) US CI: 714/47; 714/37; 714 ...

Cited by 18 Related articles: All 2 versions Import into BibTeX More ▼

[PDF] Writing MIDP Games

S. Mason - ... Available from developer, symantec.com/... 2006 - mobileintelligence.com.au

... MIDlet is executing on different threads – so it is possible that the OS may direct the AMS to destroy the MIDlet. The easiest way to do this is to use the `java` utility class that comes with MidP. ... `kill` and `kill -9` on any key press protected void keyPressed(int keyCode){ closeSplash ...}

Cited by 2 Related articles: All 4 versions Import into BibTeX More ▼

[Book] Worshipping Siva and Buddha: The Temple Art of East Java

AR. Klauser, MJ. Klauser, I. Klauser - 2005 - google.com

Page 1. Worshipping Siva and Buddha THE TEMPLE ART OF EAST JAVA Page 2. The Temple Art of East Java, a study of the temples created in East Java between the tenth and sixteenth centuries, fills an important scholarly lacuna. ...

Cited by 6 Related articles: All 3 versions Import into BibTeX More ▼

[Book] Essential System Administration: Tools and Techniques for Linux and Unix Administration

K. Pratik - 2002 - books.google.com

... About Serial Lines Specifying Terminal Characteristics Adding a New Serial Device Troubleshooting Terminal Problems Controlling Access to Serial Lines HP-UX and Tru64 Terminal Line Attributes The Hyfax Fax Service USB Devices Printers and the Spooling Subsystem ...

Cited by 144 Related articles: All 30 versions Import into BibTeX More ▼

[Citation] Operating Systems

R. Joshi - 2005 - DreamTheater Press

Related articles: Import into BibTeX More ▼

[PDF] Analysis of the Android Architecture

S. Balakir: - Carnegie Mellon University, 2015 - csibcs.cs.mcmaster.ca

... The threads used on application level are standard Java threads running in the Dalvik VM. ... After running this hook, the system is allowed to kill the application at any time. ... onStop() The activity is no longer visible, the process type is set to background and the application may be ...

Cited by 5 Related articles: Import into BibTeX More ▼

[PDF] Using Hierarchical Scheduling to Support Soft Real-Time Applications in General-Purpose Operating Systems

J. Bigham - 2001 - www.eecs.berkeley.edu

Related articles: Import into BibTeX More ▼
Background and Implementation-Level Design Decisions. 79 7.3 Example scheduling hierarchy for a terminal server... in a general-purpose operating system in combination with the scheduler for a user-level thread package, the scheduler for threads in a Java virtual machine...

Reference Mwap
J May - 1998 - ibroolz.google.com
... MTERM windows, which resemble "DOS boxes" in an MS Windows environment, are for terminal emulation, and may not have descendants... If a modified attribute has a default value, the use of the M KILL command on the attributes SSVN node reference will cause the value to...

An Introduction to User Interfaces for Computer Science Students
W Crowson - waterloo.ca
... My approach to teaching this material, probably because of my background in experimental psychology, is... most unadorned for are chosen, with the presentation relying strongly on C/IXlib and Java... In these interfaces the user types at a terminal - a teltype or a glass teletype...

Learning the Unix Operating System: A Concise Guide for the New User
J Peck, G Todino, J String - 2001 - books.google.com
... Started / Working in the Unix Environment 1 Syntax of Unix Command Lines // Types of Commands 14 The Unresponsive Terminal 14 2... 104 Electronic Mail 108 Usenet News 118 Interactive Chat 123 7. Multitasking 130 Running a Command in the Background 131 Checking on...

Concurrency in Android
J Parviainen - 2012 - theses.17.iki.fi
... passed to ExecutorService, which basically is a service for handling the execution, termination and tracking of asynchronous tasks... removed by built-in garbage collector which is expected to destroy and clean the idle... keeping in the stack and which ones to kill...

More About This Macintosh
J Ostrowick - 2010 - books.google.com
... commands, reader exercises, and conguration of contents appear in Courier font when they
represent interactive terminal sessions. ... The UNIX-based Rhapsody OS would continue to run in the background, but the Classic Mac OS would come to the foreground.

[PDF] Foundations of GTK+ development
A Krause - 2007 - Agrees
Cited by 34 Related articles All 7 versions Import into BibTeX More »

[PDF] SicStus Prolog user's manual
... 8.1 Multiple SicStus Runtimes in Java... 351 8.2 Multiple SicStus Runtimes
in... 9.4.3 Placement of Cuts ... 340 9.4.4 Terminating a Backtracking ... 460
10.13.6.13 descendant_of(2);... 461 10.13.6.14 destroy!1;... 462 ...
Cited by 270 Related articles All 40 versions Import into BibTeX More »

[PDF] An electronic chalkboard for classroom and distance teaching
L. Knipping - 2005 - page.math.tu-berlin.de
... It had custom-built multimedia teaching terminal stations connected to the mainframe. ...
Interaction is usually limited to online multiple-choice tests and possibly a few interactive experiments such as Java Applets or Flash animations. ...
Cited by 23 Related articles All 9 versions Import into BibTeX More »

[PDF] Unix Tools
P. Lb - 128.232.5.23
... emacs grep -l some_variable "Java ... 6 Exit codes and conditional execution When commands terminate they exit with an integer value known as a exit code. ... This function can be used to send a kill signal to the named process(es): ...
Related articles All 9 versions Import into BibTeX More »

[PDF] Unix Tools
P. Lb - 128.232.5.23
... emacs grep -l some_variable "Java Note that grep has its own idea of what a "word" is, and so in some ... 6 Exit codes and conditional execution When commands terminate they exit with an integer value known as a ... executed foreground command can be accessed via the
Related articles All 10 versions Import into BibTeX More »

[BOOK] A practical guide to UNIX for Mac OS X users
M. S. Kieh, P. Seebach - 2006 - books.google.com
... Buffer 156 Line Length and File Size 157 Windows 157 File Locks 157 Abnormal Termination
of an ... Inserting Special Characters 219 Global Buffer Commands 220 Files 222 Buffers 224
Windows 225 Foreground Shell Commands 227 Background Shell Commands ...
All 6 versions Import into BibTeX More »

[BOOK] From Still to Motion: A photographer's guide to creating video with your DSLR
J. East, R. Carter, M. Gilsbalka, R. Harrington - 2010 - books.google.com
... Rod Harlan Kimi Heil Ben Howard Kayrin Johnson Staff of Jammin' Java Ben Kozuch Bob Kriest
Brian Kriest Orlando Luna Lynda.com John Lyle Stu Masschiltz Adam Marion Joe McNally Nikon
National Association of Photoshop Professionals Jason Oder Barbara Parker ...
All 2 versions Import into BibTeX More »

[BOOK] Advanced UNIX programming
M. Fechland - 2004 - books.google.com
... are nine chapters: Fundamental Concepts, Basic File I/O, Advanced File I/O, Terminal I/O ... I've provided interfaces in Appendices B and C so you can program in C++, Java, or Jython ... caring about the contents of files), and because an invalid directory could easily destroy an entire ...
Cited by 188 Related articles All 17 versions Import into BibTeX More »

[PDF] from ucl.edu
Essentially, our design consists of a Java applet that will be considered standalone and will not be run on a server environment. The rest of this document further details our design. We begin by stating our motivation and giving a brief background on carpal tunnel syndrome.

Related articles: Import into BibTeX More »

---

My background is in Java from back before it was even called that. From the beginning, I was very interested in embedded development as a way to program various devices that surround us in everyday life. Because ...

Cited by 22 Related articles: All 2 versions Import into BibTeX More »

---

Scheduling Commands to run in the background... 77... Also try CTRL-D: Input Logs out of a terminal, also try CTRL-D: echo A little command ... emacs /home/ted/mywork java

/home/test/java In an attempt to execute emacs on the above two ...

Cited by 1 Related articles: All 1291 versions Import into BibTeX More »

---

Women in Indian Literature

S Bharati - 2016 - books.google.com


Related articles: Import into BibTeX More »

---

Principles of network and system administration

M Burgess - 2004 - books.google.com

... 374 9.10 Samba ... 378 9.11 The printer service ... 379 9.12 Java webboard enterprise services ... 382 10 Network-level services 391 10.1 The Internet ... 392 ...

Cited by 47 Related articles: All 55 versions Import into BibTeX More »

---

Making Linux work: essential tips & techniques

A Danner - 2002 - books.google.com

... Moving a Foreground Application to the Background 709 Tip 23 7: Moving a Background Command to the Foreground 709 Tip 23.8: Bringing a Stopped Application Back to the Foreground 709 Tip ... Linux PCs 232 Tip 4.24: Using a Linux System as an X Terminal By Requesting ...

Cited by 1 Related articles: All 2 versions Import into BibTeX More »

---

Dearborn Independent Magazine June 1926-September 1926

H Ford - 2003 - books.google.com


All 2 versions Import into BibTeX More »

---

Operating systems

JL Dering, WJT Testy, ED Loomis - 2003 - books.google.com

... The extent of the string may be specified in the instruction, but in many cases it is determined by a count field or by a termination code in the operand itself. Saul Rosen ... Fork, suspend, resume, join, signal, exit, kill ... communication 8 Memory Address, segment Create, destroy, map ...

Cited by 13 Related articles: All 22 versions Import into BibTeX More »

---

Minnesota plant life

C MacDicken - 1899 - books.google.com

... 56. Bed of ferns. Sensitive fern in middle of foreground. After photograph by Williams 165 F1G. 57. Cliffs-brake. After Britton and Brown 166 F1G. 58. The interrupted fern (in background) and
shield-terms (in foreground). After photograph by Williams 167. Fig. 59, ...
Cited by 4 Related articles All 3 versions Import into BibTeX More »

8. Greta Garbo
L. FISCHER - Giants of Modesty, Movie Stars of the 1920s. 2010 - books.google.com
... Though Irene and André temporarily terminate their affair out of a sense of propriety, another man ... Afraid he would kill Pierre, Irene took a gun from Guany's desk drawer and fired upon him. ... Eventually, Robledo saves the day and vows that this woman shall destroy no one else...
Import into BibTeX More »

[PDF] from 12.165.240.191

[PDF] from 118.98.137.181

... 33 Compaq Alpha processor, 904 compatibility, hardware, 32-33 compilation in Java. 715 of ...
Depth directive, 277 Depth option, 150 Desk Guide applet, 148, 149 Desktop Background applet, 196 ... 199-200 virtual, 148 in X Windows application, 102-105, 103 Destroy option, 150 ...
Cited by 5 Related articles All 7 versions Import into BibTeX More »

[PDF] from nps.edu

... 43 Chapter 4: Conjuring and Embellishing an Android App ... 73 Book II: Android Background Material ... 109 Chapter 1: Using the Eclipse Workbench ... 111 Chapter 2: It's Java! ...
All 4 versions Import into BibTeX More »

Multi-Agent Task Negotiation Among UAVs to Defend Against Swarm Attacks
M. Day - 2012 - caltech.edu
... This is a Java based toolkit, which allows us to do data collection using custom Java programming. ...
Also, should a blue find its mark and destroy a red, ... resulting in reinforced blue subteams, which increases blue subteams' effective kill probabilities (see Section 3.2.4). ...
Related articles All 4 versions Import into BibTeX More »

[PDF] from tjiera.ru

... engineers who have learnt 'a bit of Java' and who wish to teach yourself enough to program Java (or C# or other, similar languages) for your own work. You might be students who are converting to information technology from a science or engineering background or you ...
Cited by 7 Related articles All 5 versions Import into BibTeX More »

[PDF] from houseofrusseell.com

Restorative justice & responsive regulation
J. Braithwaite - 2001 - books.google.com
Page 1. Restorative Justice and Responsive Regulation JOHN BRAITHWAITE Page 2. Restorative Justice & Responsive Regulation Page 3. STUDIES IN CRIME AND PUBLIC POLICY Michael Tonry and Nonual Morris, General ...
Cited by 1082 Related articles All 12 versions Import into BibTeX More »

[PDF] from houseofrusseell.com

... demo for the UNISCOPE 3000, which many have called the first commercial video terminal ...

The new hacker's dictionary
ES Raymond - 1996 - books.google.com
...
and unimaginative background for the human foreground; it often contains symbols ...

[PDF] from socratics.net

Game Development 101
M. Zeichner, R. Green - Beginning Android 4 Games Development, 2011 - Springer
... a set of connectors at each side that allow you to connect to viruses and thereby destroy them ...
Prevent in an attempt to show that one can write high-performance games in pure Java on Android ...
For each enemy you kill, you usually get some amount of money or points that you can ...
All 10 versions. Import into BibTeX. More+

[PDF] from uni.com

An electronic on-board recorder for intermodal cargo operations
A. Grouning - 2012 - gradworks.umi.com
... For example, Namoothiri and Erera applied a variant of the problem denoted pickup with time windows (PDPTW) algorithm and found that terminal congestion ... background literature. ... development platform for Java-based devices. Google has created a SDK specifically for ...
Related articles. Import into BibTeX. More+

Restorative justice: Assessing optimistic and pessimistic accounts
J. Braithwaite - Crime and Justice, 1999 - JSTOR
... Section I of this essay first seeks to conceptualize what restorative justice is against the background of these histories. Sections II-V follow the author's peculiar history of engagement with restorative pro- Page 4. 4 John Braithwaite...
Cited by 493. Related articles. All 7 versions. Import into BibTeX. More+

[PDF] from socratics.net

Getting started with android
D. Smith, J. Friesen - Android Recipes, 2011 - Springer
... The compiled Java code for an app's components is further transformed into Dalvik's DEX format. ...
NOTE: Android can kill the process hosting the activity at any time after onPause(), onStop(), or ...
For example, if the activity has a thread running in the background to download data ...
All 3 versions. Import into BibTeX. More+

[PDF] from iit.ac.in

Introduction to Operating System Design and Implementation
M. Kifer, S. Smidt - 2007 - iitr.ernet.in
... Additionally, the project generator automatically creates "java" files containing class ...
method headings for each of the assigned modules. ... In such cases, the simulator will gracefully terminate execution of the program by delivering ...
Cited by 2. Related articles. All 10 versions. Import into BibTeX. More+

[PDF] from nikde.eu

Operating Systems
Z. Tuma, I. Soileu, E. Strub, V. Babka - mst.nikde.eu
... The sections are styled as a crash course on things either known in general or outside the scope of this book, presented to familiarize the reader with the background and terminology. Needless to say, none of the things outlined here is definitive. ...
Related articles. All 3 versions. Import into BibTeX. More+

[PDF] from control.com

The eCos component writer's guide
E. Vae, J. Dillaway - 2006 - control.com
... the text may be displayed in a variety of fonts; the foreground and background color may ... The eCos approach allows the user to select that the thread kill functionality is not required ... 3. The application may or may not require the ability to create and destroy mutexes dynamically ...
Cited by 13. Related articles. All 31 versions. Import into BibTeX. More+

[PDF] from books.google.com

Android Recipes: A Problemsolution Approach
J. Friesen, D. Smith - 2011 - books.google.com
... The compiled Java code for an app's components is further transformed into Dalvik's DEX format. ...
NOTE: Android can kill the process hosting the activity at any time after onPause(), onStop(), or ...
For example, if the activity has a thread running in the background to download data ...
Cited by 3. Related articles. All 3 versions. Import into BibTeX. More+

http://scholar.google.com/scholar?q=%28kill+or+terminal+or+destroy%29+and+java+and+background+and+foreground&hl=en&as_sdt=0%2C2047[10/22/2012 10:36:48 AM]
Physical computing: sensing and controlling the physical world with computers
G O'Sullivan, T Igra - 2004 - books.google.com

Output from a Microcontroller 143 Testing with an LED 149 Testing with Terminal Software 149 ...
In Max/MSP 239 Video Tracking in Processing 240 Video Tracking in Java 242 CMUcam ...
Regardless of your background or technical experience, this book is designed to help you ...
Cited by 104 Related articles All 4 versions Import into BibTeX More

Pro Android 4
S Rokaus, L MacLean - 2012 - books.google.com

Service Components Android Media and Telephony Components Android Java Packages Taking ...
Process The Destroy Create Cycle of Activities The Destroy Create Cycle of Using the ...
HttpURLConnection Using the AndroidHttpClient Using Background Threads AsyncTask ...
Cited by 11 Related articles All 4 versions Import into BibTeX More

The Architecture of Open Source Applications
A Brown, G Watt - Lulu.com - 2011 - zedshifi-b.googlecode.com

... This is incredibly important for performance reasons. Once a call is over, the abstract channel ...
handling code that lives in the Asterisk core will invoke the ast_channel tech hangup callback ...
and then destroy the ast_channel object. 1.2.2. Dialplan Applications ...
Cited by 2 Related articles All 10 versions Import into BibTeX More

Agent Tl: Alpha Release 1.1
R Gray - Available by WWW at http://www.cs.dartmouth.edu ... 1985 - Citesser

... The server should now be running in the background as a daemon ... All error and informational ...
messages will be displayed on the terminal rather than written to the log file ...
6. To bring down the server, you need to kill all running agent processes (there will be more than two if there ...
Cited by 16 Related articles All 11 versions Import into BibTeX More

The winds of change: Climate, weather, and the destruction of civilizations
E Lindberg - 2006 - books.google.com

... sinister shift in meaning in the emphasis placed on one word: 'He'd kill us if ... teased a picture of the ...
context of these events, revealing an agent working in the background ... landowners discovered ...
that sharp-horned animals such as sheep and goats were rapidly destroy ing the ...
Cited by 67 Related articles All 4 versions Import into BibTeX More

More crops per drop
S Carter, D Vallejo - Ripe Today, 2007 - dspace.irri.org

... 30 AND DAMAGED RICE Two volcanic disasters in Indonesia's East Java Province are destroying ...
rice crops and making life tough, if not ... These would be harvested and refined for use in medicines ...
to fight diarrhea and dehydration, which kill more than a million infants and ...
Cited by 14 Related articles Import into BibTeX More

Reports of the survey: Botanical series
Natural History Survey of M provision - 1899 - books.google.com

... 150. Sumac bushes, with golden-rods in foreground and maples in background. After photograph ...
by Williams 399 F1G. 151. ... In foreground golden-rods, sunflowers and asters; in background on ...
brow of cliff, wormwood or sage-brush. After photograph by Williams ... 408 F1G. ...
All 2 versions Import into BibTeX More

The specter of genocide: mass murder in historical perspective
B Gerstley, B Rieman - 2003 - books.google.com

... raids, millions were put to death as part of deliberate Aries plans to kill them because ... the Herero ...
were any real threat, the local German military commander issued an ex termination order ... try, ...
usually with considerable success, to cover up their crimes and to destroy the evidence ...
Cited by 87 Related articles All 8 versions Import into BibTeX More

Securing Graphical User Interfaces

[PDF] from psu.edu

[PDF] from irri.org

[HTML] from google.com

http://scholar.google.com/scholar?q=%28kill%20or%20terminal%20or%20destroy%29%20and%20java%20and%20background%20and%20foreground&hl=en&as_sdt=0%2C47[10/22/2012 10:36:48 AM]
... This technique was introduced with the Bill terminal [56] and the Xerox Alto per-sonal computer ...
... cause the web browser to locally execute JavaScript code, Adobe Flash applets, or Java code. ... 
Before drawing the cursor, the GUI server saved the cursor’s background into a buffer. ...

Cited by 2. Related articles. All 4 versions. Import into BibTeX. More+

[HTML] Author Archive
2 Roth - manuel.mayaadventure.wordpress... ...
... [28] The film’s plot was fabricated about white bodyguards being ordered to kill them should they fall... In the foreground, commending them for their work is Lieutenant Colonel James G. Smith, signal officer for the First Marine Division. Hqtrs. ... Temiijn was of a noble background... ...
Related articles. All 10 versions. Import into BibTeX. More+

[PDF] from 137.222.102.8

A Sophisticated Shell Environment
H ussijn Peelikam - 2000 - 137.222.102.8
... It only has external control, eg to pause or terminate it... The ability to undo commands and recover old versions of files. • The ability to kill rogue programs cleanly. ... The ability to run untrusted programs in a protected environ- ment, as with Java’s sandbox... ...
Related articles. All 15 versions. Import into BibTeX. More+

[book] Eccentricities of the animal creation
J Timbs - 2003 - books.google.com
... Anecdote of, 257.—Great Number of Fishes, 258.—Little Fishes Eaten by Medusae, 259.—Migration of Fishes, 261.—Enormous Grampus, 262.—Bonita and Flying-fish, 263.—Lacustrine Fish of Java, 264.—Port... Bookworms, their Destructiveness, 338,337.—How to... 
Cited by 3. Related articles. All 2 versions. Import into BibTeX. More+

G. Crawford - 2002 - books.google.com
... Human Interaction versus Human-Computer Interaction 74 7 ARCHITECTURES A Simple Interactive Diagram 76 Foldback 79 Kill’Em If... 127 Cacophonous Messages 127 An Extended Bloomer 128 Discussion 134 A Special Potshot 135 Some Background 135 Meanwhile...
Cited by 122. Related articles. All 5 versions. Import into BibTeX. More+

[book] Impoundment
EP. Wright - 2007 - books.google.com
... with grenades and mines, how to kill them with knives and bayonets, and how to kill them with... for the glint of steel or the faint outline of a human form against the background of green... swirled across the black tamaric, and I saw my parents waiting for me inside the tiny terminal... 
All 2 versions. Import into BibTeX. More+

[book] Writing machines
HR. Hayles - 2002 - books.google.com
... Telling a fuller story requires these narrative chapters interrogating the author’s position, her background and experiences, and especially the com- munity of writers, theorists... Unlike Moravec, Egan does not find it necessary to destroy the original in creating the SIMULACRUM... 
Cited by 344. Related articles. All 7 versions. Import into BibTeX. More+

FO. Braynard, VH. Miller - 1991 - books.google.com
... Her terminal port in the United States was Boston... Boston, suddenly realizing that the engraving publicized a drawback of their harbor, did their best to buy up all the copies and destroy them... His fine old print showing two new Cunanders, the Europa (in the foreground) heading to...
Cited by 4. Related articles. All 4 versions. Import into BibTeX. More+

[ps] Interactive HTML
O. Hudson, CM. Parsade - 1997 - angryredplanet.com

http://scholar.google.com/scholar?q=%28kill%29%20or%20terminal%20or%20destroy%29%20and%20java%20and%20background%20and%20foreground&hl=en&as_sdt=0%202047[10/22/2012 10:36:48 AM]
... 13.2.3 The Java Platform. This allows plug-ins to remain independent not only of the hardware on which they are running, but also the Java Virtual Machine (JVM) and the Java platform. Related articles: More...

(book) Creating Android Applications: Develop and Design
C. Haseman - 2011 - books.google.com
... I assume you were already comfortable with Java before diving into this book; I'm also going to assume you're comfortable installing the Android development kits as well as Eclipse screenshots for all important commands and tasks in case you're rocking the terminal with Vim or ...
All 2 versions More...

(pdf) SETL for internet data processing
E. Bacon - 2003 - cs.nyu.edu
... 212.6.9.6 Java . . . . . . . . . . their purpose. If that algebra has been designed with feasible goals in mind, the exer- cise will converge. It is typical to ensure this termination by taking advantage of the fact that any set generated by inductive definitions (such as a power set) can...
Cited by 4 Related articles: More...

(book) Red Hat Linux system administration handbook
MF Komarudin, C. Oglesby - 2003 - books.google.com
... 12 Linux Database Software 201.12.1 MySQL and mSQL 202.12.2 Other Native Linux DMBSes 210.12.3 Summary 213 Chapter 13 Programming Languages 275.13.1 C 276.13.2 C++ 277 13.3 Perl 218 13.4 Python 219 13.5 Lisp, Scheme, and Guile 222 13.6 Java...
Cited by 1 Related articles: More...

(int) android mega pack
F. Picco, T. Venkatesh, J. Shanksy - mail.edsaha.wordpress.com
... This gave him much clearer pictures in Android; he wrapped his JMF calls using the div4Start utility. div4Start jmfinit div4Start java DiscordBroadcaster. ...
Cited by 1 Related articles: More...

(citation) Talking to the enemy: Violent extremism, sacred values, and what it means to be human
S. Anin - 2010 - ePengun
Cited by 19 Related articles: More...

(book) The Sugar Cane Industry: An historical geography from its origins to 1914
J. H. Gulliver - 2005 - books.google.com
... of Madeira as anticipated in the alvar of 1498 52 5.1 Population of the West Indian sugar colonies during the eighteenth century 114 6.1 Indian indentured laborers in the Caribbean, 1838-1917 127 6.2 The efficiency of sugar factories 140 8.1 Sugar exports of Java ...
Cited by 143 Related articles: More...

(book) Pro PHP-GTK
S. Matlock - 2006 - books.google.com
... 281 CHAPTER 15 Doing Background Work . . . . . . . . . . 302 CHAPTER 15 Doing Background Work . . . . . . . . . . 303 Progress Bars . . . . . . . . . .
All 9 versions More...

(pdf) LISP EAI4, Evaluating Configuration Guide for Red Hat Enterprise Linux on IBM hardware
R. Wedani - 2003 - publib.boulder.ibm.com
... Java and all Java-based products are trademarks of Sun Microsystems, Inc., in the United States, other ... This will DESTROY the data on this disk without prompting, use with care. ... You MAY use a computer using terminal emulation software and a null modem cable instead of a ...
Related articles: More...

(pdf) Component-based Statistical Computing
[PDF] from hu-berlin.de
... The first WWW and Java interfaces are also described there. ... Page 29. Client / Server based Statistical Computing executable (MDCOM.exe) instead of a Java executable archive (MDserv.jar) and to slightly improve performance while working with MDReX (figures 2.1 and ... Related articles: All 4 versions. Import into BbTeX More+

Geology of the upper Tilling River district, Alaska
FH Holfit - 1941 - pubs.usgs.gov
... Although the Java flows have been extensively eroded, they, unlike all the older rocks, are not ... trouble in crossing some small streams and soft places on Kill slopes, especially those places ... The limestone beds, however, are not so greatly metamorphosed as to destroy all the ... Cited by 3 Related articles. Import into BbTeX More+

Talking to the enemy: Faith, brotherhood, and the (un)making of terrorists
S Ahren - 2010 - books.google.com
... enough to pull me back from years of fieldwork in the rain forest with Maya Indians to try to understand and convey what makes humans willing to kill and die ... They are recent immigrants from different parts of Java, and some of the Christian fighters are imported from East Timor. ... Cited by 34 Related articles. All 457 versions. Import into BbTeX More+

GNU coding standards
R Stallman - 1992 - gnu.org
... 8 In Unix, realloc can destroy the storage block if it returns zero ... This allows the user to kill other processes to free up virtual memory, and then try the command again ... sourcefile linemanumber: message in an interactive program (one that is reading commands from a terminal), it is ...
Cited by 41 Related articles. All 457 versions. Import into BbTeX More+

Goldfish
ME Otsow - 2008 - books.google.com
... Later on you may want to change the decorative background behind your aquarium or simply clean the outside of the back glass. ... making them very susceptible to certain diseases. Temperature shock can even kill them directly and rather quickly. ...
All 4 versions. Import into BbTeX More+

The shame and the sorrow: Dutch-Amerindian encounters in New Netherland
D Merwin - 2006 - books.google.com
... Here, it meant, Will you kill us by setting our village ablaze and burning us alive? ... It could consume and destroy wildly: inhabitants and their houses and barns, stores of supplies, boats, animals ... Soon it was extending its reach from Java in Indonesia to the Molucases, the Mala-bar ... Cited by 14 Related articles. Import into BbTeX More+

OpenJaid for Developers
M Butcher - 2007 - books.google.com
... He has coauthored two books on advanced software development for Prentice Hall PTR and Sun Microsystems press, including HighPerformance Java Platform Computing: Threads and Networking (seehttp://jhpoc.googlecode.com) and Web Programming in Python (see http ... Cited by 3 Related articles. All 2 versions. Import into BbTeX More+

Malaria: cause and control
WS Harris - 1913 - books.google.com
... Exceedingly warm, moist seasons always brought more malaria, while a prolonged drought was said to kill the disease, as did the approach of cold weather. ... The disease is known to occur endemically in Java, Madagascar, and elsewhere at an elevation of 3000 feet. ... Cited by 7 Related articles. All 3 versions. Import into BbTeX More+

P-38 Lightning Vs Ki-61 Tony: New Guinea 1943-44
D Nilpser - 2010 - books.google.com
... experimental Ki-61s. Here, two of the prototypes (serials 6101 in the background
and 6102 in the **foreground** have their engines warmed up while a flight of JAAF Ki-49 heavy bombers fly overhead. The new prototypes were ...

Did you mean to search for: (**kill** or **terminator** destroy) and java and background and foreground
Method for managing Java applications
... YES S178 switch 1st java application in the background to foreground. ... initXletQ startXletG
startApp(child1, Bound, c2URL) pauseXlet() startXletG destroyApp(child1) destroyXletQ ... manager
AB startApp(root, Standard, s2URL) destroyXlet() destroyApp ... All 2 versions. Import into BibTeX More>

Application frameworks for mobile devices
... the paused state 305. The mobilat manager may move the mobilat between the
background state 302, foreground state 304, and the paused state 305, depending
on priorities and usage requirements. In this fashion, the mobilat ...
All 2 versions. Import into BibTeX More>

Technology
M. Ponz, P. Laamanen - Developing Services for the Wireless Internet, 2006 - Springer
... language used. Unlike WML, cHTML is a subset of HTML that leaves out coding for
JPEG images, tables, image maps, multiple character fonts and styles, background
colour or images, frames and cascading style sheets. They ...
Related articles All 4 versions. Import into BibTeX More>

Sviluppo di un'applicazione per un palinsesto televisivo in tecnologia Java TV
G. Pico, L. Diano - Import into BibTeX More>

Fachhochschule München
Dr. Prof. - 2006 - in dr
Page 1. Fachhochschule München University of Applied Sciences Fachbereich
Elektrotechnik | Datentechnik Diplomarbeit Audio-Systemtechnik PDA alle ähnliche
User-Interfaces für Interaktion mit bewegten Objekten in GMF4TV ...
Related articles. Import into BibTeX More>

Create alert

About Google Scholar All About Google Privacy & Terms Give us feedback

http://scholar.google.com/scholar?q=%28pauseXlet%29+or+%28destroyXlet%29+and+%28background+or+foreground%29&btnG=&hl=en&as_sdt=0%2C47[10/22/2012 10:36:29 AM]
**Issue Classification**

<table>
<thead>
<tr>
<th>Application/Control No.</th>
<th>Applicant(s)/Patent Under Reexamination</th>
</tr>
</thead>
<tbody>
<tr>
<td>12656872</td>
<td>BAEK ET AL.</td>
</tr>
</tbody>
</table>

**Examiner**

<table>
<thead>
<tr>
<th>Art Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>2199</td>
</tr>
</tbody>
</table>

**ORIGINAL**

<table>
<thead>
<tr>
<th>CLASS</th>
<th>SUBCLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>719</td>
<td>313</td>
</tr>
</tbody>
</table>

**INTERNATIONAL CLASSIFICATION**

<table>
<thead>
<tr>
<th>CLAIMED</th>
<th>NON-CLAIMED</th>
</tr>
</thead>
<tbody>
<tr>
<td>G 0 6 F</td>
<td>9 / 46 (2006.01.01)</td>
</tr>
</tbody>
</table>

**CROSS REFERENCE(S)**

<table>
<thead>
<tr>
<th>CLASS</th>
<th>SUBCLASS (ONE SUBCLASS PER BLOCK)</th>
</tr>
</thead>
<tbody>
<tr>
<td>718</td>
<td>100</td>
</tr>
</tbody>
</table>

**Claims renumbered in the same order as presented by applicant**

<table>
<thead>
<tr>
<th>Final</th>
<th>Original</th>
<th>Final</th>
<th>Original</th>
<th>Final</th>
<th>Original</th>
<th>Final</th>
<th>Original</th>
<th>Final</th>
<th>Original</th>
<th>Final</th>
<th>Original</th>
<th>Final</th>
<th>Original</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>17</td>
<td>17</td>
<td>2</td>
<td>2</td>
<td>18</td>
<td>18</td>
<td>3</td>
<td>3</td>
<td>19</td>
<td>19</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>20</td>
<td>20</td>
<td>5</td>
<td>5</td>
<td>21</td>
<td>21</td>
<td>6</td>
<td>6</td>
<td>22</td>
<td>22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td></td>
<td></td>
<td>8</td>
<td>8</td>
<td></td>
<td></td>
<td>9</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td></td>
<td></td>
<td>11</td>
<td>11</td>
<td></td>
<td></td>
<td>12</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>13</td>
<td></td>
<td></td>
<td>14</td>
<td>14</td>
<td></td>
<td></td>
<td>15</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**NONE**

<table>
<thead>
<tr>
<th>(Assistant Examiner)</th>
<th>(Date)</th>
<th>Total Claims Allowed:</th>
</tr>
</thead>
<tbody>
<tr>
<td>/VAN NGUYEN/</td>
<td></td>
<td>22</td>
</tr>
<tr>
<td>Primary Examiner, Art Unit 2199</td>
<td>10/22/2012</td>
<td>O.G. Print Claim(s):</td>
</tr>
<tr>
<td>(Primary Examiner)</td>
<td>(Date)</td>
<td>O.G. Print Figure:</td>
</tr>
</tbody>
</table>
**BIB DATA SHEET**

**CONFIRMATION NO. 2380**

<table>
<thead>
<tr>
<th>SERIAL NUMBER</th>
<th>FILING or 371(c) DATE</th>
<th>CLASS</th>
<th>GROUP ART UNIT</th>
<th>ATTORNEY DOCKET NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/656,872</td>
<td>02/18/2010</td>
<td>718</td>
<td>2199</td>
<td>0366.1006</td>
</tr>
</tbody>
</table>

**APPLICANTS**

Wonjung Baek, Seongnam-si, KOREA, REPUBLIC OF; 
John Kim, Seoul, KOREA, REPUBLIC OF;

**CONTINUING DATA**

**FOREIGN APPLICATIONS**

REPUBLIC OF KOREA 10-2009-0014161 02/20/2009

**IF REQUIRED, FOREIGN FILING LICENSE GRANTED**

03/08/2010

<table>
<thead>
<tr>
<th>STATE OR COUNTRY</th>
<th>SHEETS</th>
<th>TOTAL CLAIMS</th>
<th>INDEPENDENT CLAIMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>KOREA, REPUBLIC OF</td>
<td>9</td>
<td>22</td>
<td>3</td>
</tr>
</tbody>
</table>

**ADDRESS**

STEIN MCEWEN, LLP
1400 EYE STREET, NW
SUITE 300
WASHINGTON, DC 20005
UNITED STATES

**TITLE**

Method for managing java applications

**FILING FEE RECEIVED**

597

FEES: Authority has been given in Paper No._________ to charge/credit DEPOSIT ACCOUNT No._________ for following:

- All Fees
- 1.16 Fees (Filing)
- 1.17 Fees (Processing Ext. of time)
- 1.18 Fees (Issue)
- Other_________________
- Credit

BIB (Rev. 05/07).
## WEST Search History for Application 12656872

**Creation Date:** 2012-02-21:31

### Interference Searches

<table>
<thead>
<tr>
<th>Query</th>
<th>DB</th>
<th>Op.</th>
<th>Plur.</th>
<th>Thes.</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>((Baek near3 Wonjang) or (Kim near3 John)).in.</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>(((Baek near3 Wonjang) or (Kim near3 John)).in.) and java near10 chang $ near10 state $ 1</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>(((Baek near3 Wonjang) or (Kim near3 John)).in.) and Xlet</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>(((Baek near3 Wonjang) or (Kim near3 John)).in.) and kill $ near10 (java or application $ 1)</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>(((Baek near3 Wonjang) or (Kim near3 John)).in.) and (terminat $ or kill $ or stop $)</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>(((Baek near3 Wonjang) or (Kim near3 John)).in. and (terminat $ or kill $ or stop $)) and java</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>(kill $ or terminat $) and (java application $ 1) and (re-chang $ or chang $) and state $ 1).clm.</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>(kill $ or terminat $) and (java) and (re-chang $ or chang $) and state $ 1).clm.</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>((Baek near3 Wonjang) or (Kim near3 John)).in.</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>(((Baek near3 Wonjang) or (Kim near3 John)).in.) and java near10 chang $ near10 state $ 1</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>(((Baek near3 Wonjang) or (Kim near3 John)).in.) and Xlet</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>Claim</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>-------</td>
<td>------------------</td>
<td>-----</td>
<td>----</td>
<td>----------</td>
<td>------------</td>
</tr>
<tr>
<td>(((Baek near3 Wonjang) or (Kim near3 John)).in.) and kill $ near10 (java or application $ 1)</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>(((Baek near3 Wonjang) or (Kim near3 John)).in.) and (terminat $ or kill $ or stop $)</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>(((Baek near3 Wonjang) or (Kim near3 John)).in. and (terminat $ or kill $ or stop $) and java</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>((kill $ or terminat $) and (java application $ 1) and (re-chang $ or chang $) and state $ 1).clm.</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>((kill $ or terminat $) and (java) and (re-chang $ or chang $) and state $ 1).clm.</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>((Baek near3 Wonjang) or (Kim near3 John)).in.</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>(((Baek near3 Wonjang) or (Kim near3 John)).in.) and java</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>((kill $ or terminat $) and (java) and (re-chang $ or chang $ and switch $) and (background or foreground or state $ 1)).clm.</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>((kill $ or terminat $ or paus $ and unpaus $ and halt $ and suspen $) and (java) and (re-chang $ or chang $ and switch $) and (background or foreground or state $ 1)).clm.</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>(((kill $ or terminat $ or paus $ and unpaus $ and halt $ and suspen $) and (java) and (re-chang $ or chang $ and switch $) and (background or foreground or state $ 1)).clm.) not (((kill $ or terminat $) and (java) and (re-chang $ or chang $ and switch $) and (background or foreground or state $ 1)).clm.)</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>((kill $ or terminat $ or paus $ and unpaus $ and halt $ and suspen $) and (java) and (background or foreground or state $ 1)).clm.</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>(((kill $ or terminat $ or paus $ and unpaus $ and halt $ and suspen $) and (java) and (background or foreground or state $ 1)).clm.) not (((kill $ or terminat $ or paus $ and unpaus $ and halt $ and suspen $) and (java) and (re-chang $ or chang $</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
</tbody>
</table>

Interference Searches
| and switch $) and (background or foreground or state $ 1)).clm.) | PGPB, USPT, UPAD | ADJ | NO | ASSIGNEE | 10-22-2012 |
| ((kill $ or terminat $ or paus $ and unpaus $ and halt $ and suspen $ and destroy $) and (java application $ 1)).clm.) | PGPB, USPT, UPAD | ADJ | NO | ASSIGNEE | 10-22-2012 |
| (((kill $ or terminat $ or paus $ and unpaus $ and halt $ and suspen $ and destroy $) and (java application $ 1)).clm,) not (((kill $ or terminat $ or paus $ and unpaus $ and halt $ and suspen $) and (java) and (background or foreground or state $ 1)).clm. not ((kill $ or terminat $ or paus $ and unpaus $ and halt $ and suspen $) and (java) and (re-chang $ or chang $ and switch $) and (background or foreground or state $ 1)).clm.) | PGPB, USPT, UPAD | ADJ | NO | ASSIGNEE | 10-22-2012 |
| (((kill $ or terminat $ or paus $ and unpaus $ and halt $ and suspen $ and destroy $) and (background or foreground or state $ 1)).clm. | PGPB, USPT, UPAD | ADJ | NO | ASSIGNEE | 10-22-2012 |
| (((kill $ or terminat $ or paus $ and unpaus $ and halt $ and suspen $ and destroy $) and (background or foreground or state $ 1)).clm. and java | PGPB, USPT, UPAD | ADJ | NO | ASSIGNEE | 10-22-2012 |
| (((kill $ or terminat $ or paus $ and unpaus $ and halt $ and suspen $ and destroy $) and (background or foreground or state $ 1)).clm. and java) and $ Xlet | PGPB, USPT, UPAD | ADJ | NO | ASSIGNEE | 10-22-2012 |
| (((kill $ or terminat $ or paus $ and unpaus $ and halt $ and suspen $ and destroy $) and (background or foreground or state $ 1)).clm. and java) and destroyXlet $ | PGPB, USPT, UPAD | ADJ | NO | ASSIGNEE | 10-22-2012 |

### Prior Art Searches

<table>
<thead>
<tr>
<th>Query</th>
<th>DB</th>
<th>Op.</th>
<th>Plur.</th>
<th>Thes.</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>(kill$ or terminat$) near10 (java application$1)</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>(((kill$ or terminat$) near10 (java application$1) ) and (((kill$ or terminat$) near10 (java application$1)).ab.</td>
<td>PGPB, USPT, USOC, EPAB,</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>Prior Art Searches</td>
<td>JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------</td>
<td>-----</td>
<td>----</td>
<td>----------</td>
<td>------------</td>
</tr>
<tr>
<td>((kill$ or terminat$ or stop$) near10 (java application$1) ) and (re-chang$ or chang$ or rechang$) near10 state$1 near10 (java or application$1)</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>(kill$ or terminat$ or stop$) near10 (java application$1)</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>((kill$ or terminat$ or stop$) near10 (java application$1) ) and ((kill$ or terminat$ or stop$) near10 (java application$1)).ab.</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>((kill$ or terminat$ or stop$) near10 (java application$1) and ((kill$ or terminat$ or stop$) near10 (java application$1)).ab. ) not ((kill$ or terminat$) near10 (java application$1) and ((kill$ or terminat$) near10 (java application$1)).ab. )</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>((kill$ or terminat$ or stop$) near10 (java application$1) ) and re-chang$</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>(kill$ or terminat$ or stop$) near10 (java)</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>Prior Art Searches</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>((kill$ or terminate$ or stop$) near10 (java) ) and re-chang$</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>((kill$ or terminate$ or stop$) near10 (java) ) and ((kill$ or terminate$ or stop$) near10 (java)).ab.</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>((kill$ or terminate$ or stop$) near10 (java) ) and ((kill$ or terminate$ or stop$) near10 (java)).ab. ) and (state$1 near5 (application$1 or java))</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>((kill$ or terminate$ or stop$) near10 (java) ) and switch$ near10 java near10 (background or foreground)</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>((kill$ or terminate$ or stop$) near10 (java) ) and java near10 (background or foreground)</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>((kill$ or terminate$ or stop$) near10 (java) and java near10 (background or foreground) ) and xlet</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>(kill$ or terminate$) near10 (java application$1)</td>
<td>PGPB, USPT, USOC, EPAB,</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>((kill$ or terminat$) near10 (java application$1)) and ((kill$ or terminat$) near10 (java application$1)).ab.</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>---------------------------------</td>
<td>----</td>
<td>---</td>
<td>--------</td>
<td>---------</td>
</tr>
<tr>
<td>((kill$ or terminat$) near10 (java application$1)) and (re-chang$ or chang$ or rechang$) near10 state$1 near10 (java or application$1)</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>(kill$ or terminat$ or stop$) near10 (java application$1)</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>((kill$ or terminat$ or stop$) near10 (java application$1)) and ((kill$ or terminat$ or stop$) near10 (java application$1)).ab.</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>((kill$ or terminat$ or stop$) near10 (java application$1)) and ((kill$ or terminat$ or stop$) near10 (java application$1)).ab.) not ((kill$ or terminat$) near10 (java application$1)) and ((kill$ or terminat$) near10 (java application$1)).ab. )</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>((kill$ or terminat$ or stop$) near10 (java application$1)) ) and re-chang$</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>(kill$ or terminat$ or stop$) near10 (java)</td>
<td></td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>(((kill$ or terminat$ or stop$) near10 (java) ) and re-chang$)</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>(((kill$ or terminat$ or stop$) near10 (java) ) and (((kill$ or terminat$ or stop$) near10 (java)).ab. ) and (state$1 near5 (application$1 or java))</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>(((kill$ or terminat$ or stop$) near10 (java) ) and (((kill$ or terminat$ or stop$) near10 (java)).ab. ) and (state$1 near5 (application$1 or java))</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>(((kill$ or terminat$ or stop$) near10 (java) ) and switch$ near10 java near10 (background or foreground)</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>(((kill$ or terminat$ or stop$) near10 (java) ) and java near10 (background or foreground)</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>(((kill$ or terminat$ or stop$) near10 (java) ) and java near10 (background or foreground) ) and xlet</td>
<td>PGPB, USPT, USOC, EPAB,</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>JPAB, DWPI, TDBD</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------------------------</td>
<td>-----</td>
<td>----</td>
<td>----------</td>
<td>-----------</td>
</tr>
<tr>
<td>((kill$ or terminat$ or stop$) near10 (java) and java near10 (background or foreground) and xlet ) and @pd &gt; 20121022</td>
<td>PGPB, USPT</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>('5801785'</td>
<td>'6437811'</td>
<td>'20070058047'</td>
<td>'20080043031')![pn]</td>
<td>USPT, PGPB</td>
<td>ADJ</td>
</tr>
<tr>
<td>(719/310, 313)![CCLS]</td>
<td>PGPB, USPT</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>(718/100, 108)![CCLS]</td>
<td>PGPB, USPT</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>((718/100, 108)![CCLS] ) or ((719/310, 313)![CCLS] )</td>
<td>PGPB, USPT</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>((718/100, 108)![CCLS] or (719/310, 313)![CCLS] ) and (kill$ or terminat$ or paus$ and unpaus$ or halt$ or suspens$ or destroy$) same java</td>
<td>PGPB, USPT</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>((718/100, 108)![CCLS] or (719/310, 313)![CCLS] and (kill$ or terminat$ or paus$ and unpaus$ or halt$ or suspens$ or destroy$) same java ) and (background same foreground)</td>
<td>PGPB, USPT</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>((718/100, 108)![CCLS] or (719/310, 313)![CCLS] and (kill$ or terminat$ or paus$ and unpaus$ or halt$ or suspens$ or destroy$) same java ) destroyXlet$</td>
<td>PGPB, USPT</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>((718/100, 108)![CCLS] or (719/310, 313)![CCLS] and (kill$ or terminat$ or paus$ and unpaus$ or halt$ or suspens$ or destroy$) same java ) and destroyXlet$</td>
<td>PGPB, USPT</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>((718/100, 108)![CCLS] or (719/310, 313)![CCLS] and destroyXlet$</td>
<td>PGPB, USPT</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>((718/100, 108)![CCLS] or (719/310, 313)![CCLS] ) and pauseXlet$</td>
<td>PGPB, USPT</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>((718/100, 108)![CCLS] or (719/310, 313)![CCLS] and (kill$ or terminat$ or paus$ and unpaus$ or halt$ or suspens$ or destroy$) same java ) and (re-chang$ or chang$ or</td>
<td>PGPB, USPT</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>Recharge$() same state$1 same (java or application$1)</td>
<td>PGBP, USPT</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>-------------</td>
<td>-----</td>
<td>----</td>
<td>----------</td>
<td>------------</td>
</tr>
<tr>
<td>((718/100, 108)!CCLS or (719/310, 313)!CCLS and (kill$ or terminate$ or pause$ and unpause$ or halt$ or suspend$ or destroy$) same java and (re-change$ or change$ or rechange$) same state$1 same (java or application$1) ) not ((718/100, 108)!CCLS or (719/310, 313)!CCLS and (kill$ or terminate$ or pause$ and unpause$ or halt$ or suspend$ or destroy$) same java and (background same foreground) )</td>
<td>PGBP, USPT</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>(kill$ or terminate$ or pause$ and unpause$ or halt$ or suspend$ or destroy$) same java</td>
<td>PGBP, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>((kill$ or terminate$ or pause$ and unpause$ or halt$ or suspend$ or destroy$) same java ) and (background same foreground)</td>
<td>PGBP, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>((kill$ or terminate$ or pause$ and unpause$ or halt$ or suspend$ or destroy$) same java ) and (background same foreground)</td>
<td>PGBP, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>((kill$ or terminate$ or pause$ and unpause$ or halt$ or suspend$ or destroy$) same java and (kill$ or terminate$ or pause$ and unpause$ or halt$ or suspend$ or destroy$) same java)</td>
<td>PGBP, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>((kill$ or terminat$ or paus$ and unpaus$ or halt$ or suspen$ or destroy$) same java).ab. not (718/100, 108)!{CCLS} or (719/310, 313)!{CCLS} and (kill$ or terminat$ or paus$ and unpaus$ or halt$ or suspen$ or destroy$) same java and (re-chang$ or chang$ or rechang$) same state$1 same (java or application$1) not (718/100, 108)!{CCLS} or (719/310, 313)!{CCLS} and (kill$ or terminat$ or paus$ and unpaus$ or halt$ or suspen$ or destroy$) same java and (background same foreground) ) and (pauseXlet$ or destroyXlet$)</td>
<td>USOC, EPAB, JPAB, DWPI, TDBD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
<td></td>
</tr>
<tr>
<td>((kill$ or terminat$ or paus$ and unpaus$ or halt$ or suspen$ or destroy$) same java ) and (pauseXlet$ or destroyXlet$)</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>((kill$ or terminat$ or paus$ and unpaus$ or halt$ or suspen$ or destroy$) same java and (pauseXlet$ or destroyXlet$) ) not ((718/100, 108)!{CCLS} or (719/310, 313)!{CCLS} and (kill$ or terminat$ or paus$ and unpaus$ or halt$ or suspen$ or destroy$) same java and (background same foreground) )</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>('5121113'</td>
<td>'5537626'</td>
<td>'5887193'</td>
<td>'6140992'</td>
<td>'6151074'</td>
<td>'6744472'</td>
</tr>
<tr>
<td>((kill$ or terminat$ or paus$ and unpaus$ or halt$ or suspen$ or destroy$) same java ) and (re-chang$ or chang$ or rechang$) same state$1 same (java or application$1)</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
<tr>
<td>((kill$ or terminat$ or paus$ and unpaus$ or halt$ or suspen$ or destroy$) same java and (re-chang$ or chang$ or rechang$) same state$1 same (java or application$1) ) and (kill$ or terminat$ or halt$ or suspen$ or destroy$) same (java or application$1)</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
</tbody>
</table>
| ((kill$ or terminat$ or paus$ and unpaus$ or halt$ or suspen$ or destroy$) same java and (re-chang$ or chang$ or rechang$) same state$1 same (java or application$1) ) and (kill$ or
<table>
<thead>
<tr>
<th>terminat$ or halt$ or suspen$ or destroy$) same (java or application$1) same (execution or type$1 or information)</th>
<th>JPAB, DWPI, TDBD</th>
<th>ADJ</th>
<th>NO</th>
<th>ASSIGNEE</th>
<th>10-22-2012</th>
</tr>
</thead>
<tbody>
<tr>
<td>((kill$ or terminat$ or paus$ and unpaus$ or halt$ or suspen$ or destroy$) same java and (re-chang$ or chang$ or rechang$) same state$1 same (java or application$1) and (kill$ or terminat$ or halt$ or suspen$ or destroy$) same (java or application$1) same (execution or type$1 or information) ) and (background same foreground)</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>10-22-2012</td>
</tr>
</tbody>
</table>
IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of:

Wonjang Baek et al.

Application No. 12/656,872

Confirmation No. 2380

Filed: 2010-02-18

For: METHOD FOR MANAGING JAVA APPLICATIONS

AMENDMENT

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

This is in response to the Office Action mailed June 7, 2012, and having a period for response set to expire on September 7, 2012.

Reconsideration of the claims is respectfully requested. The following remarks are respectfully submitted.
IN THE ABSTRACT:

Please replace the abstract with the abstract provided below as amended below with added text with underlining and deleted text with strikethrough.

The present invention relates to a method for managing Java applications executable in a user device. The present invention provides an expandability for and a continuity between Java applications by changing states of the Java applications in execution and sharing information between the Java applications.
IN THE SPECIFICATION:

Please REPLACE the specification with the attached SUBSTITUTE SPECIFICATION. A clean version of the specification is also provided.
IN THE CLAIMS:

Please AMEND claims 1-22 in accordance with the following:

1. (Currently Amended) A method for managing javaJAVA applications executable in a user device, the method comprising steps of:

   (a) issuing a request for executing a second javaJAVA application, the request being issued by a first javaJAVA application and including execution type information, the execution type information containing information on a second application state of the second JAVA application, and a first application state of the first JAVA application while the second JAVA application is in the second application state:

   (b) changing a state of the first javaJAVA application to the first application state specified in according to the execution type information while the second JAVA application is in the second application state:

   (c) executing the second javaJAVA application; and

   (d) killing the second javaJAVA application when a request for killing the second javaJAVA application is issued by the second javaJAVA application and re-changing the state of the first javaJAVA application changed in the step (b) according to the execution type information.

2. (Currently Amended) The method in accordance with claim 1, wherein the first javaJAVA application generates the request for executing the second javaJAVA application based on a user input received through a user interface.

3. (Currently Amended) The method in accordance with claim 1, wherein the request for executing the second javaJAVA application further includes a path for obtaining the second javaJAVA application.

4. (Currently Amended) The method in accordance with claim 3, wherein the step (c) comprises executing the second javaJAVA application by obtaining the second javaJAVA application at a location defined by the path.
5. (Currently Amended) The method in accordance with claim 1, wherein the execution type information designates one or is selectable between:

a first execution type wherein the first javaJAVA application is killed when the second javaJAVA application is executed and the first javaJAVA application is restarted when the second javaJAVA application is killed;

a second execution type wherein the first javaJAVA application is paused when the second javaJAVA application is executed and the first javaJAVA application is unpaused when the second javaJAVA application is killed; and

a third execution type wherein the first javaJAVA application is switched to background when the second javaJAVA application is executed and the first javaJAVA application is switched to foreground when the second javaJAVA application is killed.

6. (Currently Amended) The method in accordance with claim 5, wherein the step (b) comprises:

killing the first javaJAVA application when the execution type information designates the first execution type;

pausing the first javaJAVA application when the execution type information designates the second execution type; and

switching the first javaJAVA application to background when the execution type information designates the third execution type.

7. (Currently Amended) The method in accordance with claim 5, wherein the re-changing the state of the first JAVA application in the step (d) comprises:

restarting the first javaJAVA application when the execution type information designates the first execution type;

unpausing the first javaJAVA application when the execution type information designates the second execution type; and

switching the first javaJAVA application to foreground when the execution type
information designates the third execution type.

8. **Currently Amended** The method in accordance with claim 1, wherein the first javaJAVA application and the second javaJAVA application inter-relays a-shared information.

9. **Currently Amended** The method in accordance with claim 7, wherein the first javaJAVA application and the second javaJAVA application inter-relays a-shared information when the execution type information designates one of the second execution type and the third execution type.

10. **Currently Amended** The method in accordance with claim 1, wherein each of the first javaJAVA application and the second javaJAVA application includes a javaJAVA application of Xlet format.

11. **Currently Amended** A method for managing javaJAVA applications executable in a user device, the method comprising steps of:

   (a) issuing a request for executing a second javaJAVA application, the request being issued by a first javaJAVA application and including a first application state of the first JAVA application while the second JAVA application is being executed;

   (b) pausing the first javaJAVA application, the first application state being a pause state;

   (c) executing the second javaJAVA application; and

   (d) killing the second javaJAVA application when a request for killing the second javaJAVA application is issued by the second javaJAVA application and unpausing the first javaJAVA application paused in the step (b) according to the request.

12. **Currently Amended** The method in accordance with claim 11, wherein the first javaJAVA application generates the request for executing the second javaJAVA application based on a user input received through a user interface.
13. (Currently Amended) The method in accordance with claim 11, wherein the request for executing the second javaJAVA application further includes a path for obtaining the second javaJAVA application.

14. (Currently Amended) The method in accordance with claim 13, wherein the step (c) comprises executing the second javaJAVA application by obtaining the second javaJAVA application at a location defined by the path.

15. (Currently Amended) The method in accordance with claim 11, wherein the first javaJAVA application and the second javaJAVA application inter-relays a shared information.

16. (Currently Amended) The method in accordance with claim 11, wherein each of the first javaJAVA application and the second javaJAVA application includes a javaJAVA application of Xlet format.

17. (Currently Amended) A method for managing javaJAVA applications executable in a user device, the method comprising steps of:

(a) issuing a request for executing a second javaJAVA application, the request being issued by a first javaJAVA application and including a first application state of the first JAVA application while the second JAVA application is being executed;

(b) switching the first javaJAVA application to background, the first application state being a background execution state;

(c) executing the second javaJAVA application; and

(d) killing the second javaJAVA application when a request for killing the second javaJAVA application is issued by the second javaJAVA application and switching the first javaJAVA application in background to foreground according to the request.

18. (Currently Amended) The method in accordance with claim 17, wherein the first javaJAVA application generates the request for executing the second javaJAVA application
based on a user input received through a user interface.

19. (Currently Amended) The method in accordance with claim 17, wherein the request for executing the second javaJAVA application further includes a path for obtaining the second javaJAVA application.

20. (Currently Amended) The method in accordance with claim 19, wherein the step (c) comprises executing the second javaJAVA application by obtaining the second javaJAVA application at a location defined by the path.

21. (Currently Amended) The method in accordance with claim 17, wherein the first javaJAVA application and the second javaJAVA application inter-relays a-shared information.

22. (Currently Amended) The method in accordance with claim 17, wherein each of the first javaJAVA application and the second javaJAVA application includes a javaJAVA application of Xlet format.
REMARKS

In accordance with the foregoing, the specification has been amended to improve form. Claims 1-22 have been amended, and claims 1-22 are pending and under consideration. No new matter is presented in this Amendment.

In the Office Action, the Examiner indicates that trademarks have been used and need to be corrected according to the format provided by the Examiner. However, the Examiner did not indicate which terms are believed to be trademarks. Applicants have revised the specification as best understood from the Examiner's instructions, but if the Examiner believes there are additional marks needing revision, it is respectfully requested that the Examiner specify which terms are believed to be marks.

REJECTIONS UNDER 35 U.S.C. §102:

On pages 4-7 of the Office Action, the Examiner rejects claims 1-22 under 35 U.S.C. §102(b) in view of Horii et al. (U.S. Patent Publication No. 2006/0080474). The rejection is respectfully traversed and reconsideration is requested.

By way of review, Horii et al. discloses an information processor which includes a Java middleware 740, an HTML browser 750 and a mailer 760 which are format engines for executing an application or a display of data. (Paragraph 0163; FIG. 7). The information processor further includes a navigator 720 which instructs the format engine. (Paragraph 0169). The Java middleware 740 includes an application manager 4503 which downloads and executes Java applications according to an AIT. The application manager 4503 first downloads the AIT, which includes an ID of the Java application and the control information for each application, such as autostart or kill. (Paragraphs 0246 & 0247; FIGs. 45 and 46). The application manager 4503 then changes a state of each Java application according to the AIT, an instruction of a Java application, or an instruction for the navigator 720. (Paragraph 0252). Each Java application can further change its state as well as change a state of another Java application using commands compliant with a DVB-MHP1.0 Standard. (Paragraph0254). However, to the extent that Horii et al. discloses using an AIT or a Java application to change a state of another Java application, there is no suggestion that one Java application coordinates its own and another state of another Java application using a common command or that changes in the another Java application's state affect the initiating Java application's state according to such a common command.
In contrast, claim 1 recites, among other features, "execution type information containing information on a second application state of the second JAVA application, and a first application state of the first JAVA application while the second JAVA application is in the second application state", "changing the first JAVA application to the first application state specified in the execution type information while the second JAVA application is in the second application state" or "killing the second JAVA application when a request for killing the second JAVA application is issued by the second JAVA application and re-changing the state of the first JAVA application changed in the step (b) according to the execution type information." As such, it is respectfully requested that the Examiner reconsider and withdraw the rejection of claim 1.

For at least similar reasons, it is respectfully submitted that Horii et al. does not disclose or suggest, among other features, "issuing a request for executing a second JAVA application, the request being issued by a first JAVA application and including a first application state of the first JAVA application while the second JAVA application is being executed", "pausing the first JAVA application, the first application state being a pause state", or "killing the second JAVA application when a request for killing the second JAVA application is issued by the second JAVA application and unpause the first JAVA application paused in the step (b) according to the request" as recited in claim 11; or "issuing a request for executing a second JAVA application, the request being issued by a first JAVA application and including a first application state of the first JAVA application while the second JAVA application is being executed", "switching the first JAVA application to background, the first application state being a background execution state," or "killing the second JAVA application when a request for killing the second JAVA application is issued by the second JAVA application and switching the first JAVA application in background to foreground according to the request" as recited in claim 17.

Lastly, while the Examiner relies upon paragraphs 0243 and 0247 as disclosing "a first execution type wherein the first JAVA application is killed when the second JAVA application is executed and the first JAVA application is restarted when the second JAVA application is killed" as recited in claim 5, these passages refer to the AIT shown in FIG. 46. However, as noted above in relation to the rejection of claim 1, the AIT is itself downloaded by the application manager 4503. There is no suggestion that the AIT is transmitted by a Java application or that any commands issued between Java applications have such a structure. In contrast, claim 5 recites, among other features, "issuing a request for executing a second JAVA application, the request being issued by a first JAVA application and including execution type information" where the "the execution type information is selectable between: a first execution type wherein
the first JAVA application is killed when the second JAVA application is executed and the first JAVA application is restarted when the second JAVA application is killed; a second execution type wherein the first JAVA application is paused when the second JAVA application is executed and the first JAVA application is unpaused when the second JAVA application is killed; and a third execution type wherein the first JAVA application is switched to background when the second JAVA application is executed and the first JAVA application is switched to foreground when the second JAVA application is killed.” As such, it is respectfully requested that the Examiner reconsider and withdraw the rejection of claim 5.

Claims 2-4, 6-10, 12-16, and 18-22 are deemed patentable due at least to their depending from corresponding claims 1, 11, and 17.

CONCLUSION:

There being no further outstanding objections or rejections, it is submitted that the application is in condition for allowance. An early action to that effect is courteously solicited.

Finally, if there are any formal matters remaining after this response, the Examiner is requested to telephone the undersigned to attend to these matters.

If there are any additional fees associated with filing of this Amendment, please charge the same to our Deposit Account No. 503333.

Respectfully submitted,

STEIN MCEWEN, LLP

Date: SEPT. 5, 2012

By: [Signature]

James G. McEwen
Registration No. 41983

1400 Eye St., N.W.
Suite 300
Washington, D.C. 20005
Telephone: (202) 216-9505
Facsimile: (202) 216-9510
TITLE OF THE INVENTION

METHOD FOR MANAGING JAVA APPLICATIONS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of Korean Patent Application No. 10-2009-0014161 filed on February 20, 2009, which is hereby incorporated for reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention
[0002] The present invention relates to a method for managing JAVA applications executable in a user device.

2. Description of the Related Art
[0003] Conventional televisions or conventional DVD players are only capable of playing audio/video data. However, latest digital televisions and blu-ray players are capable of providing an environment for executing applications.

[0004] Latest mobile communication terminals are also capable of providing the environment.

[0005] The applications executable in user devices such as the digital televisions, blu-ray players and mobile communication terminal is usually created using Java JAVA language (hereinafter referred to as "javaJAVA application") and provided to the user device in a byte code format.

[0006] Operations such as a start and an end of the application and an access to system resources are carried out between the application and an operating system on which the application is executed.

[0007] An application model defines how the application should be managed and also defines roles of the operating system and the application.

[0008] JavaJAVA Specifications such as J2ME(JavaJAVA 2 Platform Micro Edition) and J2SE(JavaJAVA 2 Platform Sun Edition) currently supports various application models. Applet and Xlet are examples of the application model.

[0009] Sun Microsystems provides Xlet for executing the javaJAVA application in an embedded device such as the digital television. While Applet is an embedded application which
is executed in a browser, Xlet is an embedded application executed in the embedded device.

[0010] Generally, one javaJAVA application is executed and displayed on a single screen of the user device.

[0011] For instance, when another javaJAVA application is selected for execution while a startup javaJAVA application is executed and displayed, the startup javaJAVA application is then halted and the selected javaJAVA application is executed.

[0012] In other words, the javaJAVA applications executed in the user device are independent of each other. Therefore, it is difficult to guarantee a continuity between the javaJAVA applications because information between the independent javaJAVA applications cannot be shared.

SUMMARY OF THE INVENTION

[0013] It is an object of the present invention to provide a method for managing javaJAVA applications wherein an expandability for and a continuity between javaJAVA applications are provided by changing states of the javaJAVA applications in execution and sharing information between the javaJAVA applications.

[0014] In order to achieve above-described objects of the present invention, there is provided a method for managing javaJAVA applications executable in a user device, the method comprising steps of: (a) issuing a request for executing a second javaJAVA application, the request being issued by a first javaJAVA application and including a execution type information; (b) changing a state of the first javaJAVA application according to the execution type information; (c) executing the second javaJAVA application; and (d) killing the second javaJAVA application when a request for killing the second javaJAVA application is issued by the second javaJAVA application and re-changing the state of the first javaJAVA application changed in the step (b).

[0015] Preferably, the first javaJAVA application generates the request for executing the second javaJAVA application based on a user input received through a user interface.

[0016] Preferably, the request for executing the second javaJAVA application further includes a path for obtaining the second javaJAVA application.

[0017] Preferably, the step (c) comprises executing the second javaJAVA application by
obtaining the second javaJAVA application at a location defined by the path.

[0018] Preferably, the execution type information designates one of: a first execution type wherein the first javaJAVA application is killed when the second javaJAVA application is executed and the first javaJAVA application is restarted when the second javaJAVA application is killed; a second execution type wherein the first javaJAVA application is paused when the second javaJAVA application is executed and the first javaJAVA application is unpaused when the second javaJAVA application is killed; and a third execution type wherein the first javaJAVA application is switched to background when the second javaJAVA application is executed and the first javaJAVA application is switched to foreground when the second javaJAVA application is killed.

[0019] Preferably, the step (b) comprises: killing the first javaJAVA application when the execution type information designates the first execution type; pausing the first javaJAVA application when the execution type information designates the second execution type; and switching the first javaJAVA application to background when the execution type information designates the third execution type.

[0020] Preferably, the step (d) comprises: restarting the first javaJAVA application when the execution type information designates the first execution type; unpausing the first javaJAVA application when the execution type information designates the second execution type; and switching the first javaJAVA application to foreground when the execution type information designates the third execution type.

[0021] Preferably, the first javaJAVA application and the second javaJAVA application inter-relays a shared information.

[0022] Preferably, the first javaJAVA application and the second javaJAVA application inter-relays a shared information when the execution type information designates one of the second execution type and the third execution type.

[0023] Preferably, each of the first javaJAVA application and the second javaJAVA application includes a javaJAVA application of Xlet format.

[0024] There is also provided a method for managing javaJAVA applications executable in a user device, the method comprising steps of: (a) issuing a request for executing a second javaJAVA application, the request being issued by a first javaJAVA application; (b) pausing the
first javaJAVA application; (c) executing the second javaJAVA application; and (d) killing the second javaJAVA application when a request for killing the second javaJAVA application is issued by the second javaJAVA application and unpauseing the first javaJAVA application paused in the step (b).

[0025] Preferably, the first javaJAVA application generates the request for executing the second javaJAVA application based on a user input received through a user interface.

[0026] Preferably, the request for executing the second javaJAVA application further includes a path for obtaining the second javaJAVA application.

[0027] Preferably, the step (c) comprises executing the second javaJAVA application by obtaining the second javaJAVA application at a location defined by the path.

[0028] Preferably, the first javaJAVA application and the second javaJAVA application inter-relays a shared information.

[0029] Preferably, each of the first javaJAVA application and the second javaJAVA application includes a javaJAVA application of Xlet format.

[0030] There is also provided a method for managing javaJAVA applications executable in a user device, the method comprising steps of: (a) issuing a request for executing a second javaJAVA application, the request being issued by a first javaJAVA application; (b) switching the first javaJAVA application to background; (c) executing the second javaJAVA application; and (d) killing the second javaJAVA application when a request for killing the second javaJAVA application is issued by the second javaJAVA application and switching the first javaJAVA application in background to foreground.

[0031] Preferably, the first javaJAVA application generates the request for executing the second javaJAVA application based on a user input received through a user interface.

[0032] Preferably, the request for executing the second javaJAVA application further includes a path for obtaining the second javaJAVA application.

[0033] Preferably, the step (c) comprises executing the second javaJAVA application by obtaining the second javaJAVA application at a location defined by the path.

[0034] Preferably, the first javaJAVA application and the second javaJAVA application inter-relays a shared information.
SUBSTITUTE SPECIFICATION –MARKED VERSION

[0035] Preferably, each of the first javaJAVa application and the second javaJAVa application includes a javaJAVa application of Xlet format.

BRIEF DESCRIPTION OF THE DRAWINGS

[0036] Fig. 1 is block diagram illustrating an environment for executing a method for managing a javaJAVa application in accordance with the present invention.

[0037] Fig. 2 is a flow diagram illustrating a first embodiment of a method for managing a javaJAVa application in accordance with the present invention.

[0038] Fig. 3 is a diagram illustrating a relationship between a first javaJAVa application and a second javaJAVa application in accordance with the present invention.

[0039] Fig. 4 is a flow diagram illustrating a change in a state of a first javaJAVa application based on an execution type information in accordance with the present invention.

[0040] Fig. 5 is a flow diagram illustrating re-changing a state of a first javaJAVa application based on an execution type information in accordance with the present invention.

[0041] Fig. 6 is a diagram illustrating a configuration wherein a state of a javaJAVa application is changed based on a second execution type in accordance with the present invention.

[0042] Fig. 7 is a diagram illustrating a configuration wherein a state of a javaJAVa application is changed based on a first execution type in accordance with the present invention.

[0043] Fig. 8 is a flow diagram illustrating a second embodiment of a method for managing a javaJAVa application in accordance with the present invention.

[0044] Fig. 9 is a flow diagram illustrating a third embodiment of a method for managing a javaJAVa application in accordance with the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0045] A method for managing a javaJAVa application in accordance with the present invention will be described with reference to accompanied drawings.

[0046] Fig. 1 is block diagram illustrating an environment for executing a method for managing a javaJAVa application in accordance with the present invention.

[0047] Referring to Fig. 1, the environment includes a javaJAVa application server 100 and a
user device 200.

[0048] The javaJAVA application server 100 stores a javaJAVA application and transmits the javaJAVA application to the user device 200 when requested by the user device 200.

[0049] The user device 200 executes the javaJAVA application received from the javaJAVA application server 100 to be provided to a user. The user device 200 may include a digital television, a set-top box, a blu-ray player or a mobile communication terminal.

[0050] The javaJAVA application server 100 and the user device 200 are connected via a communication network such as the Internet and a wireless network.

[0051] Hereinafter, an application developer refers to a person who develops the javaJAVA application which may be executed in the javaJAVA application server 100 or the user device 200, the user refers to a person who uses the user device 200 and takes advantage of a service provided via the javaJAVA application developed by the application developer.

[0052] The user device 200 comprises a boot program 210, a run-time program 230 and a javaJAVA application 290.

[0053] The boot program 210 is stored within the user device 200 or in a storage medium accessible by the user device 200. The boot program 210 checks a status of the user device 200 and searches and obtains the suitable run-time program 230.

[0054] The run-time program 230 provides a run-time environment for the javaJAVA application 290.

[0055] The run-time program 230 includes an X2let API 231, a method invocation API 233, a resource support API 235, an application manager API 237 and a debug support tool 239.

[0056] The X2let API 231 developed by the Applicant is implemented by an enhanced Xlet in compliance with JavaJAVA TV standard.

[0057] The X2let API 231 supports downloading of the javaJAVA application 290 from the javaJAVA application server 100 and execution of the downloaded javaJAVA application 290.

[0058] The method invocation API 233 developed by the Applicant executes a method in the javaJAVA application server 100 from the user device 200.

[0059] The resource support API 235 developed by the Applicant supports a memory-
effective caching and a resource downloading. The resource support API 235 may handle resources such as an image, a byte array, a sound and a javaJAVA class.

[0060] The application manager API 237 developed by the Applicant manages a life cycle of the javaJAVA application 290. The javaJAVA application 290 may be started, paused, unpaused, switched to background or foreground and killed via the application manager API 237.

[0061] The debug support tool 239 237 developed by the Applicant supports debugging.

[0062] The javaJAVA application 290 may be classified into a root javaJAVA application 293 and a user javaJAVA application 296.

[0063] The root javaJAVA application 293 is a base application executed in the user device 200. The root javaJAVA application 293 is required to be executed in the user device 200. The root javaJAVA application 293 is executed at a startup of the user device 200 and displayed on a screen of the user device 200. The root javaJAVA application 293 may include a user interface for selecting the user javaJAVA application 296 to be executed. The user interface may be implemented as a graphical user interface. The user may select the user javaJAVA application 296 to be executed via the user interface.

[0064] The user javaJAVA application 296 is selected by the user and then executed. The user javaJAVA application 296 refers to the javaJAVA application other than the root javaJAVA application 293.

[0065] While the root javaJAVA application 293 is created by a manufacturer of the user device 200 and distributed with the user device 200, the user javaJAVA application 296 is created by the application developer and transmitted to the user device 200 through the javaJAVA application server 100.

[0066] The method in accordance with the present invention will be described in detail hereinafter.

[0067] Fig. 2 is a flow diagram illustrating a first embodiment of the method for managing the javaJAVA application in accordance with the present invention.

[0068] Referring to Fig. 2, a request for executing a second javaJAVA application generated by a first javaJAVA application is issued (S110).

[0069] The request includes an execution type information.
The first javaJAVA application may be the root javaJAVA application denoted as reference numeral 293 in Fig. 1 or the user javaJAVA application denoted as reference numeral 296 in Fig. 1. The second javaJAVA application may be the user javaJAVA application. It is preferable that each of the first javaJAVA application and the second javaJAVA application includes the javaJAVA application of Xlet format executable in the user device 200.

Fig. 3 is a diagram illustrating a relationship between the first javaJAVA application and the second javaJAVA application in accordance with the present invention.

When the root javaJAVA application issues the request for executing a first child javaJAVA application, the root javaJAVA application and first child javaJAVA application correspond to the first javaJAVA application and the second javaJAVA application, respectively.

When first child javaJAVA application issues the request for executing a second child javaJAVA application, first child javaJAVA application and the second child javaJAVA application correspond to the first javaJAVA application and the second javaJAVA application, respectively.

The first javaJAVA application may include the user interface for receiving a user input.

The user interface may be the graphical user interface for executing the second javaJAVA application. When the user input is received while executing the first javaJAVA application, the first javaJAVA application generates and issues the request for executing the second javaJAVA application.

The request includes the execution type information which designates an execution type.

The execution type includes one of a first execution type, a second execution type and a third execution type.

In accordance with the first execution type, the first javaJAVA application is killed when the second javaJAVA application is executed and the first javaJAVA application is restarted when the second javaJAVA application is killed.

In accordance with the second execution type, the first javaJAVA application is paused when the second javaJAVA application is executed and the first javaJAVA application is unpaused when the second javaJAVA application is killed.
[0080] In accordance with the third execution type, the first javaJAVA application is switched to background when the second javaJAVA application is executed and the first javaJAVA application is switched back to foreground when the second javaJAVA application is killed.

[0081] In accordance with the present invention, a state of the first javaJAVA application is determined when the second javaJAVA application is executed.

[0082] When the execution type information does not designate certain the execution types, it may be regarded that the execution type information designates the first execution type.

[0083] That is, when the execution type information included in the request that does not designate the certain the execution type is issued, subsequent steps are carried out on an assumption that the execution type is the first execution type.

[0084] The request may also include a path for obtaining the second javaJAVA application.

[0085] When the second javaJAVA application is stored within the user device, the path may represent a location of the second javaJAVA application in the user device.

[0086] When the second javaJAVA application is stored in the javaJAVA application server, the path may represent a location of the second javaJAVA application in the javaJAVA application server.

[0087] The path may have a form of an URL (uniform resource locator).

[0088] Thereafter, the state of the first javaJAVA application is changed according to the execution type information included in the request issued in the step S110 (S130).

[0089] Fig. 4 is a flow diagram illustrating the step S130 in detail wherein the state of the first javaJAVA application is changed based on the execution type information.

[0090] Referring to Fig. 4, when the execution type designated in the execution type information is the first execution type, the first javaJAVA application is killed (S131 and S132).

[0091] When the execution type designated in the execution type information is the second execution type, the first javaJAVA application is paused (S134 and S135).

[0092] When the execution type designated in the execution type information is the third execution type, the first javaJAVA application is switched to background (S137 and S138).

[0093] Referring back to Fig. 2, the second javaJAVA application is obtained and executed in
parallel with or after carrying out the step S130 (S150).

[0094] Thereafter, When a request for killing the second javaJAVA application is issued while the second javaJAVA application is executed, the second javaJAVA application is killed and the first javaJAVA application state of which is changed in the step S130 is re-changed (S170).

[0095] Fig. 5 is a flow diagram illustrating re-changing the state of the first javaJAVA application based on the execution type information.

[0096] Referring to Fig. 5, when the execution type designated in the execution type information is the first execution type, the first javaJAVA application is re-started (S171 and S172).

[0097] When the execution type designated in the execution type information is the second execution type, the paused first javaJAVA application is unpaused (S174 and S175).

[0098] When the execution type designated in the execution type information is the third execution type, the first javaJAVA application in the background is switched back to foreground (S177 and S178).

[0099] The first javaJAVA application and the second javaJAVA application may relay a shared information to each other, i.e., inter-relay the shared information.

[00100] Specifically, if the second javaJAVA application executed in the step S150 obtains the shared information that is to be shared with the first javaJAVA application, the second javaJAVA application may relay the shared information to the first javaJAVA application in the step S170, and the first javaJAVA application may use the relayed information.

[00101] In one embodiment, when the first javaJAVA application receives the user input and the received user input includes a certain information which corresponds to the shared information, the first javaJAVA application transmits the certain information included in the user input to the second javaJAVA application, and the second javaJAVA application may use the certain information.

[00102] Similarly, when the second javaJAVA application receives the user input and the received user input includes a certain information which corresponds to the shared information, the second javaJAVA application transmits the certain information included in the user input to the first javaJAVA application, and the first javaJAVA application may use the certain information.
Particularly, the shared information may be inter- relayed when the execution type is the second execution type or the third execution type.

Fig. 6 is a diagram illustrating a configuration wherein the state of the javaJAVA application is changed based on the second execution type.

An application manager shown in Fig. 6 implements the application manager API shown in Fig. 1.

Referring to Fig. 6, the root javaJAVA application transmits a request for executing the first child javaJAVA application to the application manager.

Here, the root javaJAVA application and the first child javaJAVA application correspond to the first javaJAVA application and the second javaJAVA application, respectively.

The request for executing the first child javaJAVA application is denoted as startApp(root, Bound, c1URL) in Fig. 6.

startApp(root, Bound, c1URL) represents the request for executing, by the root javaJAVA application (denoted as “root”), the first child javaJAVA application located at a path “c1URL” according to the second execution type (denoted as “Bound”).

When the request startApp(root, Bound, c1URL) is received, the application manager pauses the root javaJAVA application using pauseXlet() function.

The application manager also initializes the first child javaJAVA application using initXlet() function, and starts the first child javaJAVA application using startXlet() function.

Thereafter, the first child javaJAVA application transmits a request for executing the second child javaJAVA application to the application manager.

Here, the first child javaJAVA application and the second child javaJAVA application correspond to the first javaJAVA application and the second javaJAVA application, respectively.

The request for executing the second child javaJAVA application is denoted as startApp(child1, Bound, c2URL) in Fig. 6.

startApp(child1, Bound, c2URL) represents the request for executing, by the first child javaJAVA application (denoted as “child1”), the second child javaJAVA application located at a path “c2URL” according to the second execution type (denoted as “Bound”).
[00116] When the request startApp(child1, Bound, c2URL) is received, the application manager pauses the root javaJAVA application using pauseXlet() function.

[00117] The application manager also initializes the second child javaJAVA application using initXlet() function, and starts the second child javaJAVA application using startXlet() function.

[00118] When destroyApp(child2) which is a request for killing the second child javaJAVA application is received from the second child javaJAVA application, the application manager kills the second child javaJAVA application using destroyXlet() function.

[00119] When the second child javaJAVA application is killed, the application manager unpauses the paused first child javaJAVA application. In other words, the application manager starts the first child javaJAVA application using startXlet() function.

[00120] When destroyApp(child1) which is a request for killing the first child javaJAVA application is received from the first child javaJAVA application, the application manager kills the first child javaJAVA application using destroyXlet() function.

[00121] When the first child javaJAVA application is killed, the application manager unpauses the paused root javaJAVA application. In other words, the application manager starts the root javaJAVA application using startXlet() function.

[00122] Fig. 7 is a diagram illustrating a configuration wherein the state of the javaJAVA application is changed based on the first execution type.

[00123] Referring to Fig. 7, the root javaJAVA application transmits a request for executing a javaJAVA application "A" to the application manager.

[00124] The request for executing the javaJAVA application "A" is denoted as startApp(root, Standard, aURL) in Fig. 7.

[00125] startApp(root, Standard, aURL) represents the request for executing, by the root javaJAVA application (denoted as "root"), the javaJAVA application "A" located at a path "aURL" according to the first execution type (denoted as "Standard").

[00126] When the request startApp(root, Standard, aURL) is received, the application manager kills the root javaJAVA application using destroyXlet() function.

[00127] The application manager also initializes the javaJAVA application "A" using initXlet() function, and starts the javaJAVA application "A" using startXlet() function.
SUBSTITUTE SPECIFICATION – MARKED VERSION

[00128] When destroyApp(A) which is a request for killing the javaJAVA application "A" is received from the javaJAVA application "A", the application manager kills the javaJAVA application "A" using destroyXlet() function.

[00129] When the javaJAVA application "A" is killed, the application manager restarts the root javaJAVA application killed by destroyXlet() function. In other words, the application manager starts the root javaJAVA application using startXlet() function.

[00130] Thereafter, the root javaJAVA application transmits a request for executing a javaJAVA application "B" to the application manager.

[00131] The request for executing the javaJAVA application "B" is denoted as startApp(root, Standard, bURL) in Fig. 7.

[00132] startApp(root, Standard, bURL) represents the request for executing, by the root javaJAVA application (denoted as "root"), the javaJAVA application "B" located at a path "bURL" according to the first execution type (denoted as "Standard").

[00133] When the request startApp(root, Standard, bURL) is received, the application manager kills the root javaJAVA application using destroyXlet() function.

[00134] The application manager also initializes the javaJAVA application "B" using initXlet() function, and starts the javaJAVA application "B" using startXlet() function.

[00135] When destroyApp(B) which is a request for killing the javaJAVA application "B" is received from the javaJAVA application "B", the application manager kills the javaJAVA application "B" using destroyXlet() function.

[00136] When the javaJAVA application "B" is killed, the application manager restarts the root javaJAVA application killed by destroyXlet() function. In other words, the application manager starts the root javaJAVA application using startXlet() function.

[00137] A configuration for changing the state of the javaJAVA application based on the third execution type is similar to that shown in Fig. 6.

[00138] When the state of the javaJAVA application is changed based on the third execution type, background() function for switching the javaJAVA application to the background is used in place of pauseXlet() function, and the javaJAVA application in the background is then switched to foreground.
[00139] Fig. 8 is a flow diagram illustrating a second embodiment of the method for managing the java JAVA application in accordance with the present invention.

[00140] Referring to Fig. 8, a request for executing the second java JAVA application generated by the first java JAVA application is issued (S210).

[00141] Thereafter, the first java JAVA application is paused (S230).

[00142] Thereafter, the second java JAVA application is obtained and executed in parallel with or after carrying out the step S230 (S250).

[00143] When a request for killing the second java JAVA application is issued while the second java JAVA application is executed, the second java JAVA application is killed and the first java JAVA application paused in the step S230 is unpaused (S270).

[00144] In accordance with the second embodiment, the execution type information is not included in the request for executing the second java JAVA application. The second embodiment is identical to the first embodiment except that the execution type information is not included in the request. Therefore, a detailed description is omitted.

[00145] Fig. 9 is a flow diagram illustrating a third embodiment of the method for managing the java JAVA application in accordance with the present invention.

[00146] Referring to Fig. 9, a request for executing the second java JAVA application generated by the first java JAVA application is issued (S310).

[00147] Thereafter, the first java JAVA application is switched to background (S330).

[00148] Thereafter, the second java JAVA application is obtained and executed in parallel with or after carrying out the step S330 (S350).

[00149] When a request for killing the second java JAVA application is issued while the second java JAVA application is executed, the second java JAVA application is killed and the first java JAVA application in the background is switched to foreground (S370).

[00150] In accordance with the third embodiment, the execution type information is not included in the request for executing the second java JAVA application. The third embodiment is identical to the second embodiment except that the first java JAVA application is switched to background instead of being paused. Therefore, a detailed description is omitted.
[00151] In accordance with the present invention, because the first javaJAVA application may be associated with the second javaJAVA application, a use of the javaJAVA application for the user is facilitated, and the javaJAVA applications developed by different application developers can be associated with each other in order to improve an expandability. For instance, the application developer may create the first javaJAVA application in a manner that the first javaJAVA application includes a path for the second javaJAVA application in order for the user to take advantage of the second javaJAVA application.

[00152] Particularly, the present invention may be applied to base applications of TV portal services, smart phones and blu-rayBlu-ray players employing the javaJAVA application.

[00153] Various services may be provided based on the execution type information and the path even when the base application does not include every function.

[00154] Moreover, the first javaJAVA application and the second javaJAVA application inter-relays the shared information in order to guarantee a continuity therebetween.

[00155] While the present invention has been particularly shown and described with reference to the preferred embodiment thereof, it will be understood by those skilled in the art that various changes in form and details may be effected therein without departing from the spirit and scope of the invention as defined by the appended claims.
IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of:
Wonjang Baek et al.

Application No. 12/656,872
Confirmation No. 2380

Filed: 2010-02-18

For: METHOD FOR MANAGING JAVA APPLICATIONS

SUBSTITUTE SPECIFICATION - CLEAN VERSION
SUBSTITUTE SPECIFICATION –CLEAN VERSION

TITLE OF THE INVENTION

METHOD FOR MANAGING JAVA APPLICATIONS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit of Korean Patent Application No. 10-2009-0014161 filed on February 20, 2009, which is hereby incorporated for reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

[0002] The present invention relates to a method for managing JAVA applications executable in a user device.

2. Description of the Related Art

[0003] Conventional televisions or conventional DVD players are only capable of playing audio/video data. However, latest digital televisions and blu-ray players are capable of providing an environment for executing applications.

[0004] Latest mobile communication terminals are also capable of providing the environment.

[0005] The applications executable in user devices such as the digital televisions, blu-ray players and mobile communication terminal is usually created using JAVA language (hereinafter referred to as “JAVA application”) and provided to the user device in a byte code format.

[0006] Operations such as a start and an end of the application and an access to system resources are carried out between the application and an operating system on which the application is executed.

[0007] An application model defines how the application should be managed and also defines roles of the operating system and the application.

[0008] JAVA Specifications such as J2ME(JAVA 2 Platform Micro Edition) and J2SE( JAVA 2 Platform Sun Edition) currently supports various application models. Applet and Xlet are examples of the application model.

[0009] Sun Microsystems provides Xlet for executing the JAVA application in an embedded device such as the digital television. While Applet is an embedded application which is executed in a browser, Xlet is an embedded application executed in the embedded device.
Generally, one JAVA application is executed and displayed on a single screen of the user device.

For instance, when another JAVA application is selected for execution while a startup JAVA application is executed and displayed, the startup JAVA application is then halted and the selected JAVA application is executed.

In other words, the JAVA applications executed in the user device are independent of each other. Therefore, it is difficult to guarantee a continuity between the JAVA applications because information between the independent JAVA applications cannot be shared.

SUMMARY OF THE INVENTION

It is an object of the present invention to provide a method for managing JAVA applications wherein an expandability for and a continuity between JAVA applications are provided by changing states of the JAVA applications in execution and sharing information between the JAVA applications.

In order to achieve above-described objects of the present invention, there is provided a method for managing JAVA applications executable in a user device, the method comprising steps of: (a) issuing a request for executing a second JAVA application, the request being issued by a first JAVA application and including a execution type information; (b) changing a state of the first JAVA application according to the execution type information; (c) executing the second JAVA application; and (d) killing the second JAVA application when a request for killing the second JAVA application is issued by the second JAVA application and re-changing the state of the first JAVA application changed in the step (b).

Preferably, the first JAVA application generates the request for executing the second JAVA application based on a user input received through a user interface.

Preferably, the request for executing the second JAVA application further includes a path for obtaining the second JAVA application.

Preferably, the step (c) comprises executing the second JAVA application by obtaining the second JAVA application at a location defined by the path.

Preferably, the execution type information designates one of: a first execution type wherein the first JAVA application is killed when the second JAVA application is executed and
the first JAVA application is restarted when the second JAVA application is killed; a second execution type wherein the first JAVA application is paused when the second JAVA application is executed and the first JAVA application is unpaused when the second JAVA application is killed; and a third execution type wherein the first JAVA application is switched to background when the second JAVA application is executed and the first JAVA application is switched to foreground when the second JAVA application is killed.

[0019] Preferably, the step (b) comprises: killing the first JAVA application when the execution type information designates the first execution type; pausing the first JAVA application when the execution type information designates the second execution type; and switching the first JAVA application to background when the execution type information designates the third execution type.

[0020] Preferably, the step (d) comprises: restarting the first JAVA application when the execution type information designates the first execution type; unpausing the first JAVA application when the execution type information designates the second execution type; and switching the first JAVA application to foreground when the execution type information designates the third execution type.

[0021] Preferably, the first JAVA application and the second JAVA application inter-relays a shared information.

[0022] Preferably, the first JAVA application and the second JAVA application inter-relays a shared information when the execution type information designates one of the second execution type and the third execution type.

[0023] Preferably, each of the first JAVA application and the second JAVA application includes a JAVA application of Xlet format.

[0024] There is also provided a method for managing JAVA applications executable in a user device, the method comprising steps of: (a) issuing a request for executing a second JAVA application, the request being issued by a first JAVA application; (b) pausing the first JAVA application; (c) executing the second JAVA application; and (d) killing the second JAVA application when a request for killing the second JAVA application is issued by the second JAVA application and unpausing the first JAVA application paused in the step (b).

[0025] Preferably, the first JAVA application generates the request for executing the second
JAVA application based on a user input received through a user interface.

[0026] Preferably, the request for executing the second JAVA application further includes a path for obtaining the second JAVA application.

[0027] Preferably, the step (c) comprises executing the second JAVA application by obtaining the second JAVA application at a location defined by the path.

[0028] Preferably, the first JAVA application and the second JAVA application inter-relays a shared information.

[0029] Preferably, each of the first JAVA application and the second JAVA application includes a JAVA application of Xlet format.

[0030] There is also provided a method for managing JAVA applications executable in a user device, the method comprising steps of: (a) issuing a request for executing a second JAVA application, the request being issued by a first JAVA application; (b) switching the first JAVA application to background; (c) executing the second JAVA application; and (d) killing the second JAVA application when a request for killing the second JAVA application is issued by the second JAVA application and switching the first JAVA application in background to foreground.

[0031] Preferably, the first JAVA application generates the request for executing the second JAVA application based on a user input received through a user interface.

[0032] Preferably, the request for executing the second JAVA application further includes a path for obtaining the second JAVA application.

[0033] Preferably, the step (c) comprises executing the second JAVA application by obtaining the second JAVA application at a location defined by the path.

[0034] Preferably, the first JAVA application and the second JAVA application inter-relays a shared information.

[0035] Preferably, each of the first JAVA application and the second JAVA application includes a JAVA application of Xlet format.

BRIEF DESCRIPTION OF THE DRAWINGS

[0036] Fig. 1 is block diagram illustrating an environment for executing a method for managing a JAVA application in accordance with the present invention.
SUBSTITUTE SPECIFICATION –CLEAN VERSION

[0037] Fig. 2 is a flow diagram illustrating a first embodiment of a method for managing a JAVA application in accordance with the present invention.

[0038] Fig. 3 is a diagram illustrating a relationship between a first JAVA application and a second JAVA application in accordance with the present invention.

[0039] Fig. 4 is a flow diagram illustrating a change in a state of a first JAVA application based on an execution type information in accordance with the present invention.

[0040] Fig. 5 is a flow diagram illustrating re-changing a state of a first JAVA application based on an execution type information in accordance with the present invention.

[0041] Fig. 6 is a diagram illustrating a configuration wherein a state of a JAVA application is changed based on a second execution type in accordance with the present invention.

[0042] Fig. 7 is a diagram illustrating a configuration wherein a state of a JAVA application is changed based on a first execution type in accordance with the present invention.

[0043] Fig. 8 is a flow diagram illustrating a second embodiment of a method for managing a JAVA application in accordance with the present invention.

[0044] Fig. 9 is a flow diagram illustrating a third embodiment of a method for managing a JAVA application in accordance with the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0045] A method for managing a JAVA application in accordance with the present invention will be described with reference to accompanied drawings.

[0046] Fig. 1 is block diagram illustrating an environment for executing a method for managing a JAVA application in accordance with the present invention.

[0047] Referring to Fig. 1, the environment includes a JAVA application server 100 and a user device 200.

[0048] The JAVA application server 100 stores a JAVA application and transmits the JAVA application to the user device 200 when requested by the user device 200.

[0049] The user device 200 executes the JAVA application received from the JAVA application server 100 to be provided to a user. The user device 200 may include a digital television, a set-top box, a blu-ray player or a mobile communication terminal.
[0050] The JAVA application server 100 and the user device 200 are connected via a communication network such as the Internet and a wireless network.

[0051] Hereinafter, an application developer refers to a person who develops the JAVA application which may be executed in the JAVA application server 100 or the user device 200, the user refers to a person who uses the user device 200 and takes advantage of a service provided via the JAVA application developed by the application developer.

[0052] The user device 200 comprises a boot program 210, a run-time program 230 and a JAVA application 290.

[0053] The boot program 210 is stored within the user device 200 or in a storage medium accessible by the user device 200. The boot program 210 checks a status of the user device 200 and searches and obtains the suitable run-time program 230.

[0054] The run-time program 230 provides a run-time environment for the JAVA application 290.

[0055] The run-time program 230 includes an X2let API 231, a method invocation API 233, a resource support API 235, an application manager API 237 and a debug support tool 239.

[0056] The X2let API 231 developed by the Applicant is implemented by an enhanced Xlet in compliance with JAVA TV standard.

[0057] The X2let API 231 supports downloading of the JAVA application 290 from the JAVA application server 100 and execution of the downloaded JAVA application 290.

[0058] The method invocation API 233 developed by the Applicant executes a method in the JAVA application server 100 from the user device 200.

[0059] The resource support API 235 developed by the Applicant supports a memory-effective caching and a resource downloading. The resource support API 235 may handle resources such as an image, a byte array, a sound and a JAVA class.

[0060] The application manager API 237 developed by the Applicant manages a life cycle of the JAVA application 290. The JAVA application 290 may be started, paused, unpaused, switched to background or foreground and killed via the application manager API 237.

[0061] The debug support tool 239 237 developed by the Applicant supports debugging.
[0062] The JAVA application 290 may be classified into a root JAVA application 293 and a user JAVA application 296.

[0063] The root JAVA application 293 is a base application executed in the user device 200. The root JAVA application 293 is required to be executed in the user device 200. The root JAVA application 293 is executed at a startup of the user device 200 and displayed on a screen of the user device 200. The root JAVA application 293 may include a user interface for selecting the user JAVA application 296 to be executed. The user interface may be implemented as a graphical user interface. The user may select the user JAVA application 296 to be executed via the user interface.

[0064] The user JAVA application 296 is selected by the user and then executed. The user JAVA application 296 refers to the JAVA application other than the root JAVA application 293.

[0065] While the root JAVA application 293 is created by a manufacturer of the user device 200 and distributed with the user device 200, the user JAVA application 296 is created by the application developer and transmitted to the user device 200 through the JAVA application server 100.

[0066] The method in accordance with the present invention will be described in detail hereinafter.

[0067] Fig. 2 is a flow diagram illustrating a first embodiment of the method for managing the JAVA application in accordance with the present invention.

[0068] Referring to Fig. 2, a request for executing a second JAVA application generated by a first JAVA application is issued (S110).

[0069] The request includes an execution type information.

[0070] The first JAVA application may be the root JAVA application denoted as reference numeral 293 in Fig. 1 or the user JAVA application denoted as reference numeral 296 in Fig. 1. The second JAVA application may be the user JAVA application. It is preferable that each of the first JAVA application and the second JAVA application includes the JAVA application of Xlet format executable in the user device 200.

[0071] Fig. 3 is a diagram illustrating a relationship between the first JAVA application and the second JAVA application in accordance with the present invention.
When the root JAVA application issues the request for executing a first child JAVA application, the root JAVA application and first child JAVA application correspond to the first JAVA application and the second JAVA application, respectively.

When first child JAVA application issues the request for executing a second child JAVA application, first child JAVA application and the second child JAVA application correspond to the first JAVA application and the second JAVA application, respectively.

The first JAVA application may include the user interface for receiving a user input.

The user interface may be the graphical user interface for executing the second JAVA application. When the user input is received while executing the first JAVA application, the first JAVA application generates and issues the request for executing the second JAVA application.

The request includes the execution type information which designates an execution type.

The execution type includes one of a first execution type, a second execution type and a third execution type.

In accordance with the first execution type, the first JAVA application is killed when the second JAVA application is executed and the first JAVA application is restarted when the second JAVA application is killed.

In accordance with the second execution type, the first JAVA application is paused when the second JAVA application is executed and the first JAVA application is unpaued when the second JAVA application is killed.

In accordance with the third execution type, the first JAVA application is switched to background when the second JAVA application is executed and the first JAVA application is switched back to foreground when the second JAVA application is killed.

In accordance with the present invention, a state of the first JAVA application is determined when the second JAVA application is executed.

When the execution type information does not designate certain the execution types, it may be regarded that the execution type information designates the first execution type.

That is, when the execution type information included in the request that does not designate the certain execution type is issued, subsequent steps are carried out on an
assumption that the execution type is the first execution type.

[0084] The request may also include a path for obtaining the second JAVA application.

[0085] When the second JAVA application is stored within the user device, the path may represent a location of the second JAVA application in the user device.

[0086] When the second JAVA application is stored in the JAVA application server, the path may represent a location of the second JAVA application in the JAVA application server.

[0087] The path may have a form of an URL (uniform resource locator).

[0088] Thereafter, the state of the first JAVA application is changed according to the execution type information included in the request issued in the step S110 (S130).

[0089] Fig. 4 is a flow diagram illustrating the step S130 in detail wherein the state of the first JAVA application is changed based on the execution type information.

[0090] Referring to Fig. 4, when the execution type designated in the execution type information is the first execution type, the first JAVA application is killed (S131 and S132).

[0091] When the execution type designated in the execution type information is the second execution type, the first JAVA application is paused (S134 and S135).

[0092] When the execution type designated in the execution type information is the third execution type, the first JAVA application is switched to background (S137 and S138).

[0093] Referring back to Fig. 2, the second JAVA application is obtained and executed in parallel with or after carrying out the step S130 (S150).

[0094] Thereafter, When a request for killing the second JAVA application is issued while the second JAVA application is executed, the second JAVA application is killed and the first JAVA application state of which is changed in the step S130 is re-changed (S170).

[0095] Fig. 5 is a flow diagram illustrating re-changing the state of the first JAVA application based on the execution type information.

[0096] Referring to Fig. 5, when the execution type designated in the execution type information is the first execution type, the first JAVA application is re-started (S171 and S172).

[0097] When the execution type designated in the execution type information is the second execution type, the paused first JAVA application is unpaused (S174 and S175).
[0098] When the execution type designated in the execution type information is the third execution type, the first JAVA application in the background is switched back to foreground (S177 and S178).

[0099] The first JAVA application and the second JAVA application may relay a shared information to each other, i.e., inter-relay the shared information.

[0100] Specifically, if the second JAVA application executed in the step S150 obtains the shared information that is to be shared with the first JAVA application, the second JAVA application may relay the shared information to the first JAVA application in the step S170, and the first JAVA application may use the relayed information.

[0101] In one embodiment, when the first JAVA application receives the user input and the received user input includes a certain information which corresponds to the shared information, the first JAVA application transmits the certain information included in the user input to the second JAVA application, and the second JAVA application may use the certain information.

[0102] Similarly, when the second JAVA application receives the user input and the received user input includes a certain information which corresponds to the shared information, the second JAVA application transmits the certain information included in the user input to the first JAVA application, and the first JAVA application may use the certain information.

[0103] Particularly, the shared information may be inter-relayed when the execution type is the second execution type or the third execution type.

[0104] Fig. 6 is a diagram illustrating a configuration wherein the state of the JAVA application is changed based on the second execution type.

[0105] An application manager shown in Fig. 6 implements the application manager API shown in Fig. 1.

[0106] Referring to Fig. 6, the root JAVA application transmits a request for executing the first child JAVA application to the application manager.

[0107] Here, the root JAVA application and the first child JAVA application correspond to the first JAVA application and the second JAVA application, respectively.

[0108] The request for executing the first child JAVA application is denoted as startApp(root, Bound, c1URL) in Fig. 6.
startApp(root, Bound, c1URL) represents the request for executing, by the root JAVA application (denoted as "root"), the first child JAVA application located at a path "c1URL" according to the second execution type (denoted as "Bound").

When the request startApp(root, Bound, c1URL) is received, the application manager pauses the root JAVA application using pauseXlet() function.

The application manager also initializes the first child JAVA application using initXlet() function, and starts the first child JAVA application using startXlet() function.

Thereafter, the first child JAVA application transmits a request for executing the second child JAVA application to the application manager.

Here, the first child JAVA application and the second child JAVA application correspond to the first JAVA application and the second JAVA application, respectively.

The request for executing the second child JAVA application is denoted as startApp(child1, Bound, c2URL) in Fig. 6.

startApp(child1, Bound, c2URL) represents the request for executing, by the first child JAVA application (denoted as "child1"), the second child JAVA application located at a path "c2URL" according to the second execution type (denoted as "Bound").

When the request startApp(child1, Bound, c2URL) is received, the application manager pauses the root JAVA application using pauseXlet() function.

The application manager also initializes the second child JAVA application using initXlet() function, and starts the second child JAVA application using startXlet() function.

When destroyApp(child2) which is a request for killing the second child JAVA application is received from the second child JAVA application, the application manager kills the second child JAVA application using destroyXlet() function.

When the second child JAVA application is killed, the application manager unpauses the paused first child JAVA application. In other words, the application manager starts the first child JAVA application using startXlet() function.

When destroyApp(child1) which is a request for killing the first child JAVA application is received from the first child JAVA application, the application manager kills the first child JAVA application using destroyXlet() function.
[00121] When the first child JAVA application is killed, the application manager unpause the
paused root JAVA application. In other words, the application manager starts the root JAVA
application using startXlet() function.

[00122] Fig. 7 is a diagram illustrating a configuration wherein the state of the JAVA application
is changed based on the first execution type.

[00123] Referring to Fig. 7, the root JAVA application transmits a request for executing a JAVA
application "A" to the application manager.

[00124] The request for executing the JAVA application "A" is denoted as startApp(root,
Standard, aURL) in Fig. 7.

[00125] startApp(root, Standard, aURL) represents the request for executing, by the root JAVA
application (denoted as "root"), the JAVA application "A" located at a path "aURL" according to
the first execution type (denoted as "Standard").

[00126] When the request startApp(root, Standard, aURL) is received, the application
manager kills the root JAVA application using destroyXlet() function.

[00127] The application manager also initializes the JAVA application "A" using initXlet() function, and starts the JAVA application "A" using startXlet() function.

[00128] When destroyApp(A) which is a request for killing the JAVA application "A" is received
from the JAVA application "A", the application manager kills the JAVA application "A" using
destroyXlet() function.

[00129] When the JAVA application "A" is killed, the application manager restarts the root
JAVA application killed by destroyXlet() function. In other words, the application manager starts
the root JAVA application using startXlet() function.

[00130] Thereafter, the root JAVA application transmits a request for executing a JAVA
application "B" to the application manager.

[00131] The request for executing the JAVA application "B" is denoted as startApp(root,
Standard, bURL) in Fig. 7.

[00132] startApp(root, Standard, bURL) represents the request for executing, by the root JAVA
application (denoted as "root"), the JAVA application "B" located at a path "bURL" according to
the first execution type (denoted as "Standard").
SUBSTITUTE SPECIFICATION – CLEAN VERSION

[00133] When the request startApp(root, Standard, bURL) is received, the application manager kills the root JAVA application using destroyXlet() function.

[00134] The application manager also initializes the JAVA application "B" using initXlet() function, and starts the JAVA application "B" using startXlet() function.

[00135] When destroyApp(B) which is a request for killing the JAVA application "B" is received from the JAVA application "B", the application manager kills the JAVA application "B" using destroyXlet() function.

[00136] When the JAVA application "B" is killed, the application manager restarts the root JAVA application killed by destroyXlet() function. In other words, the application manager starts the root JAVA application using startXlet() function.

[00137] A configuration for changing the state of the JAVA application based on the third execution type is similar to that shown in Fig. 6.

[00138] When the state of the JAVA application is changed based on the third execution type, background() function for switching the JAVA application to the background is used in place of pauseXlet() function, and the JAVA application in the background is then switched to foreground.

[00139] Fig. 8 is a flow diagram illustrating a second embodiment of the method for managing the JAVA application in accordance with the present invention.

[00140] Referring to Fig. 8, a request for executing the second JAVA application generated by the first JAVA application is issued (S210).

[00141] Thereafter, the first JAVA application is paused (S230).

[00142] Thereafter, the second JAVA application is obtained and executed in parallel with or after carrying out the step S230 (S250).

[00143] When a request for killing the second JAVA application is issued while the second JAVA application is executed, the second JAVA application is killed and the first JAVA application paused in the step S230 is unpaused (S270).

[00144] In accordance with the second embodiment, the execution type information is not included in the request for executing the second JAVA application. The second embodiment is identical to the first embodiment except that the execution type information is not included in the request. Therefore, a detailed description is omitted.
[00145] Fig. 9 is a flow diagram illustrating a third embodiment of the method for managing the JAVA application in accordance with the present invention.

[00146] Referring to Fig. 9, a request for executing the second JAVA application generated by the first JAVA application is issued (S310).

[00147] Thereafter, the first JAVA application is switched to background (S330).

[00148] Thereafter, the second JAVA application is obtained and executed in parallel with or after carrying out the step S330 (S350).

[00149] When a request for killing the second JAVA application is issued while the second JAVA application is executed, the second JAVA application is killed and the first JAVA application in the background is switched to foreground (S370).

[00150] In accordance with the third embodiment, the execution type information is not included in the request for executing the second JAVA application. The third embodiment is identical to the second embodiment except that the first JAVA application is switched to background instead of being paused. Therefore, a detailed description is omitted.

[00151] In accordance with the present invention, because the first JAVA application may be associated with the second JAVA application, a use of the JAVA application for the user is facilitated, and the JAVA applications developed by different application developers can be associated with each other in order to improve an expandability. For instance, the application developer may create the first JAVA application in a manner that the first JAVA application includes a path for the second JAVA application in order for the user to take advantage of the second JAVA application.

[00152] Particularly, the present invention may be applied to base applications of TV portal services, smart phones and Blu-ray players employing the JAVA application.

[00153] Various services may be provided based on the execution type information and the path even when the base application does not include every function.

[00154] Moreover, the first JAVA application and the second JAVA application inter-relays the shared information in order to guarantee a continuity therebetween.

[00155] While the present invention has been particularly shown and described with reference to the preferred embodiment thereof, it will be understood by those skilled in the art that various
changes in form and details may be effected therein without departing from the spirit and scope of the invention as defined by the appended claims.
## Electronic Acknowledgement Receipt

<table>
<thead>
<tr>
<th>EFS ID</th>
<th>13662882</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Number</td>
<td>12656872</td>
</tr>
<tr>
<td>International Application Number</td>
<td></td>
</tr>
<tr>
<td>Confirmation Number</td>
<td>2380</td>
</tr>
<tr>
<td><strong>Title of Invention:</strong></td>
<td>Method for managing Java applications</td>
</tr>
<tr>
<td><strong>First Named Inventor/Applicant Name:</strong></td>
<td>Wonjang Baek</td>
</tr>
<tr>
<td><strong>Customer Number:</strong></td>
<td>49455</td>
</tr>
<tr>
<td><strong>Filer:</strong></td>
<td>James Garrett McEwen/Sabrina Maya</td>
</tr>
<tr>
<td><strong>Filer Authorized By:</strong></td>
<td>James Garrett McEwen</td>
</tr>
<tr>
<td><strong>Attorney Docket Number:</strong></td>
<td>03662.1006</td>
</tr>
<tr>
<td><strong>Receipt Date:</strong></td>
<td>05-SEP-2012</td>
</tr>
<tr>
<td><strong>Filing Date:</strong></td>
<td>18-FEB-2010</td>
</tr>
<tr>
<td><strong>Time Stamp:</strong></td>
<td>15:08:07</td>
</tr>
<tr>
<td><strong>Application Type:</strong></td>
<td>Utility under 35 USC 111(a)</td>
</tr>
</tbody>
</table>

### Payment information:

- Submitted with Payment: no

### File Listing:

<table>
<thead>
<tr>
<th>Document Number</th>
<th>Document Description</th>
<th>File Name</th>
<th>File Size(Bytes)/Message Digest</th>
<th>Multi Part / .zip</th>
<th>Pages (if appl.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Amendment_090512.pdf</td>
<td>4787201</td>
<td>453363f4e8be9f623e27e5ba02f1064e166e55481d8f63b0502f1064e166e55481d8f63b050</td>
<td>yes</td>
<td>44</td>
</tr>
<tr>
<td>Document Description</td>
<td>Start</td>
<td>End</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------------------------------------------</td>
<td>-------</td>
<td>-----</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fee Worksheet (SB06)</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Amendment/Req. Reconsideration-After Non-Final Reject</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Abstract</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specification</td>
<td>4</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Claims</td>
<td>5</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Applicant Arguments/Remarks Made in an Amendment</td>
<td>10</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specification</td>
<td>13</td>
<td>44</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Total Files Size (in bytes):** 4787201

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

**New Applications Under 35 U.S.C. 111**
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

**National Stage of an International Application under 35 U.S.C. 371**
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

**New International Application Filed with the USPTO as a Receiving Office**
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.
REPLY/AMENDMENT FEE TRANSMITTAL

Attorney Docket No. 0366.1006
Application Number 12/656,872
Filing Date February 18, 2010
First Named Inventor Wonjang Baek
Group Art Unit 2199

AMOUNT ENCLOSED $ 0.00
Examiner Name Van H NGUYEN

FEE CALCULATION (fees effective 10/02/08)

<table>
<thead>
<tr>
<th>CLAIMS AS AMENDED</th>
<th>Claims Remaining After Amendment</th>
<th>Highest Number Previously Paid For</th>
<th>Number Extra</th>
<th>Rate</th>
<th>Calculations</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOTAL CLAIMS</td>
<td>22</td>
<td>22 = 22</td>
<td>0</td>
<td>X</td>
<td>$60.00 = $0.00</td>
</tr>
<tr>
<td>INDEPENDENT CLAIMS</td>
<td>3</td>
<td>3 = 3</td>
<td>0</td>
<td>X</td>
<td>$250.00 = $0.00</td>
</tr>
</tbody>
</table>

Since an Official Action set an original due date of, a petition is hereby made for an extension to cover the date this reply is filed for which the requisite fee is enclosed (1 month ($150)); (2 months ($560)); (3 months ($1,270)); (4 months ($1,980)); (5 months ($2,690));

If Notice of Appeal is enclosed, add ($620.00) $ 0.00
If Statutory Disclaimer under Rule 20(d) is enclosed, add fee ($140.00) $ 0.00
Information Disclosure Statement (Rule 1.17(p)) ($180.00) $ 0.00

Total of above Calculations = $ 0.00
Reduction by 50% for filing by small entity (37 CFR 1.23 & 1.28) $ 0.00
Reduction by 75% for filing by micro entity (37 CFR 1.23(a)(1)) $ 0.00

TOTAL FEES DUE = $ 0.00

(1) If entry (1) is less than entry (2), entry (3) is "0".
(2) If entry (2) is less than 20, change entry (2) to "20".
(4) If entry (4) is less than entry (5), entry (6) is "0".
(5) If entry (5) is less than 3, change entry (5) to "3".

METHOD OF PAYMENT

☐ Check enclosed as payment. ☐ Credit Card Payment Form, Form PTO-2038(attached).
☐ Charge "TOTAL FEES DUE" to the Deposit Account No. below.
☐ No payment is enclosed and no charges to the Deposit Account are authorized at this time (unless specifically required to obtain a filing date).

GENERAL AUTHORIZATION

☒ If the above-noted "AMOUNT ENCLOSED" is not correct, the Commissioner is hereby authorized to credit any overpayment or charge any additional fees necessary to:

Deposit Account No. 503333
Deposit Account Name STEIN MCEWEN, LLP

☒ The Commissioner is also authorized to credit any overpayments or charge any additional fees required under 37 CFR 1.16 (filing fees) or 37 CFR 1.17 (processing fees) during the prosecution of this application, including any related application(s) claiming benefit hereof pursuant to 35 USC § 120 (e.g., continuations/divisionals/CIPs under 37 CFR 1.53(b) and/or continuations/divisionals/CPAs under 37 CFR 1.53(d)) to maintain pendency hereof or of any such related application.

SUBMITTED BY: STEIN MCEWEN, LLP

Typed Name James G. McEwen
Reg. No. 41,883

Signature

Date Sept. 5, 2012
**PATENT APPLICATION FEE DETERMINATION RECORD**

*Application Number:* 12/656,872  
*Filing Date:* 02/18/2010

**APPLICATION AS FILED – PART I**

<table>
<thead>
<tr>
<th>FOR</th>
<th>SMALL ENTITY</th>
<th>OTHER THAN SMALL ENTITY</th>
<th>SMALL ENTITY</th>
<th>OTHER THAN SMALL ENTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>RATE ($)</td>
<td>FEE ($)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>BASIC FEE (37 CFR 1.16(a), (b), or (c))</td>
<td></td>
<td></td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>SEARCH FEE (37 CFR 1.16(c), (l), or (m))</td>
<td></td>
<td></td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>EXAMINATION FEE (37 CFR 1.16(g), (p), or (q))</td>
<td></td>
<td></td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>TOTAL CLAIMS (37 CFR 1.16(i))</td>
<td>minus 20 = *</td>
<td></td>
<td>X $ =</td>
<td>OR</td>
</tr>
<tr>
<td>INDEPENDENT CLAIMS (37 CFR 1.16(h))</td>
<td>minus 3 = *</td>
<td></td>
<td>X $ =</td>
<td>OR</td>
</tr>
</tbody>
</table>

*If the difference in column 1 is less than zero, enter "O" in column 2.*

**APPLICATION AS AMENDED – PART II**

<table>
<thead>
<tr>
<th>AMENDMENT 09/05/2012</th>
<th>SMALL ENTITY</th>
<th>OTHER THAN SMALL ENTITY</th>
<th>SMALL ENTITY</th>
<th>OTHER THAN SMALL ENTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLAIMS REMAINING AFTER AMENDMENT</td>
<td>HIGHEST NUMBER PREVIOUSLY PAID FOR</td>
<td>PRESENT EXTRA</td>
<td>RATE ($)</td>
<td>ADDITIONAL FEE ($)</td>
</tr>
<tr>
<td>Total (37 CFR 1.16(i))</td>
<td>22 Minus ** 22</td>
<td>** 0</td>
<td>X $30 =</td>
<td>0</td>
</tr>
<tr>
<td>Independent (37 CFR 1.16(i))</td>
<td>3 Minus ** 3</td>
<td>** 0</td>
<td>X $125 =</td>
<td>0</td>
</tr>
</tbody>
</table>

**AMENDMENT**

<table>
<thead>
<tr>
<th>CLAIMS REMAINING AFTER AMENDMENT</th>
<th>HIGHEST NUMBER PREVIOUSLY PAID FOR</th>
<th>PRESENT EXTRA</th>
<th>RATE ($)</th>
<th>ADDITIONAL FEE ($)</th>
<th>RATE ($)</th>
<th>ADDITIONAL FEE ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total (37 CFR 1.16(i))</td>
<td>Minus ** 0</td>
<td>**</td>
<td>X $ =</td>
<td>OR</td>
<td>X $ =</td>
<td></td>
</tr>
<tr>
<td>Independent (37 CFR 1.16(i))</td>
<td>Minus ** 0</td>
<td>**</td>
<td>X $ =</td>
<td>OR</td>
<td>X $ =</td>
<td></td>
</tr>
</tbody>
</table>

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.

**LEGAL INSTRUMENT EXAMINER:** BONNIE COLE/

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS: SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.
Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

docketing@smiplaw.com
### Office Action Summary

<table>
<thead>
<tr>
<th>Application No.</th>
<th>Applicant(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/656,872</td>
<td>BAEK ET AL.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Examiner</th>
<th>Art Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>VAN NGUYEN</td>
<td>2199</td>
</tr>
</tbody>
</table>

---

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If a period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication. Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).

Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1) **☒** Responsive to communication(s) filed on **18 February 2010**.

2a) ☐ This action is FINAL.

2b) **☒** This action is non-final.

3) ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

4) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

5) **☒** Claim(s) 1-22 is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6) ☐ Claim(s) _____ is/are allowed.

7) **☒** Claim(s) 1-22 is/are rejected.

8) ☐ Claim(s) _____ is/are objected to.

9) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

10) ☐ The specification is objected to by the Examiner.

11) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.

   Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

   Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

12) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

13) **☒** Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

   a) ☒ All  b) ☐ Some *  c) ☐ None of:

   1. ☒ Certified copies of the priority documents have been received.

   2. ☐ Certified copies of the priority documents have been received in Application No. _____.

   3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

   * See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1) **☒** Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson’s Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08)

   Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)

   Paper No(s)/Mail Date _____.

5) ☐ Notice of Informal Patent Application

6) ☐ Other: ______.
DETAILED ACTION

1. This action is responsive to the application filed 02/18/2010.

Claims 1-22 are presented for examination.

Priority

2. Receipt is acknowledged of papers submitted under 35 U.S.C. 119(a)-(d), which papers have been placed of record in the file.

Specification

3. The specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

The use of trademarks has been noted in this application. Trademarks should be capitalized wherever they appear and be accompanied by the generic terminology.
Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in a manner which might adversely affect their validity as trademarks.

To expedite correction on this matter, the examiner suggests the following guidelines for Applicant to follow in amending the specification: capitalize each letter of a trademark or accompany the trademark with an appropriate designation symbol, e.g., \textsuperscript{TM} or \textcopyright, as appropriate.

**Claim Rejections - 35 USC § 102**

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless -

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.
Claims 1-22 are rejected under 35 U.S.C. 102(b) as being anticipated by Horii et al. (US 20060080474).

As to claim 1:

Horii teaches a method for managing java applications (Java applications; see paragraphs 0247-0273 and 0291-0316) executable in a user device, the method comprising steps of:

(a) issuing a request for executing a second java application (Java applications /second Java application), the request being issued by a first java application (Java applications/ first Java application) and including a execution type information (control information of the Java applications);

(b) changing a state (state of Java application... transitions between the operating states) of the first java application according to the execution type information;

(c) executing the second java application (executes the corresponding Java application); and

(d) killing (kill...Java application) the second java application when a request for killing the second java application is issued by the second java application and re-changing (transitions between the operating states...the operating state of the Java application transits to “Active”...transits to “Paused”...transits to “Destroyed”) the state of the first java application changed in the step (b).

As to claim 2:

Horii teaches the first java application generates the request for executing the
second java application based on a user input received through a user interface

(see paragraphs 0247-0273).

As to claim 3:

Horii teaches the request for executing the second java application further includes a path for obtaining the second java application (see paragraphs 0247-0273).

As to claim 4:

Horii teaches executing the second java application by obtaining the second java application at a location defined by the path (see paragraphs 0247-0273).

As to claim 5:

Horii teaches the execution type information designates, among other thing, a first execution type wherein the first java application is killed when the second java application is executed and the first java application is restarted when the second java application is killed (see paragraphs 0247 and 0243).

As to claim 6:

Horii teaches killing the first java application when the execution type information designates the first execution type; pausing the first java application when the execution type information designates the second execution type; and switching the first java application to background when the execution type
information designates the third execution type (see paragraphs 0250-0267).

As to claim 7:
Horii teaches restarting the first java application when the execution type information designates the first execution type; unpausing the first java application when the execution type information designates the second execution type; and switching the first java application to foreground when the execution type information designates the third execution type (see paragraphs 0250-0267).

As to claim 8:
Horii teaches the first java application and the second java application inter-relays a shared information (see paragraphs 0248-0261).

As to claim 9:
Horii teaches the first java application and the second java application inter-relays a shared information when the execution type information designates one of the second execution type and the third execution type (see paragraphs 0248-0261).

As to claim 10:
Horii teaches each of the first java application and the second java application includes a java application of Xlet format (see paragraphs 0250-0253).
As to claim 11:

The rejection of claim 1 above is incorporated herein in full. Horii further teaches (b) pausing the first java application (see paragraphs 0251-0267); (d) killing the second java application when a request for killing the second java application is issued by the second java application and unpausing the first java application paused in the step (b) (see paragraphs 0301-0309 and 0314-0323).

As to claims 12-16:

Refer to claims 2-4, 8, and 10 above, respectively, for rejections.

As to claim 17:

The rejection of claim 1 above is incorporated herein in full. Horii further teaches (b) switching the first java application to background (see paragraphs 0261-0262 and 0272); (d) killing the second java application when a request for killing the second java application is issued by the second java application and switching the first java application in background to foreground (see paragraphs 0301-0309 and 0314-0323).

As to claims 18-22:

Refer to claims 2-4, 8, and 10 above, respectively, for rejections.
Conclusion

5. The prior art made of record, listed on PTO 892 provided to Applicant is considered to have relevancy to the claimed invention. Applicant should review each identified reference carefully before responding to this office action to properly advance the case in light of the prior art.

Contact Information

6. Any inquiry or a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: (571) 272-2100.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to VAN H. NGUYEN whose telephone number is (571) 272-3765. The examiner can normally be reached on Monday- Friday from 9:00AM- 5:30 PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner’s supervisor, LEWIS BULLOCK can be reached at (571) 272-3759.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.
Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/VAN H NGUYEN/
Primary Examiner, Art Unit 2199
**Notice of References Cited**

**U.S. PATENT DOCUMENTS**

<table>
<thead>
<tr>
<th>*</th>
<th>Document Number</th>
<th>Date</th>
<th>Name</th>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>A US-2006/0080474</td>
<td>04-2006</td>
<td>Horii et al.</td>
<td>710/008</td>
</tr>
<tr>
<td>*</td>
<td>C US-7,600,045</td>
<td>10-2009</td>
<td>Horii et al.</td>
<td>709/246</td>
</tr>
</tbody>
</table>

**FOREIGN PATENT DOCUMENTS**

<table>
<thead>
<tr>
<th>*</th>
<th>Document Number</th>
<th>Date</th>
<th>Country</th>
<th>Name</th>
<th>Classification</th>
</tr>
</thead>
</table>

**NON-PATENT DOCUMENTS**

Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages

| * | 
|---|---|

---

* A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).) Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.
Did you mean: (KILL OR STOP OR TERMINATE) AND "JAVA APPLICATION" AND XLET AND BACKGROUND AND FORGROUND

Method for managing java applications


... for executing 2nd java application r switch 1st java application to background r obtain and execute
2nd java application r kill 2nd java ... It is preferable that each of the first java application and the second java application include the java application of Xlet format executable in ...
All 2 versions - Import into BibTeX

Method of Maintaining Applications in a Computing Device


... control of the underlying operating system, which can suspend or resume their operation, or even kill an application. However, the Java AMS can only manage the entire lifecycle of an application model (mainly in xlets) ... Hence, the non native Java application shown in FIG. ...
All 2 versions - Import into BibTeX

PLAY BACK APPARATUS, PLAYBACK METHOD AND PROGRAM FOR PLAYING BACK 3D VIDEO


... Decoders Subtitle Graphics Subtitle in the Output Video WHO Video Background JPEG image 11 3D display mode Interactive Graphics (L) (R) / ~10 Decoders Subtitle Graphics Subtitle N or 2D ~10 Decoders Subtitle Graphics Subtitle N or 2D ~10 Decoders Subtitle Graphics Subtitle N or 2D ~10 Decoders Subtitle Graphics Subtitle N or 2D ~10 Decoders Subtitle Graphics Subtitle N or 2D ~10 Decoders Subtitle Graphics Subtitle N or 2D ~10 Decoders Subtitle Graphics Subtitle N or 2D ~10 Decoders Subtitle Graphics Subtitle N or 2D ~10 Decoders Subtitle Graphics Subtitle N or 2D ~10 Decoders Subtitle Graphics Subtitle N ... Cited by 1 - All 2 versions - Import into BibTeX

REPRODUCTION DEVICE, INTEGRATED CIRCUIT, AND REPRODUCTION METHOD CONSIDERING SPECIALIZED REPRODUCTION


... show foreground pixel data and background pixel data before and after subtitles are shifted to the right and to the left FIGS. ... The application is, for example, a Java™ application, and is composed of one or more xlet programs loaded in the heap area (also ... Cited - Import into BibTeX

PLAY BACK APPARATUS, PLAYBACK METHOD, AND PROGRAM FOR PERFORMING STEREOSCOPIC PLAYBACK


... 5 2D display mode Interactive Graphics 10 Decoders Subtitle Output Video Video Background JPEG image 11 ON 3D display mode Interactive Graphics (L) (R) / ~10 Decoders Subtitle Graphics Subtitle K or 2D r right Subtitle I 11 e_2 y (R) J Video Video N, O, ... All 2 versions - Import into BibTeX

PLAYBACK DEVICE, PLAYBACK METHOD AND PROGRAM ... (54) PLAYBACK DEVICE, PLAYBACK METHOD AND PROGRAM ...

All 2 versions - Import into BibTeX

Recording medium, play apparatus, recording method, and playback method

W. Bierk - EP Patent 1,844,731, 2001 - freepatentsonline.com

... Living in the work memory refers to the xlet programs composing the application having been ... table makes it possible to perform a synchronization control to run a Java application during a ... Cited by 1 - Related articles - Cached - All 2 versions - Import into BibTeX

Reproduction device, program and reproduction method


... Living in the work memory refers to the xlet programs composing the application having been ... table makes it possible to perform a synchronization control to run a Java application during a ... The AV playback functions include: Play, Stop, Pause On, Pause Off, Still Off, Forward ... Cited - Import into BibTeX

Play apparatus and method, recording method and recording medium

EP Patent 2,267,711, 2010 - freepatentsonline.com

... Living in the work memory refers to the xlet programs composing the application having been ... table makes it possible to perform a synchronization control to run a Java application during a ... The AV playback functions include: Play, Stop, Pause On, Pause Off, Still Off, Forward ...
Cited - Import into BibTeX
Recording medium, playback apparatus, recording method, and playback method
... Living in the work memory refers to the xlet programs composing the application having been ...
... table makes it possible to perform a synchronization control to run a Java application during a ...
The AV playback functions include: Play, Stop, Pause On, Pause Off, Still Off, Forward ...
Cached - Import into BibTeX

PLAYBACK DEVICE CAPABLE OF STEREOSCOPIC PLAYBACK, PLAYBACK METHOD, AND PROGRAM
... The application is a Java™ application, and is composed of one or more xlet programs loaded ...
The application is composed of the xlet program loaded in the work memory and data ... used in ...
DVD players and CD players, and includes playback start, playback stop, pause, release ...
Cached - Import into BibTeX

REPRODUCTION DEVICE, RECORING MEDIUM, AND INTEGRATED CIRCUIT
... videos [Background Art]. ... directory. An application is, for example, a Java® application containing one or more xlet programs that have been loaded onto a ...
Cached - Import into BibTeX

RECORDING MEDIUM, REPRODUCTION DEVICE, SYSTEM LSI, REPRODUCTION METHOD, SPECTACLE, AND DISPLAY DEVICE ASSOCIATED WITH 3D VIDEO
... images. [Background Art]. ... directory. An application may be, for example, a Java™ application that is composed of one or more xlet programs having been loaded into a ...
Cached - Import into BibTeX

RECORDING MEDIUM, PLAYBACK DEVICE, SYSTEM LSI, PLAYBACK METHOD, GLASSES, AND DISPLAY DEVICE FOR 3D IMAGES
... 30, 2008. BACKGROUND OF THE INVENTION. (1) Field of the Invention. ... An application may be, for example, a Java™ application that is composed of one or more xlet programs having been loaded into a ...
Cached - Import into BibTeX

Recording medium, playback device, integrated circuit, playback method
... images. [Background Art]. ... directory. An application may be, for example, a Java™ application that is composed of one or more xlet programs having been loaded into a ...
Cached - Import into BibTeX

REPRODUCTION DEVICE, REPRODUCTION METHOD, AND PROGRAM FOR STEREOSCOPIC REPRODUCTION
... The Java™ application for controlling playback instructs a Java™ virtual machine to generate a ...
Cached - Import into BibTeX

Create email alert

Did you mean to search for: (KILL OR STOP OR TERMINATE) AND "JAVA APPLICATION" AND XLET AND BACKGROUND AND FOREGROUND
Did you mean: (KILL OR STOP OR TERMINATE) AND JAVA AND XSLT

Cinga-NCL: the declarative environment of the Brazilian digital TV system
LF G Soares, RP Rodrigues... - Journal of the Brazilian... 2007 - SciELO Brasil
... etc.), audio objects (MP3, WMA, etc.), text objects (TXT, PDF, etc.), execution objects (Xlet, Lua, etc.) a browser code through datacasting and install it as a plug-in (usually a Java plug-in). Events may generate actions on other media objects, like to start or stop their presentations ...
Cited by 132 - Related articles - Cached - All 42 versions - Import into BibTeX

Method for managing Java applications
... request for executing 2nd java application r switch 1st java application to background r obtain and execute 2nd java application r kill 2nd java ... [2005] Preferably, each of the first java application and the second java application includes a java application of Xlet format ...
All 2 versions - Import into BibTeX

Playback of audio-video content and an associated Java application from an optical disc
... 240 back into paused state 320 by a call to a pause method 265 or kill active Java ... In paused state 320, Java object 220-240 must stop providing any service and might release resources ... A/V control may be performed after the start method and may be held until Java object 220 ...
Cached - Import into BibTeX

Java lifecycle model for BD-discs
... 240 back into paused state 320 by a call to a pause method 265 or kill active Java ... In paused state 320, Java object 220-240 must stop providing any service and might release resources ... An control may be performed after the start method and may be held until Java object 220 ...
All 2 versions - Import into BibTeX

Interactive Television in Brazil: System Software and the Digital Divide
LF G Soares... - Proc. of EuroTV, 2007 - tvd.inf.br
... Events may generate actions on other media objects, like to start, pause or stop their presentations ...
typical Home Area Networks technologies (Bluetooth, WiF, PLC etc.;), finally the third one provides support for controlling the presentation of NCL documents through Java Xlets ...
Cited by 21 - Related articles - View as HTML - All 11 versions - Import into BibTeX

Method for managing applications related to data broadcasting, class/interface structure for controlling the same, and broadcast receiver for controlling the class/
TJ Park, JY Kwak... - US Patent 7,553,342, 2010 - Google Patents
... The above-mentioned "App Proxy" determines whether the calling application has an authority to pause, destroy, or stop a corresponding application. ... FIG. 3 exemplarily shows a life cycle of a Java program (Xlet). Referring to FIGS ...
Cited by 1 - Related articles - All 5 versions - Import into BibTeX
Method for managing applications related to data broadcasting, class/interface structure for controlling the same, and broadcast receiver for controlling the class/...

TJ Park, SY Kwon... - EP Patent 1,818,552, 2007 - freespatentsonline.com

... The above-mentioned "AppProxy" determines whether the calling application has an authority to pause, destroy, or stop a corresponding application. FIG. 3 exemplarily shows a life cycle of a Java program (Xlet). Referring to FIGS...}

Cached - Import into BibTeX

**Software Lifecycle Management in Java Environments**

A. Hauptmann - ox-informatik.uni-rostock.de

... Xlets are supported in Personal Basis Profile (PBP) and in its superset, the Personal Profile (PP).... Start/Stop a bundle. ... All applications in a bundle are started in the same JVM (Java Virtual Machine) for efficiency reasons (memory, resources, CPU cycles).... Update a bundle. ...

Related articles - View as HTML - All 2 versions - Import into BibTeX

Manageability of Java-based digital TV receivers

B. Manczka... - Areas in Communications, IEEE Journal on, 2006 - ieeexplore.ieee.org

... OF JAVA-BASED DIGITAL TV RECEIVERS 801 Fig. 6. Xlet and Xlet context, Fig. 7. Service selection component presenter/application hierarchy, Fig. 8. Xlet lifecycle based on ITU-T X.731. service. The result of calling its suspend() method is identical to calling its stop() method...

Cited by 3 - Related articles - BL Direct - All 3 versions - Import into BibTeX

Towards dynamic component isolation in a service oriented platform

K. Gama... - Component-Based Software Engineering, 2009 - Springer

... The Xlet application model resembles Java applets and MIDlets, providing also small applications with life cycle (init, start, pause, destroy).... communication between Xlets with the Inter Xlet Communication (IXC) mechanism, which uses a subset of Java RMI...

Cited by 6 - Related articles - All 7 versions - Import into BibTeX

**DOWNLOAD OF PROGRAMS INTO BROADCAST-RECEIVERS**

Y Wei... - EP Patent 1,800,050, 2006 - freespatentsonline.com

... 440 provides the system-level support needed to implement the Java VM and the Java packages. ... Xlet by pressing the radio button in row 510 and column 590, the Xlet will be ... KILL: The destroy method will be called by the application manager when the control code changes...

Cached - Import into BibTeX

**DVd virtual machine**


... provides the system-level support needed to implement the Java VM and the Java packages. ... The soft ware layers, including the Xlets, may be stored in a repro- grammable memory. ... number, Time)—Play title at a specific time [0049] PTT_Search (PIT number)—stop current pre...

All 2 versions - Import into BibTeX

**Beginnig Java ME Platform**

R. Rischpater... - 2008 - books.google.com

... Data Using the Record Store .....136 CHAPTER 7 Accessing Files and Other Data .....161 CHAPTER 8 Using the Java Mobile Game API .....193 Intermediate PART 3... CDC Development CHAPTER 9 Introducing Xlets and the...

Cited by 9 - Related articles - Library Search - All 18 versions - Import into BibTeX

Downloading of programs into broadcast-receivers


... The Real Time Operating System (RTOS) 440 provides the system-level support needed to implement the Java VM and the Java packages. ... a down-load for the seeker training Xlet by pressing the radio button in row 510 and column 590, the Xlet will be ... KILL: The destroy...

All 2 versions - Import into BibTeX
Method for obtaining context of corresponding Xlet while playing BD-J title
TJ Chen, W Li... US Patent 6,015,548, 2011 - Google Patents

... Java application format for MHP (Multimedia Home Platform) and related digital TV standards
gives us something pretty close to what we need. Like applet, the Xlet interface allows an external
source (the application manager in the case of a digital TV receiver) to start and stop ...
Related articles - All 5 versions - Import into BibTeX

Xlet-based, ESG (Integrated Electronic Service Guide) in ATSC-ACAP
H Jang... Software Engineering Research, 2006 - ieeexplore.ieee.org

... to generate bit streams of program events such as event name, start and stop time and ...
As described in java tv xlet, the application execution schedule is controlled by 4 status including ...
JMF (Java Media Framework) version 1.1 is an API to control audio/video streaming data ...
Cited by 1 - Related articles - All 3 versions - Import into BibTeX

Giropa: J-Open Java Based Application Environment for Interactive Digital Television Services

... of the JMF API, since it provides basic reproduction functionalities for the Java API through the ...
It is called ApplicationProxy, which offer the control of the applications lifecycle (start, stop, pause,
resume... that has the capabilities to call the antiJavaMain() function to start the a xlet ...
Related articles - All 2 versions - Import into BibTeX

Usages of a SML player in digital television
JL Larradon, P Cesar, C Herrera... 7th IASTED International, 2009 - books.google.com

... DVB-Java applications are called Xlets, which can be resident or downloadable. The set-top
box's application manager downloads and controls the life cycle of the Xlets (ie, initialise, start, ...
stop and destroy). Figure 2 Application manager Set-top box Resources Xlet ...
Cited by 7 - Related articles - View as HTML - All 3 versions - Import into BibTeX

Embedded Java security: Security for mobile devices
M Debord... 2007 - books.google.com

... mechanism. The main difference with the applet application model is that the Xlet
application model does not have implicit API requirements like java applet, which allows
it to be used in a greater variety of product scenarios. Page 20 ...
Cited by 11 - Related articles - Library Search - All 6 versions - Import into BibTeX

Dynamic application frameworks using OSGi and Bea none
H Caramantes, D Dones... Proceedings of the International, 2012 - 143.206.48.174

... or updated software components, coarse- and/or fine-grained extensibility of functionality (via
dynamically loadable code), potentially non-stop execution, the... In this context, an application
(i.e., Java Xlet) must be deployed with minimum intervention of the content provider ...
Cited by 3 - Related articles - View as HTML - All 6 versions - Import into BibTeX

Xlet-based, IESG (Integrated Electronic Service Guide) in Ubiquitous Interactive TV
Environment
HY Jiang... WCNC, 2007 - Cleaner

... to generate bit streams of program events such as event name, start and stop time and ...
As described in java tv xlet, the application execution schedule is controlled by 4 status including ...
JMF (Java Media Framework) version 1.1 is an API to control audio/video streaming data ...
Cited by 1 - Related articles - View as HTML - All 5 versions - Import into BibTeX
Home control system over an MHP based architecture
O. Mirabella, M. Bischofberger, ... - System Interactions, 2005, 2006 - ieeexplore.ieee.org
... Both the red and green communication, as well as the Stop All and Stop This, allowing the user to stop the communication, of a package called Hac which includes all the Java classes reported in TABLE I. The main class Hac.java implements the Xlet interface. ...
Cited by 1 - Related articles - Import into BibTeX

Data broadcast processing device, method and program
... as NCL objects, and have states such as start, stop, pause, and resume. [0041] The Lua script 72 and the Java(R) library 74 are capable of operating each other, by defining a <link> element, which represents a reference to a <media> element in which Xlet supporting Ginga-J ...
All 2 versions - Import into BibTeX

Ginga Game: A Framework for Game Development for the Interactive Digital Television
DC Barboza, ... - Games and Digital Entertainment ( ... - ieeeexplore.ieee.org
... for the Digital TV Applications for the Digital TV are called Xlets, just like Java applications for the ... difference between an Xlet and an Applet is the possibility to pause and resume an Xlet. ... This way, it's possible to temporary stop an application that is not visible and release the ...
Cited by 2 - Related articles - All 9 versions - Import into BibTeX

System and method Xlet and applet manager testing
... That is, the applet manager 110 determines when to load, start, and stop each applet 112 ... for small devices is developed using Java 2 Micro Edition (J2ME), which is an edition of Java 2 for ...
One type of program often used in J2ME is the Xlet, which is a program similar to an ...
Cited by 1 - Related articles - All 4 versions - Import into BibTeX

Design and Implementation of an MHP Demonstration System
W.W. Mai,...
Related articles - All 2 versions - Import into BibTeX

A wrapper and broker model for collaboration between a set-top box and a home service gateway
G. Li, P. Wang, ... - Consumer Electronics, IEEE ..., 2006 - ieeexplore.ieee.org
... A similar flow of operations is triggered as a request from an OSGi bundle to stop an MHP Xlet.
Fig. 3 After entering the Prefetch state, the MHPAppManager handles the lifecycle for each Xlet.
The Application Information Table (AIT) can be retrieved by JNI (Java Native Interface).
Cited by 18 - Related articles - All 4 versions - Import into BibTeX

Automating Acceptance Testing of Digital Television Applications with EasyAccept DTV
MAS Cuello, JF Damanio, ... - ibd.doc.uvic.ca.br
... are some similarities between the java.lang.Xlet Xlet and the java.applet.Applet services. Xlets have methods that allow them to initialize, to start and to stop such as the Applets. There are some major differences, however, between them. The main one is that an Xlet can also be ...
Related articles - All 5 versions - Import into BibTeX

Exploiting OSGi capabilities from MHP applications
PPG Redon, AF Xasa, M. Cabot ... - Journal of Virtual Reality, 2007 - citeseer
... to the activation and deactivation order, whatever deactivated order received (Xlet.pause() or bundle.stop()), the other ... a method reference and all the input and re-turn parameters are of type Java object, so that The wrapped code acts as an interface between the caller Xlet and the code which, in ...
Cited by 17 - Related articles - View as HTML - All 4 versions - Import into BibTeX

Xlet Resource Estimation
C. Robert, ... - 2004 - citeseer
... As already mentioned, DVB-J applications, so called Xlets, are Java programs. Java programs...
usually have full control of their lifecycle, eg they may exit the JVM. This conventional Java application model does not work well for MHP. If a single Xlet would 12 Page 23. 2.3. ... Related articles - View as HTML - All 2 versions - Import into BibTeX

Real-time carousel caching and monitoring in data broadcasting
DH Park, TY Ku... - Consumer Electronics, IEEE ... 2006 - iseexplore.ieee.org
... WORKS A data broadcasting application generally consists of some applications, typically java applications, which ... the transmission of broadcasting objects such as the file and directory of Xlet in the ... the broadcaster to send AV related commands such as pause, play and stop. ... Cited by 6 - Related articles - BL Direct - All 4 versions - Import into BibTeX

(book) Enterprise J2ME: developing mobile Java applications
J.J. Yinyu... - 2004 - books.google.com
Page 1. Foreword by Jim Colson IBM Distinguished Engineer Enterprise ^^^^A • ^^^^A • ^^^^A • ^^^^A • DEVELOPING MOBILE JAVA APPLICATIONS UI Page 2 ... vI Contents 1.5 Summary 13 Resources 13 2 J2ME: Is Mobile Java Ready for Enterprise? 15 2.1 Why Java? ... Cited by 137 - Related articles - Library Search - All 10 versions - Import into BibTeX

(book) Java development on PDAs: building applications for PocketPC and Palm devices
D. Wilding-McBride - 2003 - books.google.com

(prof) Programming Java 2 Micro Edition on Symbian OS
M. De Jode, J. Allen, G. Holland, ... John Wiley & ... 2004 - pointlessprophecy.com

(prof) A T-learning Courses Development and Presentation Framework
F.Gelbukh, S.Vernikov, E.Patros, P.Loucas ... - IEEE Multidisciplinary ... 2006 - Ciseeser
... The Xlet that contains the t-learning course runs on the STB MHP middleware presenting the content of the ... From a software point of view, Game Templates is a hierarchy of Java classes ... le loading, starting, managing its exit and possibly forcing a pause and/or stop), according to ... Cited by 8 - Related articles - View as HTML - All 7 versions - Import into BibTeX

(prof) DTV APPLICATION SOFTWARE ENVIRONMENT LEVEL 1 (DASE-1) PART 3: PROCEDURAL APPLICATIONS AND ENVIRONMENT
A Standard - 2005 - alsb.org
... and restricted by this specification. Furthermore, any DASE application which employs a Java TV Xlet shall maintain and report Xlet state and status information as described in the following subsections. 4.1.1 State Attributes ... Related articles - All 4 versions - Import into BibTeX

Developing applications for EBIF and tv2way™
RM Arlens, RG Gaetanello, D Lib... ... (EBMSS), 2010 IEEE ... 2010 - iseexplore.ieee.org
... the commercial STB. Before testing on the STB we run the application as a Java application which ... from the point of view of the STB, the scheduler is the only XLET it ever ... The simulator can simulate various platforms and can schedule the application to start and stop execution at ... Import into BibTeX

Web-based television
... are a video elementary stream, an audio elementary stream, a Java application (Xlet), or other ...

PDF from pointlessprophecy.com

PDF from psu.edu

PDF from alsb.org
the content received by the receiver), for example similar to a Java Virtual Machine. ... Time)—Play title at a specific time [0054] PTT Search (PTT number)—Stop current presentation ...
All 2 versions - Import into BibTeX

**ProQ: CISMUNDUS: terminal hardware middleware and applications**

S Batterfield, D Garenc, S Cosmadis... - 4th Conference on ..., 2003 - data.batmel.ac.uk
... from the newly selected DOM node is extracted and encapsulated within Java AWT.Event ... The CM then drives the Decoder Set with this information to start stop playing of ... During the system initialisation phase, the MHPs application manager launches the default/daemons Xlet ... Cited by 2 - Related articles - View as HTML - All 3 versions - Import into BibTeX

**METHOD FOR PERMITTING AN EXCHANGE OF DATA BETWEEN AN XLET-BASED AND A BUNDLE-BASED DATA SET VIA A BUNDLE-BASED INTERFACE**

R Mant, Pelt, ... - EP Patent 1,500,670, 2005 - freepatentsonline.com
... Wenn ein erweiterter Xlet-basierter Datensatz 11 Xlet-Bundle den in der DAB-Java-Spezifikation vorgesehenen Eintrag ... Gemäß Figur 6a folgt zum Beenden der Funktion eines erweiterten Xlet-basierter Datensatzes 11 'Xlet-Bundle eine Methode 37 stop() aufgrund derer ...
Cached - Import into BibTeX

**System and Method for Reducing the Start-up Time of MHP Applications**

... The pause Xlet signals the Xlet to stop providing service and enter the paused state ... when an MHP application is started the MHP system will execute the following steps: (1) load the Xlet class, see label 51 (ie the Java class implementing the Xlet interface, AnXlet ...
All 2 versions - Import into BibTeX

**Controlling the smart home from TV**

MP Celebre, RPD Henderso, AF Yung ... Consumer ... 2006 - ieeeexplore.ieee.org
... Similarly to the activation order, whatever deactivation order received (Xlet.pauseXlet or bundle stop): the other ... method reference and all the input and return parameters are of type Java object, so ... If the hostile-atmosphere application is broadcast as an Xlet (filmXlet), it uses the ... Cited by 42 - Related articles - All 4 versions - Import into BibTeX

**SECONDARY JAVA: HEAPS IN SHARED MEMORY**

... 28, 2011 Sheet 4 of 5 US 2011/0165129 A1 Stop activity on JV M processors ... may be passed either through another shared object or through serialization (eg, Inter Xlet communication (XITC)). ... In one or more embodiments of the invention, the JAVA® class object data defines the ...
All 2 versions - Import into BibTeX

**Controlling digital TV set-top box with mobile devices via an IP network**

SH Lo, CC Lin ... Consumer Electronics, IEEE ... 2006 - ieeeexplore.ieee.org
... The design of the server program must follow the Java Xlet architecture which is the standard interface defined for MHP applications. ... 939 content within a transport stream is described. Command 4 is to record a service. Command 5 is to stop recording the current service ...
Cited by 19 - Related articles - BL Direct - All 9 versions - Import into BibTeX

**A Technological Framework for the Authoring and Presentation of T-learning Courses**

F Bertolini, S Viarchidis, S Rasm... - International Journal ... 2008 - mshlab.fmi.gr
... GameTemplate is a Java class that abstracts general game functionalities: ... included (ie loading, starting, managing its exit and possibly forcing a pause and/or stop), according to ... encoder while the content produced by the authoring tool and the Course Player Xlet are inserted ...
Cited by 2 - Related articles - View as HTML - All 2 versions - Import into BibTeX

**Variable handling in time-based XML declarative languages**

LFD Gozanos, P Rodriguez, R Cresp... - Proceedings of the ... 2008 - dl.acm.org
... These are common situations found in a DTV environment where, in addition, it is very usual
to stop a presentation and then to ... whose name and semantics are defined in the NCL profile specification [1]. 5 Lua is the Giga scripting language, while Xlet is the Java code that ...
Cited by 6 · Related articles · All 6 versions · Import into BibTeX

Classification and evaluation of middleware collaboration architectures for converging MHP and OSGi in a smart home
CL Liu, FC Wang, ... · Journal of Information Science and ... · 2009 · its.sinica.edu.tw
... (a) and (b). Each Xlet has its own classloader, and it cannot invoke in- stances of other Xlets. But MHP 1.x specification [9] allows Inter Xlet Communication (IXC). IXC uses Java RMI to allow each Xlet to be accessible by other Xlet(s) loaded by another classloader. Fig. ...
Cited by 2 · Related articles · View as HTML · All 7 versions · Import into BibTeX

Remote Managing an Application on a Device by a Management Server
Y Royon... · US Patent 20,102,954,328, 2012 · freepatentonline.com
... Google Android allows developers to write managed code in the Java language, controlling the device via Google-developed Java libraries, etc. For instance, Xlets supports start and stop semantics, but it does not support install and uninstall semantics, whereas, Linux ...
Cached · Import into BibTeX

Object oriented communication between isolates
N Fresko... · EP Patent 1,785,860, 2007 · freepatentonline.com
... encapsulates an application or component, and provides the means to start and stop an Isolated ... to the registry and from the registry to the importing Xlet using JAVA™ object serialization ... The application manager can kill applications in response to a user’s request, for example ...
Related articles · Cached · Import into BibTeX

JAVA CONDITIONAL ACCESS APPARATUS
Y Kawakami... · US Patent Appl. 11/815,122, 2006 · Google Patents
... 12, 2009 (P4) JAVA CONDITIONAL ACCESS APPARATUS Publication Classification (75) Inventors: Yoshio Kawakami, Osaka (JP); Takekazu Shiono, Osaka (JP) Correspondence Address: GREEBLUM & BERNSTEIN, PLC 1950 ROLAND CLARKE PLACE RESTON, VA ...
All 2 versions · Import into BibTeX

Remotely managing an application on a device by a management server
Y Royon... · EP Patent 2,047,032, 2010 · freepatentonline.com
... Google Android allows developers to write managed code in the Java language, controlling the device via Google-developed Java libraries, etc. For instance, Xlets supports start and stop semantics, but it does not support install and uninstall semantics, whereas, Linux ...
Cached · Import into BibTeX

DIGITAL BROADCASTING APPARATUS AND METHOD FOR PROCESSING KEY SIGNAL
... key signal to the main screen A. For example, the received key signal is a signal output from a blue button requesting a temporary stop of the Xlet ... 9. The apparatus as claimed in claim 1, wherein the appli- cation programs comprise an Xlet executed by a Java Appli- cation ...
All 2 versions · Import into BibTeX

Method for providing record information in a digital broadcast receiver and a digital broadcast receiver for providing record information
PROVIDING A UNIFORM USER EXPERIENCE


No. US 2011/0258592 A1 Hindle et al. (43) Pub. Date: Oct. 20, 2011 (54) PROVIDING A UNIFORM USER RELATED US Application Data EXPERIENCE (60) Provisional application No. ...

All 2 versions - Import into BibTeX

Method and a digital broadcast receiver for providing a list of records

G. Raafat, S. Spinellis, E. Gambhi ... - isiscram.org

... MHP allows either the network operator to control when an application should start and stop (telling the receiver whether it should execute ... MHP has chosen DSM-CC as the file-based data broadcast protocol. Xlet Java classes and resource files are structured as a file ...

Related articles - View as HTML - Import into BibTeX

Data broadcasting software architecture supporting real-time caching and monitoring in interactive TV

CH Park, TY Ju ... - Computer and Information Sciences ... 2008 - iseeplore.isee.org

... Interactive TV generally consists of some applications, typically java applications, which are transmitted together ... defines the transmission of broadcasting objects like file and directory of Xlet in the ... allow the broadcaster to send A/V related commands like pause, play and stop ...

Cited by 2 - Related articles - All 4 versions - Import into BibTeX

Object oriented communication between isolates

N. Freksa ... - US Patent 7,765,562, 2010 - Google Patents

... class, which encapsulates an application or component, and provides the means to start and stop an isolated ... It is therefore should not be construed as being limited to JAVA. Furthermore, the term, Xlet, is intended to refer to a type of object-oriented application that can ...

Related articles - All 5 versions - Import into BibTeX

VELOCITY STABILIZATION FOR ACCELEROMETER BASED INPUT DEVICES


... Acceleration 0 Measured Deceleration Estimated Velocity Decelerating Acceleration is Zero + Time True Stop Velocity is Constant True Stop FIG. ... The unit of executable software may run in a predetermined environment; for example, a downloadable Java Xlet™ which runs ...

All 2 versions - Import into BibTeX

Automation apparatus and methods


... modifications to the user's premise. [0009] Furthermore, many users of such prior art systems simply stop using them due to the high level of complexity and expertise required for proper operation. This problem is especially ...

Cited by 9 - Related articles - All 2 versions - Import into BibTeX

A Mixed XML-javabeans approach to developing T-learning applications for the multimedia home

A Mixed XML-javabeans approach to developing T-learning applications for the multimedia home

[PDF] from psu.edu
platform
A Lopez-Heras, A Fernandez-Villar, ... Multimedia on Next, 2003 - Springer
... It is built as an xlet that, when initialized, reads the name of an XML configuration file from ... that, when they are not visible, they automatically minimize their memory needs and stop any running ... It is done by using the Java reflection mechanism and has low performance costs. ...
Cited by 17 - Related articles - BL Direct - All 9 versions - Import into BibTeX

MHP / OSGI convergence: a new model for open residential gateways
&k. Vitiag, RP Diaz Redondo, ... Software: Practice, 2006 - Wiley Online Library
... Bundle stop Bundle.start Figure 2. Lifecycle management in the MHP and the OSGI: (a) the Xlet lifecycle; (b) the bundle lifecycle. Dashed lines represent automatic or pseudo-transitions. When the initial Java class of an Xlet is loaded and instantiated (either from the transport ...
Cited by 5 - Related articles - BL Direct - All 2 versions - Import into BibTeX

Method of invoking inline method and java virtual machine using the method
... the Java virtual machine 110 may drive a web browser or execute an Applet, MIDlet, or Xlet. ... current method for later use in the situations where there is the need to stop the execution ... as the invocation of a method, the invocation of the garbage collector 300, and Java exception ...
All 2 versions - Import into BibTeX

BOOKMARKING DIGITAL CONTENT ON BLU-RAY DISCS
... bookmarks allow for additional functionality, such as adjusting marking elements or lighting in a theater room automatically at the start or end of a motion picture, or allowing a user to stop a movie ... 0057) BD-J titles operate by loading Java programs, or "xlets," onto the ...
All 2 versions - Import into BibTeX

book: Demystifying Embedded Systems Middleware
T Nguyen do - 2010 - books.google.com
... e, Common Object Request Broker Object (CORBA) • Data Access Object (DAO) Frameworks • Authentication and Security, ie. • Java Authentication and ... In fact, the most common mistakes that kill complex embedded systems projects, especially those that utilize middleware ...
Cited by 1 - Related articles - Library Search - Import into BibTeX

Recording medium, playback apparatus, program, and playback method
W Ikeda, H Hino, ... - US Patent 7,715,606, 2010 - Google Patents
... 34 M Module Manager Title Jump! API Call V or Title End Java Module xlet Applica ...
... tion / Manager "BF Branch Destination BD-J Object Notification *38 {1r3 JVM N'2 BD MW Local | memory | > 1 JAR xlet LOAD AND RUN CONTROL L1."
Related articles - All 4 versions - Import into BibTeX

Variable and state handling in NCL
LEG Greaves, RP Rodrigues, R Cesare, ... - Multimedia Tools and ... 2010 - Springer
... These are common situations found in a DTV environment where, in addition, it is very usual to stop a presentation and then to resume it at a future moment in time, preserving all actions previously ... 7 Lua is the Ginga scripting language, while Xlet is the Java code that runs in ...
Cited by 6 - Related articles - All 8 versions - Import into BibTeX

TVGrid: A Grid Architecture to use the Idle Resources on a Digital TV Network
OEO Bastida, TAM de Andujo, ... - Computing and the ... 2007 - www.epj-conferences.org
... ECMAScript. The Java applications executed on the receivers are called Xlets ...
... The Java application control code values for Ginga are "autostart", "present", "destroy", "prefetch", "kill", "remote" and "unbound" ...
Cited by 9 - Related articles - All 6 versions - Import into BibTeX

Playback apparatus, program and playback method

Java 2 Micro Edition
J Card - Mobile phone programming and its application to ..., 2007 - books.google.com
... J2ME has to support a variety of mobile devices with different networking Xlet comes from
JavaTV to support applications running on digital TV receivers... setVisible(true); } });
private java.awt. You can terminate the emulator from Netbeans by clicking Build->Stop Build/Run ...
Related articles - All 3 versions - Import into BbTeX

RECORDING MEDIUM, PLAYBACK APPARATUS, PROGRAM, AND PLAYBACK METHOD
... 20 Module Manager 34 J- T -r Title Jump API Call or Title End Java Module xlet Application -tion
"1 Manager Branch Destination BD-J Object Notification 36 th3 JVM N2 BD M/W L1" Local ...
memory " # JAR 29 xlet LOAD AND RUN CONTROL Page 22 ...
All 2 versions - Import into BbTeX

EFFECTS FOR INTERACTIVE GRAPHIC DATA IN DISC AUTHORIZING
... A BD-J application is a Java Xlet which is controlled by the BD-ROM player's Application Manager ...
The Xlet interface has four states as follows: loaded, paused, active and destroyed... is a second
layer 710 that includes two buttons, a "Continue" button 712 and a "Stop" button 714 ...
All 2 versions - Import into BbTeX

PDF: UIER IMAG
The paper is presented by Kev ... - www.uie.tele.imag.fr
... OSGi Services Platform, component isolation, dynamic services, profiling, memory leaks,
component testing, fast-stop services, dynamic proxies Page 6. Résumé La plateforme des services
OSGi™ est de plus en plus le standard de fait pour des applications Java modulaires ...
Related articles - View as HTML - All 5 versions - Import into BbTeX

Development of a GINGA-NCL receiver for Brazilian mobile broadcasting services
C Lim, ..., - Consumer Electronics, IEEE Transactions on, 2011 - ieeeexplore.ieee.org
... Procedural applications are written using the Java language and are known as GINGA-J.
Declarative applications ... media> element whose content is a NCL document, a Lua script, an
Xlet procedural code ... The NCL provides a set of functions to start, stop, pause or resume in the ...
Related articles - Import into BbTeX

Study and Implementation of Interactive Digital TV on DVB-Multimedia Home Platform
W Li, ..., - 2007 - etd.lib.ncepu.edu.tw
... Page 8. x 4.1 Scenario ... 47 4.2 Uses of Xlets on OpenMHP and XletView Emulator ..... 49 4.3 EPG / VOD / Menu using XML & JAVE technology ..... 53 ...
Related articles - Import into BbTeX

Image forming apparatus, information processing apparatus, information processing method,
information processing program and storage medium
... 104 FIG. 10 148 JSDK Platform: 403 OSGi Framework Bundle 404 XletBundle Activator startO
JL 406 Xlet initXletO startXletO ... S51 'S54 InstallManager APPLICATION SEARCH APPLICATION
SEARCH RESULT S33 START REQUEST -S52 FIG.16 STOP REQUEST 'S51 ...
All 2 versions - Import into BbTeX

PDF: OCAP 1.0 Specific Requirements
A Server ..., - Document Status Sheet - cablelabs.com
...
... An OCAP 1.0 Java application implements the Java TV Xlet interface which provides methods which are called by an application manager. These methods are called to start and stop the application and invoke other state changes defined by a simple finite state machine. 7.2 ...

Semi-automated mobile television interactive application generation based on XHTML and Java ME

Rui Liu - 2011 - rui@brunel.ac.uk

... Extensible Hypertext Mark-up Language (XHTML), Cascading Style Sheets (CSS), Document Object Model (DOM), and European Computer Manufacturers Association Script Language (ECMAScript) ... Defines the application model based on DVB Java (DVB-J) Xlet as well as ...

Related articles - All 7 versions - Import into BibTeX

PDF JAVA APPLICATIONS IN MOBILE DEVICES

M. Heiselmüller - w3.rz.uni-duesseldorf.de

... and mimic the functionality one can roll one’s own thread processing code, e.g. by storing a group of threads objects within a collection class and to provide methods to start, stop all objects within the collection. Page 16. Finalization: The implementation of the Java language for ...

Related articles - All 4 versions - Import into BibTeX

Convergence of iTV and home network platforms

D. Trachtenbo, H. Koselli ... Conference, 2004, CCNC ... iseeexplore.ieee.org

... The OSGi framework is executed on a Java Runtime Environment (JRE) ... In this scenario we implemented a management bundle that can install, start, stop and uninstall bundles ... The management bundle is connected with an iTV application (called an xlet), which is installed ...

Cited by: 27 - Related articles - All 2 versions - Import into BibTeX

A pervasive telemedicine system exploiting the DVB-T technology

G. Arciszus, G. Pinti, R. Raffo, S. Girsch ... for Healthcare, 2006, ... iseeexplore.ieee.org

... The communication is managed through the itvt.com Java package and the serial port behaves according with the RS-232 standard. ... The stop command word is distinguished; the Xlet deals with 8 or 16-bit signals (the data are stored in arrays of short).

Cited by: 1 - Related articles - All 2 versions - Import into BibTeX

Method of providing partially isolated execution environment for multiple applications and digital information apparatus using the same


... MAIN CLASS OF APPLICATION SS341 SS342 SS343 SS344 SS345 EXECUTE MAIN METHOD OF MAIN CLASS STOP ALL THREADS ... applications to be executed in such an information apparatus are created based on the Java specification called Xlet and are ...

All 2 versions - Import into BibTeX

Method of providing partially isolated execution environment for multiple applications and digital information apparatus using the same

C. Park, S. Chung, H. Kwon ... EP Patent 1,630,065, 2007 - impressedons.com

... in such an information apparatus are created based on the Java specification called Xlet and are ... When the Java virtual machine 20 is driven, basic global data structures are initialized, and ...

Further, when the stop of the application during the execution is required, a stop method ...

Cited - Import into BibTeX

Getting Started with the

ISE NetBeans - Springer

... Numerous tools are available for building Java ME applications, but the NetBeans ... As a result, I spend considerable time using the IDE in two step-by-step tutorials: one to build your first CDC/Java ME application, and the other to build your first CDC/AGUI application. ... Import into BibTeX
Service corner: A diagnostic tool for locating osgi stale references

... We could verify that by doing the following steps: (1) stop a range of bundles; (2) verify the number of stale references; (3) perform the call to perform GC; (4) verify ... In Java Personal Basis Profile [19], which is largely utilized in Java TV, the Inter-Xlet Communication Model ...

Cited by 9 - Related articles - All 14 versions - Import into BibTeX

Method of access to applications transmitted within data streams of different television channels and device giving access to broadcasted applications

... The signaling information contained in AIT is then used for managing an Xlet application ... org.dvb.application.AppAttributes; java.util Enumeration. ... The application can be launched and stopped with start() and stop() methods of AppProxy class. ...

Cached - Import into BibTeX

Sviluppo di un'applicazione per un palinsesto televisivo in tecnologia Java Tv

G. Fino ...

Import into BibTeX

REPRODUCTION DEVICE, REPRODUCTION METHOD, AND PROGRAM

... The Xlet interface has four states: "loaded", "paused", "active", and "destroyed" ... a function call from the command interpreter which operates in the HDMV mode or the Java platform which ... from DVD players and CD players, such as the playback start, playback stop, pause, release ...

Cached - Import into BibTeX

PRESENCE-AWARENESS FOR WIRELESS DEVICES

... a Java MIDlet is not essential, and could equally be provided as Java applet or xlet, or as ... [0051] In order to indicate presence of the wireless device 10, the Java MIDlet 11 ... 3-5 minutes On failure/exception: Break the loop, set alive = false; } Interrupt and kill message notification ...

All 2 versions - Import into BibTeX

Hardened Vulnerability and Dynamic Monitoring for DVB-Multimedia Home Platform

... Page 27. 16 MHP APIs Resources System software Java Virtual Machine Application Manager Xlet-Applic ation Xlet-Applic ation Xlet-Applic ation Xlet-Applic ation 資 (resources) 資展現出某 體或 體上約能 可以在同一個平台上包含多種的 ...

Related articles - Import into BibTeX

Handling feature availability in a broadcast

... By placing all the applications in one xlet, all the functionalities provided by the xlet are active. ...

[0033] In the case of java class files, which use feature ... Further, user instructions could be start/activate, stop/terminate/deactivate, hide/iconise/ minimise or show/iconise/maximise. ...

All 2 versions - Import into BibTeX

A distribuição de aplicativos em dispositivos móveis: O caso Java-PMI

A BORGES PONTES ... ... de Computação em ... 2010 - iesam-pa.edu.br

[PDF] from iesam-pa.edu.br
APPARATUS AND METHOD FOR PROVIDING INTERACTIVE SERVICE TO DEVICE USING DIFFERENT DIGITAL BROADCAST MIDDLEWARE STANDARDS
The execution information may include at least one of an OCAP bound application, an OCAP Xlet, an application information... [0068] The above-mentioned mapping process may be necessary for realizing functions such as 'play' and 'stop' in a Java Media Framework ...
All 2 versions - Import into BibTeX

PLAYBACK DEVICE, PLAYBACK METHOD, PLAYBACK PROGRAM, AND INTEGRATED CIRCUIT
(54) PLAYBACK DEVICE, PLAYBACK METHOD, PLAYBACK ...
All 2 versions - Import into BibTeX

CITATION Efficient Native Processing Modules for Interactive DTV Middleware Based on the Small Footprint Set-Top Box
S. G. Shin, D. G. en...
Related articles - Import into BibTeX

Accessing Ambient Intelligence through Devices with Low Computational and Communication Power
F. Boreto, A. Bortzho... - Developing Ambient Intelligence, 2006 - Springer
The second is an Xlet, developed in JAVATV [11], designed for a digital television Set Top Box (DVB-T, with MHP capabilities). 3.1 BTicino MyHome System... It exposes a TCP/IP connection to receive commands such as play, next song, stop, etc., and to provide its status...
Cited by 1 · Related articles · All 7 versions - Import into BibTeX

Print Support for MHP
T. Lehmen - 2005 - mhpproject.org
... An analysis of existing Java printing tools revealed that the print support would be best implemented with the Java Print Service (JPS) API which is part of the standard Java platform ...
JPS on kehitettynyt tulostusapun ja osa nkyistä Java- alustaa...
Related articles · All 4 versions - Import into BibTeX

Create email alert

Did you mean to search for: (KILL OR STOP OR TERMINATE) AND JAVA AND XSLT
## WEST Search History for Application 12656872

**Creation Date:** 2012060317:37

### Interference Searches

<table>
<thead>
<tr>
<th>Query</th>
<th>DB</th>
<th>Op.</th>
<th>Plur.</th>
<th>Thes.</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>((Baek near3 Wonjang) or (Kim near3 John)).in.</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>(((Baek near3 Wonjang) or (Kim near3 John)).in.) and java near10 chang $ near10 state $ 1</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>(((Baek near3 Wonjang) or (Kim near3 John)).in.) and Xlet</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>(((Baek near3 Wonjang) or (Kim near3 John)).in.) and kill $ near10 (java or application $ 1)</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>(((Baek near3 Wonjang) or (Kim near3 John)).in.) and (terminate $ or kill $ or stop $)</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>(((Baek near3 Wonjang) or (Kim near3 John)).in. and (terminate $ or kill $ or stop $)) and java</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>((kill $ or terminate $) and (java application $ 1) and (re-chang $ or chang $) and state $ 1).clm.</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>((kill $ or terminate $) and (java) and (re-chang $ or chang $) and state $ 1).clm.</td>
<td>PGPB, USPT, UPAD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
</tbody>
</table>

### Prior Art Searches

<table>
<thead>
<tr>
<th>Query</th>
<th>DB</th>
<th>Op.</th>
<th>Plur.</th>
<th>Thes.</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>(kill$ or terminat$) near10 (java application$1)</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, JPAB, DWPI</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>Prior Art Searches</td>
<td>TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>--------------------</td>
<td>------</td>
<td>-----</td>
<td>----</td>
<td>----------</td>
<td>------------</td>
</tr>
<tr>
<td>((kill$ or terminat$) near10 (java application$1) ) and ((kill$ or terminat$) near10 (java application$1)).ab.</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>((kill$ or terminat$) near10 (java application$1) ) and (re-chang$ or chang$ or rechang$) near10 state$1 near10 (java or application$1)</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>(kill$ or terminat$ or stop$) near10 (java application$1)</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>((kill$ or terminat$ or stop$) near10 (java application$1) ) and ((kill$ or terminat$ or stop$) near10 (java application$1)).ab.</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>((kill$ or terminat$ or stop$) near10 (java application$1) ) and (kill$ or terminat$ or stop$) near10 (java application$1)).ab. ) not ((kill$ or terminat$) near10 (java application$1) and ((kill$ or terminat$) near10 (java application$1)).ab. )</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>((kill$ or terminat$ or stop$) near10 (java application$1) ) ) and re-chang$</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>(kill$ or terminat$ or stop$) near10 (java)</td>
<td>PGPB, USPT, USOC,</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>(((kill$ or terminate$ or stop$) near10 (java) ) and re-change$)</td>
<td>EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>(((kill$ or terminate$ or stop$) near10 (java) ) and ((kill$ or terminate$ or stop$) near10 (java)).ab.)</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>(((kill$ or terminate$ or stop$) near10 (java) ) and ((kill$ or terminate$ or stop$) near10 (java)).ab.) and (state$1 near5 (application$1 or java))</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>(((kill$ or terminate$ or stop$) near10 (java) ) and switch$ near10 java near10 (background or foreground)</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>(((kill$ or terminate$ or stop$) near10 (java) ) and java near10 (background or foreground)</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
<tr>
<td>(((kill$ or terminate$ or stop$) near10 (java) and java near10 (background or foreground) ) and xlet</td>
<td>PGPB, USPT, USOC, EPAB, JPAB, DWPI, TDBD</td>
<td>ADJ</td>
<td>NO</td>
<td>ASSIGNEE</td>
<td>06-03-2012</td>
</tr>
</tbody>
</table>
### Application/Control No.
12656872

### Applicant(s)/Patent Under Reexamination
BAEK ET AL.

### Examiner
VAN NGUYEN

### Art Unit
2199

---

### SEARCHED

<table>
<thead>
<tr>
<th>Class</th>
<th>Subclass</th>
<th>Date</th>
<th>Examiner</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### SEARCH NOTES

<table>
<thead>
<tr>
<th>Search Notes</th>
<th>Date</th>
<th>Examiner</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVENTOR NAME SEARCH IN PALM AND WEST</td>
<td>6/3/2012</td>
<td>VN</td>
</tr>
<tr>
<td>WEST AND GOOGLE SCHOLAR SEARCH</td>
<td>6/3/2012</td>
<td>VN</td>
</tr>
</tbody>
</table>

### INTERFERENCE SEARCH

<table>
<thead>
<tr>
<th>Class</th>
<th>Subclass</th>
<th>Date</th>
<th>Examiner</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CLAIM</td>
<td>DATE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>06/03/2012</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Did you mean: (KILL OR STOP OR TERMINATE) AND "JAVA APPLICATION" AND XLET

Controlling the smart home from TV
MP, Cabrera, RPP, Redondo, AF, Villas... - Consumer..., 2006 - www.iphone.lee.org
... To sum up, both OSGi and MHP specifications define and implement functionality as a Java application with a lifecycle externally controllable. ... Similarly to the activation order, whatever deactivation order received (Xlet, pauseXlet or bundle stop), the other one is internally ... Cited by 42 - Related articles - All 4 versions - Import into BibTeX

[HTML] Ginga-NCL: the declarative environment of the Brazilian digital TV system
LG, Soares, RFF, Rodrigues... - Journal of the Brazilian..., 2007 - SciELO Brasil
... They have been included in MHP using HTML and plug-ins for non-Java application formats. ... audio objects (MP3, WMA, etc.), text objects (TXT, PDF, etc.), execution objects (Xlet, Lua, etc.) ... Events may generate actions on other media objects, like to start or stop their presentations ... Cited by 132 - Related articles - Cached - All 12 versions - Import into BibTeX

[HTML] Ginga-j: The procedural middleware for the Brazilian digital TV system
GL, Souza Filho, ELC, Letie... - Journal of the Brazilian..., 2007 - SciELO Brasil
... Ginga-J is the middleware subsystem in charge of defining all the Java Application Program Interfaces ... Broadcasted applications (Xlets) must use standardized APIs provided by the Ginga-J. ... stream must be decoded, which subtitle shall be used, to initiate and to stop the media ... Cited by 62 - Related articles - Cached - All 19 versions - Import into BibTeX

System and method Xlet and applet manager testing
... relates generally to computer program testing, and more particularly to applet and Xlet manager software. ... That is, the Java application 108 generally does not need other applications to execute prop ... is, the applet manager 110 determines when to load, start, and stop each applet ... Cited by 1 - Related articles - All 4 versions - Import into BibTeX

Method for managing java applications
... f issue request for executing 2nd java application r switch 1st java application to background r obtain and execute 2nd java application r kill 2nd java ... [0028] Preferably, each of the first Java application and the second Java application includes a Java application of Xlet format. ... All 2 versions - Import into BibTeX

Playback of audio-video content and an associated Java application from an optical disc
Java lifecycle model for bdi-based
... which is software only, further includes, in addition to Java VM 120, the Java Application Programming Interface... into paused state 320 by a call to a pause method 265 or kill active Java... in paused state 320, Java object 220-240 must stop providing any service and might release...
Related articles - All 2 versions - Import into BibTeX

Method for obtaining context of corresponding Xlet while playing BD-J file
... Java application format for MHP (Multimedia Home Platform) and related digital TV standards gives us something pretty close to what we need. Like applet, the Xlet interface allows an external source (the application manager in the case of a digital TV receiver) to start and stop...
Related articles - All 5 versions - Import into BibTeX

Software Lifecycle Management in Java Environments
A. Hugganicht - on informatik.uni-rostock.de
... A JAD (Java Application Descriptor) is a separate text file listing important information about a set of classes in a JAR, in this case, MIDlets in a MIDlet suite. Xlets are supported in Personal Basis Profile (PBP) and in its superset, the Personal Profile (PP). Start stop a bundle...
Related articles - View as HTML - All 2 versions - Import into BibTeX

Gingo-|l: An Open Java-Based Application Environment for Interactive Digital Television Services
... is called ApplicationProxy, which offers the control of the applications lifecycle (start, stop, pause, resume and... The Figure 7(a) displays an Java (Xlet) application using the access APIs for Service (...
7(c) and the Figure 7(d) illustrate the possibility to execute a Java application from a...
Related articles - All 2 versions - Import into BibTeX

DTV APPLICATION SOFTWARE ENVIRONMENT LEVEL 1 (OASE-1): PART 3: PROCEDURAL APPLICATIONS AND ENVIRONMENT
A. Standard - 2003 - assr.org
... 11 4.1.2.3 availability status.... 12 4.2 Xlet Lifecycle Management ...... ...
18 5.1.1.2 Java Application Programming Interfaces...... ...
Related articles - All 4 versions - Import into BibTeX

MHP OSGi convergence: a new model for open residential gateways
AF. Yass, RP Diaz Perdomo., - Software, Practice, ..., 2006 - Wiley Online Library
... Lifecycle management in MHP and OSGi One of the main similarities between OSGi and MHP is that functionality can be implemented as a Java application with limited external management of the lifecycle. Xlet destroy/Xlet Xlet start/Xlet Xlet pause/Xlet installed ... Bundle stop ...
Cited by 6 - Related articles - Bl Direct - All 2 versions - Import into BibTeX

Exploring OSGi capabilities from MHP applications
RPD Roscetti, AF. Yass, NR Gazzara,... - jvrb.org
... To sum up, both OSGi and MHP specifications define an implemented functionality as a Java application with a restricted lifecycle externally manageable. ... Similarly to the activation order, whatever deactivation order received (Xlet.pause/Xlet or bundle:stop), the other one is...
Related articles - Cached - All 2 versions - Import into BibTeX

Embedded Java security: Security for mobile devices
At Debbs - 2007 - books.google.com
... The CDC configuration supports managed applications models namely, applets and Xlets. The Xlet application model is very sim-ilar to the applet application model. Xlets are loaded into an Xlet manager and controlled through a life cycle interface ... Cited by 11 - Related articles - Library Search - All 6 versions - Import into BibTeX

**DIGITAL BROADCASTING APPARATUS AND METHOD FOR PROCESSING KEY SIGNAL**


... The function module 150 comprises the native application regardless of the Java application. ... the main screen A and sub-screen B. [0048] In a situation when displaying the Xlets being received ... key signal is a signal output from a blue button requesting a temporary stop of the ... All 2 versions - Import into BibTeX

**Beginning Java ME Platform**

R Rischpater - 2008 - books.google.com

... the Game Canvas...210 Wrapping Up...218 Intermezzo PART 3 **CDC Development** CHAPTER 9 Introducing Xlets and the Personal Basic Profile ...223 Understanding the Xlet...223 Looking at the Xlet ... Cited by 9 - Related articles - Library Search - All 10 versions - Import into BibTeX

**Image forming apparatus, information processing apparatus, information processing method, information processing program and storage medium**


... AF FIG 10 148 JSDK Platform 403 CGI Framework Bundle 404 XletBundleActivator startO stopO JL 406 Xlet initXletO startXletO ... 351 1545 InstallManager APPLICATION SEARCH APPLICATION SEARCH RESULT 533 START REQUEST-552 FIG.16 STOP REQUEST "SB1 ... All 2 versions - Import into BibTeX

**Towards dynamic component isolation in a service oriented platform**

Y Seto, ... - Component-Based Software Engineering, 2003 - Springer

... The JXC would continue working as the communication mechanism between Xlets, but in a more robust environment with fault isolation ... A research [2] performed on alternatives for Java application isolation and resource accounting mentions component isolation as a means of ... Cited by 8 - Related articles - All 7 versions - Import into BibTeX

**DOWNLOADING OF PROGRAMS INTO BROADCAST-RECEIVERS**

Y Wei - EP Patent 1,552,290, 2008 - freepatentsonline.com

... of service components are video elementary stream, an audio elementary stream, a Java application (Xlet), or other ... Xlet by pressing the radio button in row 510 and column 590, the Xlet will be ... KILL! The destroy method will be called by the application manager when the control ... Cashed - Import into BibTeX

**METHOD AND APPARATUS FOR AUTOMATICALLY SELECTING A SOFTWARE APPLICATION**

J Hirschmup, D Adulj, ... - EP Patent ... 2005 - freepatentsonline.com

... this basic model, the mentioned profiles define an enhanced application model, called the "Xlet" application model. ... E.g the AM can load an application into a JVM, start and stop it ... Thus, when a Java application finishes, the player may apply the exit status information of a finishing ... Related articles - Cashed - Import into BibTeX

**Downloading of programs into broadcast-receivers**


... Examples of service components are video elementary stream, an audio elementary stream, a Java application (Xlet), or other data type. ... a down- load for the seeker training Xlet by pressing the radio button in row 510 and column 590, the Xlet will be ... [0038] KILL! The destroy ... All 2 versions - Import into BibTeX

**System and Method for Reducing the Start-up Time of M20 Applications**


http://scholar.google.com/scholar?hl=en&q=%22STOP+OR+TERMINAT%22+AND+%22JAVA+APPLICATION%22+AND+XLET&btnG=&as_sdt=1%3C247&as_sdtp=18/3/2012 7:51:07 PM
... The pause/Resume signals the Xlet to stop providing service and enter the paused state. A Java application using these generic APIs is called a DVB-J application. Therefore, for a DVB-J application running under MHP 1.0.x, "javax.tv.xlet.Xlet" is the defined interface and is the only ...

All 2 versions - Import into BibTeX

Developing applications for EJB and tru2way™

RM Arela, RD Gigantesio, D Udo, ... - (BMGS8), 2010 IEEE ... 2010 - ieee.org
...
... commercial STB Before testing on the STB we run the application as a Java application which allows ... from the point of view of the STB, the scheduler is the only XLET it ever ... simulator can simulate various platforms and can schedule the application to start and stop execution at ...

Import into BibTeX

Object oriented communication between isolates

...
... The JAVA™ Application Isolate API is defined in JAVA™ Specification Request 121 USR-121 ...
... an application or component, and provides the means to start and stop an isolated ... The application manager can kill applications in response to a user's request, for example, or when ...

Related articles - Cached - Import into BibTeX

Dvd virtual machine

...
... components are a video elementary stream, an audio elementary stream, a Java application (Xlet), or other data ... The soft-ware layers, including the Xlets, may be stored in a repro-grammable ...
... Time)—Play title at a specific time [0049] PTT_Search (PTT number)—stop current pre ...

All 2 versions - Import into BibTeX

[PDF] from cablelabs.com

P2P 7.2 OCAP 1.0 Specific Requirements

A Suyama, ... - Document Status Sheet - cablelabs.com
...
... An OCAP 1.0 Java application implements the Java TV Xlet interface which provides methods which are called by an application manager. These methods are called to start and stop the application and induce other state changes defined by a simple finite state machine. 7.2 ...
... View as HTML - All 7 versions - Import into BibTeX

Parsing from RSS feeds to MHP-Television systems

J Compernault, B Casas - 2006 - upcommons.upc.edu
...
... eletion ... 28 2.4 MHP ... 29 2.4.1 MHP architecture ... 30 ...
2.4.2 Life cycle of Xlet (application manager) ... 31 ...

Cited by 1 - Related articles - All 14 versions - Import into BibTeX

[PDF] from up.edu

Object oriented communication between isolates

Jz Prestic, ... - US Patent 7,765,585, 2010 - google.com
...
... The JAVA™ Application Isolate API is defined in JAVA™ Specification Request 121 (JSR-121) which is ... application or component, and provides the means to start and stop an isolated ... 3 shows...
an exemplary configuration for inter Xlet communication (XJC) between isolates in ...

Related articles - All 5 versions - Import into BibTeX

Method for managing applications related to data broadcasting, class/interface structure for controlling the same, and broadcast receiver for controlling the class/....

... 65 The Java application 'Xlet' is downloaded, is loaded in a memory, and is initialized. In this case, in order to command another Xlet to pause or destroy a specific Xlet, the another Xlet must call a pause or stop function using "org.ch.b.application.AppProxy object". ....
Cited by 1 - Related articles - All 5 versions - Import into BibTeX

Java development on PDAs: building applications for PocketPC and Palm devices
D Wilding-McBride - 2003 - books.google.com

Cited by 15 - Related articles - Library Search - All 5 versions - Import into BibTeX

Method for managing applications related to data broadcasting, class/interface structure for controlling the same, and broadcast receiver for controlling the class/....

... The Java application 'Xlet' is downloaded, is loaded in a memory, and is initialized. In this case, in order to command another Xlet to pause or destroy a specific Xlet, the another Xlet must call a pause or stop function using "org.ch.b.application.AppProxy object". ....
Cited by 1 - Related articles - View as HTML - Library Search - All 8 versions - Import into BibTeX

Digital television applications
C Peng - 2002 - lib.hut.fi

... control keys and to adapt the graphical device environment. The architecture of both application manager and Xlet forms the basic framework for running multiple interactive services simultaneously in future set-top box designs. ....
Cited by 46 - Related articles - View as HTML - Library Search - All 8 versions - Import into BibTeX

Development environment for development of graphical user interface of DTV
T Kilander - 2009 - svl.inf.br

... Tuner control API + DAVIC Page 26. 14 2 2.3 DVB-J application lifecycle Xlets have a specific lifecycle compared to classical Java application. Instead, they have a lifecycle closer to that of a Java applet. They are controlled by the broadcaster or user via the middleware. ....
Cited by 1 - Related articles - View as HTML - All 5 versions - Import into BibTeX

APPARATUS AND METHOD FOR PROVIDING INTERACTIVE SERVICE TO DEVICE USING DIFFERENT DIGITAL BROADCAST MIDDLEWARE STANDARDS

... information may include at least one of an OCAP unbound application, an OCAP Xlet, an appli ....
A Java application, and provides the application information regarding a UI or a Java application to the ... process may be nec- essary for realizing functions such as play and 'stop' in it ...
All 2 versions - Import into BibTeX

Programming Java 2 Micro Edition on Symbian OS

Cited by 54 - Related articles - View as HTML - Library Search - All 104 versions - Import into BibTeX

Accessing Ambient Intelligence through Devices with Low Computational and Communication

PDF from polito.it
Power
F Boreto, D Buziha. - Developing Ambient Intelligence, 2006 - Springer
... It exposes a TCP/IP connection to receive commands such as play, next song, stop, etc., and to provide its ... Intrusion alarms are settable The same abstractLayout.xml file, has been used to configure two clients: a Java application running on a laptop and an Xlet running on a ...
Cited by 1 - Related articles - All 7 versions - Import into BibTeX

JAVA CONDITIONAL ACCESS APPARATUS
(54) JAVA CONDITIONAL ACCESS APPARATUS Publication ...
All 2 versions - Import into BibTeX

HTML Interactive Digital Terrestrial Television: The Interoperability Challenge in Brazil G Barde, G Bennett, GL Oiarbide... - International Journal of... 2009 - hindawi.com
... are based on the common GEM core, it is possible to write Java-application with interoperable ... image objects, video objects, audio objects, text objects, execution objects (eg, Xlet, Lua, etc ... generate actions on other media objects, like to start, pause, or stop their presentations. ...
Cited by 5 - Related articles - Cached - All 15 versions - Import into BibTeX

Method for wireless communication and apparatus for conducting the same
... Sender of Message Delivery and Recipient Decodes and Renders Message 511a 511b Stop Fig. S... Device Profile (MIDP) 209 including a set of higher-level Java Application Programming Interfaces ... An XLET 221, as a mobile application, can be programmed using MMAPI 211b ...
Cited by 10 - Related articles - All 2 versions - Import into BibTeX

Playback apparatus for performing application-synchronized playback
... The above elements 17 to 20 enable contents downloaded by a Java application via a network ... program constituting an application that is made executable by placing the xlet program in ... of functions inherited from DVD players and CD players, including PLAY, STOP, PAUSE ON ...
Cached - Import into BibTeX

Playback apparatus for performing application-synchronized playback
... The above elements 17 to 20 enable contents downloaded by a Java application via a network ... program constituting an application that is made executable by placing the xlet program in ... of functions inherited from DVD players and CD players, including PLAY, STOP, PAUSE ON ...
Cached - Import into BibTeX

Playback apparatus for performing application-synchronized playback
... The above elements 17 to 20 enable contents downloaded by a Java application via a network ... program constituting an application that is made executable by placing the xlet program in ... of functions inherited from DVD players and CD players, including PLAY, STOP, PAUSE ON ...
Cached - Import into BibTeX

Playback apparatus for performing application-synchronized playback
... The above elements 17 to 20 enable contents downloaded by a Java application via a network ... program constituting an application that is made executable by placing the xlet program in ... of functions inherited from DVD players and CD players, including PLAY, STOP, PAUSE ON ...
Cached - Import into BibTeX

Playback apparatus for performing application-synchronized playback

http://scholar.google.com/scholar?q=%28%5B Kill OR STOP OR TERMINATE%5D%29 AND %22JAVA APPLICATION%22 AND %22XLET%22&h=as&sof=1&scfd=1&scdate=83201275107PM
Playback apparatus for performing application-synchronized playback

K Tanaka, H Ohta... - EP Patent 2,178,083, 2010 - freepatentonline.com

... The above elements 17 to 20 enable contents downloaded by a Java application via a network ... program constituting an application that is made executable by placing the xlet program in ... of functions inherited from DVD players and CD players, including PLAY, STOP, PAUSE ON ...

Cached - Import into BibTeX

INFORMATION STORAGE MEDIUM STORING MANAGEMENT INFORMATION OF APPLICATION EXECUTABLE FROM A PLURALITY OF STORAGE MEDIA AND...


... C or JAVA. For example, the present inven- tion will describe a JAVA application such as xlet [0053] The system data 130 includes start-up information, title information, and application management information 131. Start-up ...

All 2 versions - Import into BibTeX

Playback apparatus for performing application-synchronized playback


... java ... aaaa class bbb.class { image } menu.jpg CLASS FILES FOR xlet PROGRAMS FIG.11B CLASS FILE FOR xlet PROGRAM CONSTANT ... Fig.16A FIG.16B AVClip#1 Local Storage Java APPLICATION AVClip#2 AVClip#3 AVClip#4 Title On BD-ROM Java APPLICATION ...

Cited by 1 - Related articles - All 4 versions - Import into BibTeX

Service coroner: A diagnostic tool for locating osgi stale references

T Gatta... - Software Engineering and Advanced ... 2009 - ieeexplore.ieee.org

... Moreover, the OSGI specification applies service- oriented architecture principles to Java application design ... We could verify that by doing the following steps: (1) stop a range of bundles; (2 ... Basic Profile [19], which is largely utilized in Java TV, the Inter-Xlet Communication Model ...

Cited by 9 - Related articles - All 14 versions - Import into BibTeX

[book] Xlet Resource Estimation

C Köberl... - 2004 - Ghosegar

... As already mentioned, DVB-J applications, so called Xlets, are Java programs. Java programs usually have full control of their lifecycle, eg they may exit the JVM. This conventional Java application model does not work well for MHP. If a single Xlet would 12 Page 23 2.3 ... Related articles - View as HTML - All 2 versions - Import into BibTeX

Java 2 Micro Edition

J Card... - Mobile phone programming and its application to ..., 2007 - books.google.com

... 5) provided an upgrade path for applications created for the Personal Java Application Environment Specification ... to support a variety of mobile devices with different networking 5 Xlet comes from ... You can terminate the emulator from Netbeans by clicking Build-> Stop Build Run ...

Related articles - All 5 versions - Import into BibTeX

[PDF] JET IIMAG

R4 presented by Yass... - www.xtele.imag.fr

... tool called Service Coroner which provides runtime detection of stale references and has shown the reality of that problem by diagnosing four large OSGi applications: JOnAS, Sling, SIP Communicator, Newton; the second implementation is a fail-stop services mechanism to ...

Related articles - View as HTML - All 5 versions - Import into BibTeX

PLAYBACK APPARATUS FOR PERFORMING APPLICATION-SYNCHRONIZED PLAYBACK

http://scholar.google.com/scholar?q=%28%5B%5D%29+OR+%28STOP+OR+TERMINAT%29+AND+%22JAVA+APPLICATION%22+AND+%22XLET%20with%22+AND+&as_sdt=1%2C47&as_sdtp=1&start=6&hl=en&as_sdt=1%2C47&as_sdtp=1
PLAYBACK APPARATUS FOR PERFORMING APPLICATION-SYNCHRONIZED PLAYBACK


... TIB CLASS FILE FOR xlet PROGRAM CONSTANT POOL INTERFACE | METHOD| ... as o m $ 00

- a 0 x 0.5 s O = G: O = GC @ T @ O @ O s s = H FIG.16A AVGclip#1 Local Storage Java APPLICATION
AVClip#2 AVGclip#3 AVGclip#4 FIG.16B Title ON BD-ROM Java APPLICATION ... 

All 2 versions - Import into BibTeX

information storage medium storing management information of application executables from a
 plurality of storage media, and method and apparatus for executing the ....


... C or JAVA. For example, the present invention will describe a JAVA application
 such as xlet [0052] The system data 130 includes start-up information, title
 information, and application management information 131. Start-up ...

All 2 versions - Import into BibTeX

CITATION Efficient Native Processing Modules for Interactive DTV Middleware Based on the Small
Footprint Set-Top Box

Shi Jin, GJ Arta ... Related articles - Import into BibTeX

STORAGE MEDIUM STORING TEXT-BASED SUBTITLE DATA INCLUDING STYLE
 INFORMATION, AND APPARATUS AND METHOD OF PLAYING BACK THE ...


... 1 import BDROM.media.*; 2 : public class PlayXlet extends Xlet implements ControllerListener { 
3: Player player; 4: public void init() { 5: setLayout(new BorderLayout()); 6: String mediaFile ... 9
illustrates an example of a Java application written using the API defined in FIG. ...

All 2 versions - Import into BibTeX

Storage medium storing application data providing programming function, and apparatus and
method for reproducing the application


... In particular, the present invention will describe a JAVA application such as xlet. ... The suspend
method and time is used to indicate time to stop the reproduction of the application or the
command used to suspend the reproduction of the application or event information. ...

All 2 versions - Import into BibTeX

Web-based television


... Examples of service components are a video elementary stream, an audio elementary stream,
Java application (Xlet), or other data type. ... the title [0053] Time, Play (Title number, Time)—Play
title at a specific time [0054] PTT, Search (PTT number)—stop current presentation ...

All 2 versions - Import into BibTeX

PDF Application and System Migration from OpenTV to DVB-MHP

M Fagerquist, ... - Studies in Higher Education, 2000 - fondazionerosselli.it

... 16 4 2 4.1 Broadcast Applications 17 4 2.4 2 DVB-J Applications
and the Xlet Interface 17 4 2.5 DVB-J API ...

Cited by 2 - Related articles - All 7 versions - Import into BibTeX

PDF Concierge: a service platform for resource-constrained devices

[PDF] from psu.edu
... code, e.g. by storing a group of threads objects within a collection class and to provide methods to start, stop all objects. Handling security All devices that are running a Java application need to be protected from malicious code (intentional or otherwise) that may access system ...

Related articles - View as HTML - All 4 versions - Import into BibTeX

Rendering of text-based subtitle data including user selectable style information

Smarsh ... EP Patent 2,149,159, 2008 - freesoft.org

... "stop();" stops outputting the text subtitle. ... 9 illustrates an example of a Java application written using the API defined in Fig. ... 9, application PlayXlet inherits a function of application Xlet, which is a basic unit of the Java application and included in the full data 110 executed by the ...

Cached - Import into BibTeX

system and method for creating, distributing, and executing rich multimedia applications

S Morris ... 2005 - books.google.com

... The Impact of the Java Community Process Portability Performance Issues 4 Applications and Application Management An Introduction to Xlets Xlet Basics Xlet Contents Writing Your First Xlet Dos and Don'ts for Application Developers Application Signaling Extending the AWT ...

Cited by 395 - Related articles - Library Search - All 7 versions - Import into BibTeX

recording medium, reproduction device, program, reproduction method

T Okada, W Ikegami, Y Iesawa ... EP Patent 1,711,012, 2006 - freesoft.org

... A class file of an xlet program is explained. ... With the Network Device 17 to De-mux 20, the contents that a Java application downloaded via a network can be ... similar to that found in DVD and CD players, such as starting playback (Play); stopping playback (Stop); pausing (Pause) ...

Cached - Import into BibTeX

producer, integrated circuit, reproduction method, application program, recording medium, recorder, and recording

Y Matsukawa, Y Kiyosawa ... US Patent ... 2012 - freesoft.org

... of the local storage during execution of the application program and before the stop of the ... The BD-J application is a Java application operated in a Java platform fully provided with Java™ 2. The BD-J application is controlled by Application Manager through a xlet interface ...

Cached - Import into BibTeX

recording medium, reproduction device, program, reproduction method


... class bbb class ( image ) menu. jpg 5-DIGIT INTEGER, ZZZZ, BECOMES Application ID CLASS FILES FOR xlet PROGRAM FIG.12B CLASS FILES FOR xlet PROGRAM CONSTANT POOL INTERFACE METHOD 1 METHOD 2 METHOD 3 < ©Event Listener METHOD ...

All 2 versions - Import into BibTeX

recording medium, reproduction device, program, reproduction method, and system integrated circuit

H Itamoto, W Ikegami, T Okada ... US Patent 7,585,082, 2009 - Google Patents

... 21, 2009 Sheet 12 of 54 RG.12A JAR file (ZZZZJAR) C Root commons.jar Cava ) . aaaa.class bbb.class ( image ) menu. jpg 5-digit integer ZZZZ indicates Application ID Class files for xlet program FIG.12B Class File for xlet Program Constant Pool Interface Method 1 Method ...

Cited by 1 - Related articles - All 4 versions - Import into BibTeX

integrated circuit for use in a playback apparatus
Method for providing record information in a digital broadcast receiver and a digital broadcast receiver for providing record information


go) Pub. No.: US 2009/0254964 AI Park et al. (43) Pub. Date: Oct. 8,2009 (54) METHOD FOR PROVIDING RECORD INFORMATION IN A DIGITAL ...

All 2 versions - Import into BibTeX

Method for a digital broadcast receiver for providing a list of records


All 2 versions - Import into BibTeX

INTEGRATED CIRCUIT OR USE IN PLAYBACK APPARATUS


... Out time Page 12. Patent Application Publication Aug. 13, 2009 Sheet 11 of 33 US 2009/0202228 AI FIG. 11A JAR File Util., JAR ( Root ) common, pic ( Java ) image ) aaa. class bbb class menu, jpg m*5-digit integer = Application ID Class Files for xlet program FIG. ...

Cited by 11 - Related articles - All 2 versions - Import into BibTeX

INTEGRATED CIRCUIT FOR USE IN A PLAYBACK APPARATUS


... Composite Image of Parent Screen and Child Screen Parent: Execution Image of Java Application Child: Playback ... in Full Screen 13 PLAY LIST #1 Operation Status application #1 "Stop" Start-Up ... $S 5-digit integer ZZZZ indicates Application ID Class files for xlet program FIG ...

Related articles - All 2 versions - Import into BibTeX

Recording medium, reproduction device, program, and reproduction method

W. Ikeda, H. Iwamoto, ... - US Patent 7,615,312, 2009 - Google Patents

... 7C JAVA APPLICATION ... $B xlet PROGRAM JMF A "BD:/00001.mpl"; A playO: Jump Title[Title#:1] J PLAYER INSTANCE J CONTROL TITLE BRANCH ... 3B title #2 TIME AXIS: vTerminat
[j [Terminat]^1 & i Kun j] "-> ~ > Kun 1 > application #3 LIFE CYCLE application #1 LIFE ...

Cited by 1 - Related articles - All 4 versions - Import into BibTeX

REPRODUCIBLE PROGRAM AND REPRODUCING METHOD


... 1B Class File for xlet Program Constant Pool Interface Method 1 Method 2 Method 3 v*5-digit integer ZZZZ = Application ID Class File for xlet Program @Event Listener Method @Method of JMF Player Instance @Method Calling Function API Method n Page 13. Network FIG ...

All 2 versions - Import into BibTeX

REPRODUCTION DEVICE, REPRODUCTION METHOD, AND PROGRAM
Recording medium, playback apparatus, program, and playback method


... LIFE CYCLE application ID RUN ATTRIBUTE FIG. 7C JAVA APPLICATION Page 10 US Patent May 11, 2010 Sheet 8 of 60 US 7,715,696 B2 FIG. 8A JAR FILE ( Root ) common.pkg aaa class bbb.class - image.jpg xlet PROGRAM Event listener PROGRAM FIG. ...

Related articles - All 4 versions - Import into BibTeX

Playback apparatus program and playback method


... LIFE CYCLE application ID RUN ATTRIBUTE FIG. 7C JAVA APPLICATION Page 10 US Patent Apr. 20, 2010 Sheet 8 of 60 US 7,702,222 B2 FIG. 8A JAR FILE common.pkg ( java ) aaa class bbb.class - image.jpg xlet PROGRAM Event listener PROGRAM FIG. ...

Related articles - All 4 versions - Import into BibTeX

Recording medium, reproduction device, program, reproduction method, and integrated circuit


... 4/2011 Sheet 13 of 45 US 7,865,069 B2 FIG.8 information correlating Java application with stream—BD—Object XXXXXX.B0B (AMT) Application Management Table Defining Java application -- whose life cycle is Title 1 corresponding to BD—Object PlayList Management ...

Related articles - All 4 versions - Import into BibTeX

READING DEVICE, PROGRAM, AND READING METHOD


... A Java™ application in BD-J mode is controlled by an Application Manager via an xlet interface. The xlet interface has four statuses of "loaded", "paused", "active" and "destroyed". The above-mentioned Java™ platform includes ...

Related articles - Cached - Import into BibTeX

Reading device, recording method and reading method


... A Java™ application in BD-J mode is controlled by an Application Manager via an xlet interface. The xlet interface has four statuses of "loaded", "paused", "active" and "destroyed". The above-mentioned Java™ platform includes ...

Cached - Import into BibTeX

Recording medium, playback apparatus, recording method, and playback method


... FIG. 7C JAVA APPLICATION Page 10 US Patent US 7,620,815 B2 Dec. 8, 2009 Sheet 8 of 60 FIG. 8A JAR FILE ( Root ) common.pkg aaa.class bbb.class - image.jpg xlet PROGRAM Event listener PROGRAM FIG. 8B ...

Cited by 1 - Related articles - All 4 versions - Import into BibTeX

PDF TAPAS for wireless PDA

E. Lühr - Project at department of Teleinformatics, NT6U, Spring, 2003 - tapas@nt6u.njit.edu
Recording medium, playback apparatus, program, and playback method


... 7C JAVA APPLICATION Page 9. Patent Application Publication Oct. 14, 2010 Sheet 8 of 60 US 2010/0260016 Al FIG. 9A JAR FILE ( Root ) common.sh ( Java ) aab.class bbb.class ( image ) menu.jpg xlet PROGRAM Event listener PROGRAM FIG. ...

All 2 versions - Import into BibTeX

Reproduction device, reproduction method, and program


All 2 versions - Import into BibTeX

Recording medium, playback apparatus, program, and playback method


... 7C JAVA APPLICATION Page 9. Patent Application Publication Aug. 12, 2010 Sheet 8 of 60 US 2010/0202278 Al FIG. 8A JAR FILE ( Root ) common.sh aab.class bbb.class ( image ) menu.jpg xlet PROGRAM Event listener PROGRAM FIG. ...

All 2 versions - Import into BibTeX

Recording medium, reproduction device, program, reproduction method, and integrated circuit


... PLMT Sound Management Table (SMT) Defining Java application whose life cycle is Title corresponding to BD-JObject Indicating Playlist to be reproduced simultaneously with execution of Java application in Title ... class Java3*. Class files *-defining xlet programs 00007. ...

All 2 versions - Import into BibTeX

Create email alert

Did you mean to search for: (KILL OR STOP OR TERMINATE) AND "JAVA APPLICATION" AND XLET?
NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 08/18/2011.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

/sbrahim/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101
NOTICE REGARDING CHANGE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 08/18/2011.

• The Power of Attorney to you in this application has been revoked by the assignee who has intervened as provided by 37 CFR 3.71. Future correspondence will be mailed to the new address of record (37 CFR 1.33).

/sibrahim/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101
POWER OF ATTORNEY BY ASSIGNEE OF ENTIRE INTEREST
AND REVOCATION OF PRIOR POWERS

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

The undersigned corporation is the assignee of record of the entire right, title, and
interest of the patent applications and patents identified listed in the enclosed Attachment A:

REVOCATION OF PRIOR POWERS OF ATTORNEY

revokes all powers of attorney previously given, and

NEW POWER OF ATTORNEY

appoints the attorneys and/or agents of STEIN MCEWEN LLP under Customer No.
49,455 to prosecute and transact all business in the United States Patent and Trademark Office
connected therewith.

CORRESPONDENCE CHANGE OF ADDRESS

All correspondence and telephone communications should be directed to the address
associated with Customer Number 49,455, which is currently:

STEIN MCEWEN LLP
1400 EYE ST., N.W.
SUITE 300
WASHINGTON, D.C. 20005
PHONE: (202) 216-9505
FACSIMILE: (202) 216-9510
STATEMENT AND CERTIFICATION UNDER 37 CFR §3.73(B)

ANYPOINT MEDIA GROUP, a United States corporation, certifies that it is the assignee of the entire right, title and interest in the patent applications and patents identified in the enclosed Attachment A, by way of assignments, and those assignments were recorded in the USPTO with available data identified in the Attachment A and/or are attached hereto as indicated in Attachment A.

The undersigned is empowered to sign this certificate on behalf of the assignee.

I hereby declare that all statements made herein of my own knowledge are true, and that all statements made on information and belief are believed to be true; and further, that these statements are made with knowledge that willful false statements, and the like so made, are punishable by fine or imprisonment, or both, under Section 1001, Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

If there are any fees associated with the filing of this Statement and Certification, please charge and/or credit the same to Deposit Account No. 503333.

By: [Signature]

Name: HAN JUN-SIK

Title: Managing Director

Dated 08/16/11
<table>
<thead>
<tr>
<th>Serial No.</th>
<th>Assignment 1 Reel/Frame</th>
<th>Assignment 2 Reel/Frame</th>
<th>Assignment 3 Reel/Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/937,057</td>
<td>025121/0810</td>
<td>026132/0956</td>
<td></td>
</tr>
<tr>
<td>12/937,111</td>
<td>025121/0694</td>
<td>026132/0956</td>
<td></td>
</tr>
<tr>
<td>12/882,665</td>
<td>024998/0384</td>
<td>026132/0956</td>
<td></td>
</tr>
<tr>
<td>12/824,941</td>
<td>024606/0855</td>
<td>026132/0956</td>
<td></td>
</tr>
<tr>
<td>12/819,272</td>
<td>024574/0578</td>
<td>026132/0956</td>
<td></td>
</tr>
<tr>
<td>12/656,872</td>
<td>024026/0158</td>
<td>026132/0956</td>
<td></td>
</tr>
<tr>
<td>12/450,273</td>
<td>023277/0367</td>
<td>026132/0956</td>
<td></td>
</tr>
<tr>
<td>12/450,067</td>
<td>023231/0392</td>
<td>026132/0956</td>
<td></td>
</tr>
<tr>
<td>12/450,066</td>
<td>023231/0687</td>
<td>026132/0956</td>
<td></td>
</tr>
<tr>
<td>12/449,565</td>
<td>023111/0831</td>
<td>026132/0956</td>
<td></td>
</tr>
<tr>
<td>12/449,563</td>
<td>023112/0344</td>
<td>026132/0956</td>
<td></td>
</tr>
<tr>
<td>12/458,877</td>
<td>023053/0751</td>
<td>026132/0956</td>
<td></td>
</tr>
<tr>
<td>12/457,508</td>
<td>022880/0463</td>
<td>026132/0956</td>
<td></td>
</tr>
<tr>
<td>12/457,186</td>
<td>022817/0648</td>
<td>026132/0956</td>
<td></td>
</tr>
<tr>
<td>12/314,069</td>
<td>021969/0898</td>
<td>026132/0956</td>
<td></td>
</tr>
<tr>
<td>12/314,149</td>
<td>021959/0841</td>
<td>026132/0956</td>
<td></td>
</tr>
<tr>
<td>12/285,534</td>
<td>021733/0027</td>
<td>026132/0956</td>
<td></td>
</tr>
<tr>
<td>12/222,337</td>
<td>021414/0287</td>
<td>021850/0408</td>
<td>026132/0956</td>
</tr>
<tr>
<td>12/216,260</td>
<td>021247/0376</td>
<td>021850/0408</td>
<td>026132/0956</td>
</tr>
<tr>
<td>12/216,192</td>
<td>021245/0348</td>
<td>021850/0408</td>
<td>026132/0956</td>
</tr>
<tr>
<td>12/216,826</td>
<td>021279/0573</td>
<td>021850/0408</td>
<td>026132/0956</td>
</tr>
</tbody>
</table>
**Electronic Acknowledgement Receipt**

<table>
<thead>
<tr>
<th>Field</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>EFS ID:</td>
<td>10765973</td>
</tr>
<tr>
<td>Application Number:</td>
<td>12656872</td>
</tr>
<tr>
<td>International Application Number:</td>
<td></td>
</tr>
<tr>
<td>Confirmation Number:</td>
<td>2380</td>
</tr>
<tr>
<td>Title of Invention:</td>
<td>Method for managing java applications</td>
</tr>
<tr>
<td>First Named Inventor/Applicant Name:</td>
<td>Wonjang Baek</td>
</tr>
<tr>
<td>Customer Number:</td>
<td>20529</td>
</tr>
<tr>
<td>Filer:</td>
<td>James Garrett McEwen/Dana Jenkins</td>
</tr>
<tr>
<td>Filer Authorized By:</td>
<td>James Garrett McEwen</td>
</tr>
<tr>
<td>Attorney Docket Number:</td>
<td>30721U</td>
</tr>
<tr>
<td>Receipt Date:</td>
<td>18-AUG-2011</td>
</tr>
<tr>
<td>Filing Date:</td>
<td>18-FEB-2010</td>
</tr>
<tr>
<td>Time Stamp:</td>
<td>15:01:39</td>
</tr>
<tr>
<td>Application Type:</td>
<td>Utility under 35 USC 111(a)</td>
</tr>
</tbody>
</table>

**Payment information:**

- Submitted with Payment: no

**File Listing:**

<table>
<thead>
<tr>
<th>Document Number</th>
<th>Document Description</th>
<th>File Name</th>
<th>File Size(Bytes)/Message Digest</th>
<th>Multi Part/.zip</th>
<th>Pages (if appl.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Change of Address</td>
<td>Req_Change_of_POA-Corr_Address_08-18-11.pdf</td>
<td>84298</td>
<td>no</td>
<td>1</td>
</tr>
</tbody>
</table>

**Warnings:**

- Information:
<table>
<thead>
<tr>
<th>2</th>
<th>Power of Attorney</th>
<th>Executed_POA_ANYPOINT_MENDA_GROUP.pdf</th>
<th>457870</th>
<th>no</th>
<th>3</th>
</tr>
</thead>
</table>

**Warnings:**

**Information:**

| Total Files Size (in bytes) | 542168 |

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

**New Applications Under 35 U.S.C. 111**

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

**National Stage of an International Application under 35 U.S.C. 371**

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

**New International Application Filed with the USPTO as a Receiving Office**

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.
IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of:
Wonjang BAEK et al.

Application No. 12/656,872
Group Art Unit: 2195

Confirmation No. 2380

Filed: February 18, 2010
Examiner: Meng Ai T. An

For: METHOD OF MANAGING JAVA APPLICATIONS

LETTER TO THE EXAMINER REQUESTING ENTRY OF CHANGE IN POWER OF ATTORNEY AND CORRESPONDENCE ADDRESS

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:
In accordance with the enclosed Power of Attorney, it is respectfully submitted that the attorneys of Stein McEwen, LLP, customer number 49,455, are associated with the instant application and all prior powers of attorney have been revoked.

Please direct all correspondence to the address associated with customer number 49,455, which is presently as follows:

Stein McEwen, LLP
1400 Eye St., NW
Suite 300
Washington, D.C. 20005

Respectfully submitted,

STEIN McEWEN, LLP

Michael D. Stein
Registration No. 37,240

Date: August 18, 2011

1400 Eye St. N.W., Suite 300
Washington, D.C. 20005
Telephone: (202) 216-9505
Facsimile: (202) 216-9510
Title: Method for managing java applications

Publication No: US-2010-0218189-A1
Publication Date: 08/26/2010

NOTICE OF PUBLICATION OF APPLICATION

The above-identified application will be electronically published as a patent application publication pursuant to 37 CFR 1.211, et seq. The patent application publication number and publication date are set forth above.

The publication may be accessed through the USPTO’s publicly available Searchable Databases via the Internet at www.uspto.gov. The direct link to access the publication is currently http://www.uspto.gov/patft/.

The publication process established by the Office does not provide for mailing a copy of the publication to applicant. A copy of the publication may be obtained from the Office upon payment of the appropriate fee set forth in 37 CFR 1.19(a)(1). Orders for copies of patent application publications are handled by the USPTO’s Office of Public Records. The Office of Public Records can be reached by telephone at (703) 308-9726 or (800) 972-6382, by facsimile at (703) 305-8759, by mail addressed to the United States Patent and Trademark Office, Office of Public Records, Alexandria, VA 22313-1450 or via the Internet.

In addition, information on the status of the application, including the mailing date of Office actions and the dates of receipt of correspondence filed in the Office, may also be accessed via the Internet through the Patent Electronic Business Center at www.uspto.gov using the public side of the Patent Application Information and Retrieval (PAIR) system. The direct link to access this status information is currently http://pair.uspto.gov/. Prior to publication, such status information is confidential and may only be obtained by applicant using the private side of PAIR.

Further assistance in electronically accessing the publication, or about PAIR, is available by calling the Patent Electronic Business Center at 1-866-217-9197.

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101
This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

Application Number: 10-2009-0014161
Filling Date: FEB. 20, 2009
Applicant(s): Dreamer

2010년 03월 29일

COMMISSIONER
제출 일자 : 2009-02-20

【서지사항】

【서류명】 특허출원서
【참조번호】 0021
【출원구분】 특허출원
【출원인】
【명칭】 드리머
【출원인코드】 5-2009-003922-3
【대리인】
【명칭】 특허법인지명
【대리인코드】 9-2007-100021-5
【지정된변리사】 김경욱, 박준용, 서일형, 이광현, 이창범
【포괄위임등록번호】 2009-008139-0
【발명의 국문명칭】 자바 애플리케이션 관리 방법 및 이를 실행시키기 위한 프로그램을 기록한 컴퓨터로 판독 가능한 기록매체
【발명의 영문명칭】 METHOD OF MANAGING JAVA APPLICATION AND COMPUTER-READABLE MEDIUM HAVING THEREON PROGRAM PERFORMING FUNCTION EMBODYING THE SAME

【발명자】
【성명】 백원장
【성명의 영문표기】 Baek Wonjang
【주민등록번호】 620511-1XXXXXX
【우편번호】 463-731
제출일자 : 2009-02-20

【주소】 경기 성남시 분당구 이매동 아름마을은행아파트 107동 402호

【국적】 KR

【발명자】

【성명】 김동수
【성명의 영문표기】 John Kim

【주민등록번호】 710910-1XXXXXXXX
【우편번호】 152-720

【주소】 서울 구로구 구로1동 650-4번지 SK허브스 8동 1416호
【국적】 KR

【심사청구】 청구

【취지】 위와 같이 특허청장에게 제출합니다.

대리인 [특허법인지명] (서명 또는 인)

【수수료】

【출원료】 0 면 38,000 원
【가산출원료】 52 면 0 원
【우선권주정료】 0 건 0 원
【심사청구료】 27 항 1,210,000 원
【합계】 1,248,000 원
[요약]

본 발명은 사용자 정치에서의 자바 애플리케이션 관리 방법으로서, (a) 제1 자바 애플리케이션으로부터 실행 유형 정보를 포함하는 제2 자바 애플리케이션의 실행 요청을 수신하는 단계와, (b) 상기 실행 유형 정보에 따라서 상기 제1 자바 애플리케이션의 실행 상태를 변경하도록 제어하는 단계와, (c) 상기 제2 자바 애플리케이션을 초기화하고 실행하도록 제어하는 단계와, (d) 상기 제2 자바 애플리케이션으로부터 상기 제2 자바 애플리케이션의 소멸 요청을 수신하는 경우 상기 제2 자바 애플리케이션을 소멸시킴으로 제어하고, 상기 변경된 상기 제1 자바 애플리케이션의 실행 상태를 재변경하여 상기 제1 자바 애플리케이션이 실행되도록 제어하는 단계를 포함하는 자바 애플리케이션 관리 방법에 관한 것이다.

본 발명에 따르면, 서로 다른 자바 애플리케이션들을 서로 연동되는 하나의 자바 애플리케이션으로 간주할 수 있어서 자바 애플리케이션의 확장성을 높일 수 있고 사용자 정지의 시스템 자원을 효율적으로 관리할 수 있다.

【대표도】

도 2

【색인어】

자바 애플리케이션, 사용자 정지, 실행 유형 정보, 일시 정지, 소멸, 백그라운드 실행
제출일자: 2009-02-20

【명세서】

【발명의 명칭】

자바 애플리케이션 관리 방법 및 이를 실행시키기 위한 프로그램을 기록한
 컴퓨터로 판독 가능한 기록 매체(METHOD OF MANAGING JAVA APPLICATION AND
 COMPUTER-READABLE MEDIUM HAVING THEREON PROGRAM PERFORMING FUNCTION EMBODYING
 THE SAME)

【발명의 상세한 설명】

【기술분야】

[0001] 본 발명은 자바 애플리케이션 관리 방법 및 이를 실행시키기 위한 프로그램을
 기록한 컴퓨터로 판독 가능한 기록 매체에 관한 것으로, 더욱 구체적으로는 사용자 장치에서 실행되는 자바 애플리케이션의 상태를 관리하는 자바 애플리케이션
 관리 방법 및 이를 실행시키기 위한 프로그램을 기록한 컴퓨터로 판독 가능한 기록
 매체에 관한 것이다.

【배경기술】

[0002] 종래의 데크비전 또는 DVD 플레이어는 단순히 오디오 및 비디오 데이터를 재생하는 기능만을 구비하였으나, 디지털 방송 및 블루레이 플레이어 등의 보급에 따라
 서 애플리케이션 실행 환경을 제공할 수 있다.

[0003] 또한 종래 이동통신 단말기는 통화 기능만을 구비하였으나, 이동통신 단말기
 운용 체제의 보급에 따라서 애플리케이션 실행 환경을 제공할 수 있다.
컨텐츠 제공자는 전승한 애플리케이션 실행 환경들을 통하여 사용자에게 다양한 애플리케이션 프로그램을 제공할 수 있다.

디지털 범례비전, 블루레이 플레이어, 이동통신 단말기 등의 사용자 장치에 있어서, 애플리케이션 프로그램은 주로 자바를 이용하여 작성되고 바이트 코드의 형태로 콘텐츠 제공자로부터 사용자 장치로 제공된다.

어플리케이션의 시작, 종료, 시스템 자원의 접근과 같은 동작들은 어플리케이션 및 어플리케이션이 실행되는 OS 사이에서 수행된다.

어플리케이션 모델은 어플리케이션이 어떻게 관리되어야 하는지를 정의하며, 어플리케이션이 실행되는 시스템과 어플리케이션 사이의 상호 역할을 정의한다.


특히 디지털 범례비전 등의 임베디드(embedded) 장치에서의 자바의 실행을 위해, 선마이크로시스템 사에서는 Xlet을 제공하고 있다. 즉 Applet이 브라우저 내에서 수행되는 임베디드 애플리케이션인 반면, Xlet은 디지털 범례비전 등과 같은 다른 응용 유형에서 수행되는 임베디드 애플리케이션이다.

이하 본문 발명의 명세서에서 자바 애플리케이션은 자바를 이용하여 작성된 애플리케이션을 지칭하며, 특히 디지털 범례비전, 블루레이 플레이어, 이동통신 단말기 등의 사용자 장치에서 실행되는 애플리케이션을 지칭한다.
디지털 블레비전, 블루레이 플레이어, 이동통신 단말기 등의 사용자 장치는 화면 내에서 일반적으로 하나의 자바 애플리케이션이 실행된다.

이전에 초기 화면에 대한 자바 애플리케이션을 실행하던 도중에 사용자가 다른 자바 애플리케이션의 실행을 선택하면, 초기 화면에 대한 자바 애플리케이션은 소멸되고 사용자가 선택한 자바 애플리케이션이 실행된다.

따라서 초기 화면에 대한 자바 애플리케이션과 사용자가 선택한 자바 애플리케이션은 서로 독립적인 애플리케이션이다. 초기 화면에 대한 자바 애플리케이션과 사용자가 선택한 자바 애플리케이션이 서로 독립적이기 때문에, 사용자가 선택한 자바 애플리케이션에서 정보의 변경이 발생하는 경우, 해당 정보를 초기 화면에 대한 자바 애플리케이션에 적용하기는 어렵다.

【발명의 내용】

【해결하고자 하는 과제】

본 발명의 목적은 서로 다른 자바 애플리케이션들을 서로 연동되는 하나의 자바 애플리케이션으로 간주할 수 있어서 자바 애플리케이션의 확장성을 높일 수 있고 사용자 장치의 시스템 자원을 효율적으로 관리할 수 있는 자바 애플리케이션 관리 방법을 제공하는 데 있다.

본 발명의 다른 목적은 상기 자바 애플리케이션 관리 방법의 각 단계를 실행시키기 위한 프로그램을 기록한 컴퓨터로 만드 가능한 기록 매체를 제공하는데 있다.
【과제 해결 수단】

상기 기술적 과제를 탐색하기 위하여, 본 발명은 사용자 장치에서의 자바 애플리케이션 관리 방법으로서, (a) 제1 자바 애플리케이션으로부터 실행 유형 정보를 포함하는 제2 자바 애플리케이션의 실행 요청을 수신하는 단계와, (b) 상기 실행 유형 정보에 따라서 상기 제1 자바 애플리케이션의 실행 상태를 변경하도록 제어하는 단계와, (c) 상기 제2 자바 애플리케이션을 초기화하고 실행하도록 제어하는 단계와, (d) 상기 제2 자바 애플리케이션으로부터 상기 제2 자바 애플리케이션의 소멸 요청을 수신하는 경우 상기 제2 자바 애플리케이션을 소멸시키도록 제어하고, 상기 변경된 상기 제1 자바 애플리케이션의 실행 상태를 재변경하여 상기 제1 자바 애플리케이션이 실행되도록 제어하는 단계를 포함하는 자바 애플리케이션 관리 방법을 제공한다.

본 발명에 따른 자바 애플리케이션 관리 방법에 있어서, 상기 제1 자바 애플리케이션은 사용자 입력을 수신하는 사용자 인터페이스를 포함하는 것이고, 상기 제1 자바 애플리케이션은 상기 사용자 인터페이스를 통하여 상기 사용자 입력을 수신하면 상기 제2 자바 애플리케이션의 실행 요청을 생성하는 것을 수 있다.

또한 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서, 상기 제2 자바 애플리케이션의 실행 요청은 상기 제2 자바 애플리케이션의 획득 경로를 포함하는 것일 수 있다.

또한 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서, 상기 단계
(c)는, 상기 획득 경로를 이용하여 상기 제2 자바 애플리케이션을 획득하는 단계를 포함하는 것을 수 있다.

또한 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서, 상기 실행 유형 정보는, 상기 제2 자바 애플리케이션의 실행시 상기 제1 자바 애플리케이션은 소멸(destroy)시키고 상기 제2 자바 애플리케이션의 소멸시 상기 제1 자바 애플리케이션을 다시 시작(start)하는 제1 실행 유형 정보와, 상기 제2 자바 애플리케이션의 실행시 상기 제1 자바 애플리케이션의 실행을 임시 정지(pause)시키고 상기 제2 자바 애플리케이션의 소멸시 상기 임시 정지된 제1 자바 애플리케이션의 실행을 속행(unpause)하는 제2 실행 유형 정보와, 상기 제2 자바 애플리케이션의 실행시 상기 제1 자바 애플리케이션을 백그라운드(background)에서 실행시키고 상기 제2 자바 애플리케이션의 소멸시 백그라운드에서 실행되던 상기 제1 자바 애플리케이션을 포어그라운드(foreground)에서 실행하는 제3 실행 유형 정보 중 어느 하나인 것일 수 있다.

또한 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서, 상기 단계 (b)는, 상기 실행 유형 정보가 상기 제1 실행 유형 정보이면 상기 제1 자바 애플리케이션의 실행 상태를 소멸 상태로 변경하도록 제어하고, 상기 실행 유형 정보가 상기 제2 실행 유형 정보이면 상기 제1 자바 애플리케이션의 실행 상태를 임시 정지 상태로 변경하도록 제어하고, 상기 실행 유형 정보가 상기 제3 실행 유형 정보이면 상기 제1 자바 애플리케이션의 실행 상태를 백그라운드 실행 상태로 변경하도록 제어하는 단계를 포함하는 것일 수 있다.
또한 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서, 상기 단계 (d)는, 상기 실행 유형 정보가 상기 제1 실행 유형 정보이면 상기 제1 자바 애플리케이션의 실행 상태를 시각 상태로 변경하도록 제어하고, 상기 실행 유형 정보가 상기 제2 실행 유형 정보이면 상기 제1 자바 애플리케이션의 실행 상태를 습행 상태로 변경하도록 제어하고, 상기 실행 유형 정보가 상기 제3 실행 유형 정보이면 상기 제1 자바 애플리케이션의 실행 상태를 포어그라운드 실행 상태로 변경하도록 제어하는 단계를 포함하는 것일 수 있다.

또한 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서, 상기 실행 유형 정보가 상기 제2 실행 유형 정보 또는 상기 제3 실행 유형 정보로서 지정되지 않은 경우, 상기 실행 유형 정보는 상기 제1 실행 유형 정보인 것일 수 있다.

또한 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서, 상기 제1 자바 애플리케이션 및 상기 제2 자바 애플리케이션은 정보를 공유하는 것이고, 상기 단계 (d)는, (d-1) 상기 제2 자바 애플리케이션에서 설정된 상기 공유된 정보를 상기 제1 자바 애플리케이션에 적용하도록 제어하는 단계를 포함하는 것일 수 있다.

또한 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서, 상기 단계 (d-1)은, 상기 제2 자바 애플리케이션과 상기 제1 자바 애플리케이션 사이의 통신을 통하여 상기 공유된 정보를 적용하도록 제어하는 단계를 포함하는 것일 수 있다.

또한 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서, 상기 단계 (d-1)은, 상기 실행 유형 정보가 상기 제2 실행 유형 정보 또는 상기 제3 실행 유형 정보가 상기 제1 실행 유형 정보인 것일 수 있다.
정보만 경우에만 상기 공유된 정보를 상기 제1 자바 에플리케이션에 적용하도록 제어하는 단계를 포함하는 것일 수 있다.

또한 본 발명에 따른 자바 에플리케이션 관리 방법에 있어서, 상기 제1 자바 에플리케이션 또는 상기 제2 자바 에플리케이션은 Xlet 형태의 자바 에플리케이션의 것일 수 있다.

또한 본 발명은 사용자 장치에서의 자바 에플리케이션 관리 방법으로서, (a) 제1 자바 에플리케이션으로부터 제2 자바 에플리케이션의 실행 요청을 수신하는 단계와, (b) 상기 제1 자바 에플리케이션의 실행 상태를 일시 정지 상태로 변경하도록 제어하는 단계와, (c) 상기 제2 자바 에플리케이션을 초기화하고 실행하도록 제어하는 단계와, (d) 상기 제2 자바 에플리케이션으로부터 상기 제2 자바 에플리케이션의 소멸 요청을 수신하는 경우 상기 제2 자바 에플리케이션을 소멸시키도록 제어하고 상기 일시 정지된 제1 자바 에플리케이션의 실행 상태를 속행 상태로 변경하도록 제어하는 단계를 포함하는 자바 에플리케이션 관리 방법을 제공한다.

본 발명에 따른 자바 에플리케이션 관리 방법에 있어서, 상기 제1 자바 에플리케이션은 사용자 입력을 수신하는 사용자 인터페이스를 포함하는 것이고, 상기 제1 자바 에플리케이션은 상기 사용자 인터페이스를 통하여 상기 사용자 입력을 수신하면 상기 제2 자바 에플리케이션의 실행 요청을 생성하는 것일 수 있다.

또한 본 발명에 따른 자바 에플리케이션 관리 방법에 있어서, 상기 제2 자바 에플리케이션의 실행 요청은 상기 제2 자바 에플리케이션의 획득 경로를 포함하는 것일 수 있다.
또한 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서, 상기 단계 (c)는, 상기 획득 정보를 이용하여 상기 제2 자바 애플리케이션을 획득하는 단계를 포함하는 것일 수 있다.

또한 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서, 상기 제1 자바 애플리케이션 및 상기 제2 자바 애플리케이션은 정보를 공유하는 것이고, 상기 단계 (d)는, (d-1) 상기 제2 자바 애플리케이션에서 설정된 상기 공유된 정보를 상기 제1 자바 애플리케이션에 적용하도록 제어하는 단계를 포함하는 것일 수 있다.

또한 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서, 상기 단계 (d-1)은, 상기 제2 자바 애플리케이션과 상기 제1 자바 애플리케이션 사이에 통신을 통하여 상기 공유된 정보를 적용하도록 제어하는 단계를 포함하는 것일 수 있다.

또한 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서, 상기 제1 자바 애플리케이션 또는 상기 제2 자바 애플리케이션은 XIet 형태의 자바 애플리케이션 인 것일 수 있다.

또한 본 발명은 사용자 장치에서의 자바 애플리케이션 관리 방법으로서, (a) 제1 자바 애플리케이션으로부터 제2 자바 애플리케이션의 실행 요청을 수신하는 단계와, (b) 상기 제1 자바 애플리케이션의 실행 상태를 백그라운드 실행 상태로 변경하도록 제어하는 단계와, (c) 상기 제2 자바 애플리케이션을 초기화하고 실행하도록 제어하는 단계와, (d) 상기 제2 자바 애플리케이션으로부터 상기 제2 자바 애플리케이션의 소멸 요청을 수신하는 경우 상기 제2 자바 애플리케이션을 소멸시키기
도목 제어하고 상기 백그라운드 실행 상태에들 상기 제1 자바 애플리케이션의 실행 상태를 포어그라운드 실행 상태로 변경하도록 제어하는 단계를 포함하는 자바 애플리케이션 관리 방법을 제공한다.

본 발명에 따른 자바 애플리케이션 관리 방법에 있어서, 상기 제1 자바 애플리케이션은 사용자 입력을 수신하는 사용자 인터페이스를 포함하는 것이고, 상기 제1 자바 애플리케이션은 상기 사용자 인터페이스를 통하여 상기 사용자 입력을 수신하면 상기 제2 자바 애플리케이션의 실행 요청을 생성하는 것을 수 있다.

또한 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서, 상기 단계 (c)는, 상기 획득 경로를 이용하여 상기 제2 자바 애플리케이션을 획득하는 단계를 포함하는 것을 수 있다.

또한 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서, 상기 제1 자바 애플리케이션 및 상기 제2 자바 애플리케이션은 정보를 공유하는 것이고, 상기 단계 (d)는, (d-1) 상기 제2 자바 애플리케이션에서 설정된 상기 공유된 정보를 상기 제1 자바 애플리케이션에 적용하도록 제어하는 단계를 포함하는 것을 수 있다.

또한 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서, 상기 단계 (d-1)은, 상기 제2 자바 애플리케이션과 상기 제1 자바 애플리케이션 사이의 통신을
통하여 상기 공유된 정보를 적용하도록 제어하는 단계를 포함하는 것일 수 있다.

또한 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서, 상기 제1 자바 애플리케이션 또는 상기 제2 자바 애플리케이션은 X1et 형태의 자바 애플리케이션 인 것일 수 있다.

또한 본 발명은 상기 자바 애플리케이션 관리 방법의 각 단계를 실행시키기 위한 프로그램을 기록한 컴퓨터로 판독 가능한 기록 매체를 제공한다.

【효과】

본 발명에 따르면 서로 다른 자바 애플리케이션들을 서로 연동되는 하나의 자바 애플리케이션으로 간주할 수 있어서 자바 애플리케이션의 확장성을 높일 수 있고 사용자 정의 시스템 자원을 효율적으로 관리할 수 있다.

【발명의 실시를 위한 구체적인 내용】

이하, 본 발명의 자바 애플리케이션 관리 방법 및 이를 실행시키기 위한 프로그램을 기록한 컴퓨터로 판독 가능한 기록 매체의 실시예를 첨부한 도면을 참조로 보다 구체적으로 설명한다.

도 1은 본 발명에 따른 자바 애플리케이션 관리 방법이 실행되는 실행 환경을 나타내는 도면이다.

도 1을 참조하면, 실행 환경은 자바 애플리케이션 서버(100), 사용자 장치 (200)를 포함할 수 있다.

자바 애플리케이션 서버(100)는 자바 애플리케이션을 저장하며 사용자 장치
(200)의 요청에 따라서 자바 애플리케이션을 사용자 장치(200)에게로 전송하는 장치이다.

사용자 장치(200)는 자바 애플리케이션 서버(100)로부터 수신한 자바 애플리케이션을 실행하여 제공하는 장치이다. 사용자 장치(200)는 에전대 디지털 멀티미디어 장, 셋탑박스, 블루레이 플레이어 또는 이동통신 단말기일 수 있다.

자바 애플리케이션 서버(100)와 사용자 장치(200)는 인터넷 또는 무선망 등의 통신망을 통하여 서로 연결된다.

본원 명세서에서 "서비스 개발자"는 자바 애플리케이션 서버(100) 또는 사용자 장치(200)에서 실행되는 자바 애플리케이션을 개발하는 사람을 지칭하며, "사용자"는 사용자 장치(200)를 사용하여 서비스 개발자에 의해서 개발된 자바 애플리케이션을 기반으로 하는 서비스를 자바 애플리케이션 서버(100)를 통하여 제공받는 사람을 지칭한다.

사용자 장치(200)는 부트 프로그램(210), 컨타임 프로그램(230) 및 자바 애플리케이션(290)을 포함할 수 있다.

부트 프로그램(210)은 사용자 장치(200) 또는 사용자 장치(200)에서 관독 가능한 저장 매체에 배치되며, 사용자 장치(200)의 상태를 확인하고 적절한 컨타임 프로그램(230)을 선택하고 활동한다.

컨타임 프로그램(230)은 자바 애플리케이션(290)을 위한 컨타임 환경을 제공 한다.
런타임 프로그램(230)은 에전다 X21et API(231), 메소드 호출(Method Invocation) API(233), 자원 지원(Resource Support) API(235), 애플리케이션 관리자(Application Manager) API(237), 디버그 지원 툴(239)을 포함할 수 있다.

X21et API(231)는 본 출원인에 의해서 개발된 API로서, Java TV 표준의 인체스드(enhanced) X1et이다. X21et API(231)는 자바 애플리케이션 서버(100)로부터 자바 애플리케이션(290)을 다운로드하여 실행하기 위한 기능을 제공한다.

메소드 호출 API(233)는 본 출원인에 의해서 개발된 API로서, 사용자 장치(200) 측에서 자바 애플리케이션 서버(100) 측의 메소드를 호출하기 위한 API이다.

자원 지원 API(235)는 본 출원인에 의해서 개발된 API로서, 사용자 장치(200)에서의 메모리-효율적(memory-effective) 캐싱(cache)과 리소스 다운로딩을 지원한다. 자원 지원 API(235)는 이미지, 바이트 배열(byte array), 사운드(sound), 자바 클래스 등의 자원을 취급할 수 있다.

애플리케이션 관리자 API(237)는 본 출원인에 의해서 개발된 API로서, 자바 애플리케이션(290)의 라이프사이클(life-cycle)을 관리하는 API이다. 애플리케이션 관리자 API(239)를 통하여 자바 애플리케이션(290)을 시작하거나, 일시 정지하거나, 속행하거나, 백그라운드 또는 포어그라운드에서 실행하거나 또는 소멸시키는 등의 관리를 수행할 수 있다.

디버그 지원 툴(239)은 본 출원인에 의해서 개발된 API로서 디버깅을 지원한다.
자바 애플리케이션(290)은 서비스 개발자에 의해서 개발되고 사용자 장치(200) 내에서 실행되는 자바 애플리케이션이다.

자바 애플리케이션(290)은 루트(root) 자바 애플리케이션(293)과 사용자 자바 애플리케이션(296)으로 구분될 수 있다.

루트 자바 애플리케이션(293)은 사용자 장치(200)에서 실행되는 기본 애플리케이션(base application)이다. 예컨대 루트 자바 애플리케이션(293)은 사용자 장치(200)에서 의무적으로 실행되어야 하는 자바 애플리케이션이다. 루트 자바 애플리케이션(293)은 예컨대 사용자 장치(200)의 실행 초기에 실행되어 사용자 장치(200)의 화면 상에 표시되는 자바 애플리케이션이다. 루트 자바 애플리케이션(293)은 사용자 자바 애플리케이션(296)의 선택 및 실행을 위한 사용자 인터페이스를 포함할 수 있다. 사용자 인터페이스는 예컨대 GUI 형태로 구현되며, 사용자는 사용자 인터페이스를 통하여 사용자 자바 애플리케이션(296)의 선택 및 실행을 지시할 수 있다.

사용자 자바 애플리케이션(296)은 사용자의 선택에 의해서 실행되는 자바 애플리케이션이다. 사용자 자바 애플리케이션(296)은 루트 자바 애플리케이션(293)을 제외한 모든 자바 애플리케이션을 지칭한다.

루트 자바 애플리케이션(293)은 예컨대 사용자 장치(200)의 제조사가 제조하여 사용자 장치(200)에 배치되며, 사용자 자바 애플리케이션(296)은 서비스 개발자가 제조하여 자바 애플리케이션 서비스(100)를 통하여 사용자 장치(200)에서 실행
이하 도 1에서 도시된 실행 관점을 참조로 하여, 본 발명에 따른 자바 애플리케이션 관리 방법에 대해서 보다 상세히 설명한다.

도 2는 본 발명에 따른 자바 애플리케이션 관리 방법의 예시적인 흐름도이다.

우선 제1 자바 애플리케이션으로부터 실행 유형 정보를 포함하는 제2 자바 애플리케이션의 실행 요청을 수신한다(S110).

제1 자바 애플리케이션은 예컨대 전솔한 루트 자바 애플리케이션(도 1의 293)이다. 또는 제1 자바 애플리케이션은 사용자 애플리케이션(도 1의 296)이 수도 있다. 제2 자바 애플리케이션은 예컨대 전솔한 사용자 애플리케이션(도 1의 296)이다. 제1 자바 애플리케이션 또는 제2 자바 애플리케이션은 임베디드 장치의 사용자 장치(200) 내에서의 실행을 위하여 X11t 형태의 자바 애플리케이션인 것이 바람직하다.

도 3은 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서 제1 자바 애플리케이션과 제2 자바 애플리케이션의 관계를 나타내는 도면이다.

루트 자바 애플리케이션은 자시1 자바 애플리케이션의 실행을 요청할 수 있다. 자시1 자바 애플리케이션은 자시2 자바 애플리케이션의 실행을 요청할 수 있다.

루트 자바 애플리케이션과 자시1 자바 애플리케이션 사이에서는, 루트 자바
에플리케이션이 제1 자바 에플리케이션이고 자식1 자바 에플리케이션이 제2 자바 에플리케이션일 수 있다.

그러나 자식1 자바 에플리케이션과 자식2 자바 에플리케이션 사이에서는, 자식1 자바 에플리케이션이 제1 자바 에플리케이션이고 자식2 자바 에플리케이션이 제2 자바 에플리케이션일 수 있다.

한편 제1 자바 에플리케이션은 사용자 입력을 수신하는 사용자 인터페이스를 포함할 수 있다.

사용자 인터페이스는 예컨대 제2 자바 에플리케이션의 실행을 위한 박스 형태의 그래픽 사용자 인터페이스이다. 사용자는 박스 형태의 그래픽 사용자 인터페이스를 이용하여 사용자 입력을 인가한다. 제1 자바 에플리케이션은 사용자 인터페이스를 통하여 사용자 입력을 수신하면 제2 자바 에플리케이션의 실행 요청을 생성 한다.

제2 자바 에플리케이션의 실행 요청은 제1 자바 에플리케이션을 실행하기 전 중에 제2 자바 에플리케이션을 실행하라는 요청이다. 제2 자바 에플리케이션의 실행 요청은 전술하듯이 실행 유형 정보를 포함한다.

실행 유형 정보는 제1 실행 유형 정보, 제2 실행 유형 정보 및 제3 실행 유형 정보 중 어느 하나일 수 있다.

제1 실행 유형 정보는 제2 자바 에플리케이션의 실행시 제1 자바 에플리케이션을 소멸(destroy)시키고, 제2 자바 에플리케이션의 소멸시 제1 자바 에플리케이션을
선을 다시 시작(start)하는 실행 유형이다.

제2 실행 유형 정보는 제2 자바 애플리케이션의 실행시 제1 자바 애플리케이션의 실행을 일시 정지(pause)시키고, 제2 자바 애플리케이션의 소멸시 일시 정지된 제1 자바 애플리케이션의 실행을 속행(unpause)하는 실행 유형이다.

제3 실행 유형 정보는 제2 자바 애플리케이션의 실행시 제1 자바 애플리케이션을 백그라운드(background)에서 실행시키고, 제2 자바 애플리케이션의 소멸시 백그라운드에서 실행되던 제1 자바 애플리케이션을 포어그라운드(foreground)에서 실행하는 실행 유형이다.

따라서 본 발명에 따르면 제2 자바 애플리케이션이 실행되는 경우 실행 유형 정보를 기초로 제1 자바 애플리케이션의 상태를 결정할 수 있다.

한편 실행 유형 정보가 전술한 제2 실행 유형 정보 또는 제3 실행 유형 정보로서 지정되지 않은 경우, 실행 유형 정보를 제1 실행 유형 정보로 간주할 수 있다.

즉 실행 유형 정보가 특정되지 않은 상태에서 제2 애플리케이션의 실행 요청을 수신하면, 실행 유형 정보를 제1 실행 유형 정보로 간주한 후 이후 단계를 수행할 수 있다.

또한 제2 자바 애플리케이션의 실행 요청은 제2 자바 애플리케이션의 확특 정보를 포함할 수 있다.

예컨대 제2 자바 애플리케이션이 사용자 장치(도 1의 200) 내에 미리 저장되
는 경우, 제2 자바 애플리케이션의 획득 경로는 사용자 장치(도 1의 200) 내의 저장 위치를 나타낼 수 있다.

예전에 제2 자바 애플리케이션이 사용자 장치(도 1의 200) 내에 미리 저장되지 않고 자바 애플리케이션 서버(100) 내에 저장되는 경우, 제2 자바 애플리케이션의 획득 경로는 자바 애플리케이션 서버(100) 내의 저장 위치를 나타낼 수 있다.

제2 자바 애플리케이션의 획득 경로는 에컨트 URL(uniform resource locator) 형태로 표시된다.

다시 도 1을 참조하여 설명을 계속한다.

단계 S110을 통하여 제2 자바 애플리케이션의 실행 요청을 수신하면, 실행 유형 정보에 따라서 제1 자바 애플리케이션의 실행 상태를 변경하도록 제어한다 (S130).

전술하듯이 실행 유형 정보는 제1 실행 유형 정보, 제2 실행 유형 정보 및 제3 실행 유형 정보 중 어느 하나일 수 있다.

도 4는 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서 실행 유형 정보에 따라서 제1 자바 애플리케이션의 실행 상태를 변경하도록 제어하는 단계를 보다 상세히 나타내는 흐름도이다.

실행 유형 정보가 제1 실행 유형 정보이며 제1 자바 애플리케이션의 실행 상태를 소멸 상태로 변경하도록 제어한다(S131, S132).

실행 유형 정보가 제2 실행 유형 정보이며 제1 자바 애플리케이션의 실행 상
제출 일자 : 2009-02-20

테를 일시 정지 상태로 변경하도록 제어한다(S134, S135).

실행 유형 정보가 제3 실현 유형 정보이면 제1 자바 애플리케이션의 실행 상태를 백그라운드 실행 상태로 변경하도록 제어한다(S137, S138).

다시 도 1을 참조하여 설명을 계속한다.

단계 S110을 통하여 제2 자바 애플리케이션의 실현 요청을 수신하면, 단계 S130과 병렬적으로 또는 단계 S130을 수행한 이후에, 제2 자바 애플리케이션을 초기화하고 실행하도록 제어한다(S150).

즉 제2 자바 애플리케이션을 실현하여 사용자에게 제공한다.

단계 S150을 통하여 제2 자바 애플리케이션이 실현되는 도중에, 제2 자바 애플리케이션으로부터 제2 자바 애플리케이션의 요청을 수신하는 경우, 제2 자바 애플리케이션을 소멸시킬도록 제어하고, 단계 S130에서 변경된 제1 자바 애플리케이션의 실행 상태를 재변경하여 제1 자바 애플리케이션이 다시 실행되도록 제어한다(S170).

도 5는 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서 실현 유형 정보에 따라서 제1 자바 애플리케이션의 실행 상태를 재변경하도록 제어하는 단계를 보다 상세히 나타내는 흐름도이다.

실행 유형 정보가 제1 실현 유형 정보이면 제1 자바 애플리케이션의 실행 상태를 시작 상태로 재변경하도록 제어한다(S171, S172).

실행 유형 정보가 제2 실현 유형 정보이면 제1 자바 애플리케이션의 실행 상태를 시작 상태로 재변경하도록 제어한다(S173, S174).
태를 수행 상태로 재변경하도록 제어한다(S174, S175).

[0101] 실행 유형 정보가 제3 실행 유형 정보이면 제1 자바 애플리케이션의 실행 상태를 포어그라운드 실행 상태로 재변경하도록 제어한다(S177, S178).

[0102] 한편 제1 자바 애플리케이션 및 제2 자바 애플리케이션은 정보를 공유할 수 있다.

[0103] 단제 S170에서는 제2 자바 애플리케이션에 의해서 설정되거나 변경된 것인 상기 공유된 정보를 제1 자바 애플리케이션에 적용하도록 제어할 수 있다.

[0104] 제2 자바 애플리케이션과 제1 자바 애플리케이션 사이의 통신을 통하여 공유된 정보를 제1 자바 애플리케이션에 적용할 수 있다.

[0105] 특히 실행 유형 정보가 제2 실행 유형 정보 또는 제3 실행 유형 정보인 경우에만, 공유된 정보를 제1 자바 애플리케이션에 적용하도록 제어할 수 있다(S175, S178에서의 선택적 사항임).

[0106] 즉 제1 실행 유형 정보인 경우 제2 자바 애플리케이션이 실행되는 경우, 제1 자바 애플리케이션은 소일 상태이다. 따라서 공유된 정보의 적용을 위하여 제2 자바 애플리케이션과 제1 자바 애플리케이션 사이에 통신이 수행되지 못한다. 따라서 실행 유형 정보가 제2 실행 유형 정보 또는 제3 실행 유형 정보인 경우에도, 공유된 정보를 제1 자바 애플리케이션에 적용하도록 제어할 수 있다.

[0107] 도 6은 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서 애플리케이션 관리자가 제2 실행 정보를 이용하여 자바 애플리케이션의 실행 상태를 변경하는 구

55-22
성의 예를 나타내는 도면이다.

[0108] 애플리케이션 관리자는 간결한 애플리케이션 관리자 API(도 1의 237) 기능을 구현하는 프로그램이다.

[0109] 우선 루트 자바 애플리케이션("루트"로 표시됨)은 애플리케이션 관리자에게 자식1 자바 애플리케이션("자식1"로 표시됨)의 실행 요청을 전송한다.

[0110] 루트 자바 애플리케이션과 자식1 자바 애플리케이션 사이에서는, 루트 자바 애플리케이션이 제1 자바 애플리케이션이고 자식1 자바 애플리케이션이 제2 자바 애플리케이션이다.

[0111] 자식1 자바 애플리케이션의 실행 요청은 startApp(root, Bound, c1URL)로 표시된다.

[0112] startApp(root, Bound, c1URL)는 root, 즉 루트 자바 애플리케이션이 Bound, 즉 제2 실행 유형 정보를 기초로 최적 경로 c1URL에 의해서 획득되는 자바 애플리케이션, 즉 자식1 자바 애플리케이션을 실행하도록 요청하는 기능으로서, 본 출원인에 의해서 정의된 기능이다.

[0113] 애플리케이션 관리자는 startApp(root, Bound, c1URL)를 수신하면, 실행 유형 정보가 제2 실행 유형 정보이므로 pauseXlet() 기능을 이용하여 루트 자바 애플리케이션의 실행 상태를 일시 정지 상태로 변경하도록 제어한다.

[0114] 또한 initXlet()을 이용하여 자식1 자바 애플리케이션을 초기화하고, startXlet()을 이용하여 자식1 자바 애플리케이션을 시작한다.
pauseXlet(), initXlet() 및 startXlet()은 각각 자바 애플리케이션의 실행 상태를 일시 정지 상태로 변경하고, 자바 애플리케이션을 초기화하고, 자바 애플리케이션을 시작하는 기능으로서, 본 출원안에 의해서 정의된 기능이다.

이후 자식1 자바 애플리케이션은 실행 도중에 애플리케이션 관리자에게 자식 2 자바 애플리케이션("자식2"로 표시됨)의 실행 요청을 전송한다.

자식1 자바 애플리케이션과 자식2 자바 애플리케이션 사이에서는, 자식1 자바 애플리케이션 자바 애플리케이션이 제1 자바 애플리케이션이고 자식2 자바 애플리케이션이 제2 자바 애플리케이션이다.

자식2 자바 애플리케이션의 실행 요청은 startApp(child1, Bound, c2URL)로 표시된다.

startApp(child1, Bound, c2URL)는 child1, 즉 자식1 자바 애플리케이션이 Bound, 즉 제2 실행 유형 정보를 기초로 획득 경로 c2URL에 의해서 획득되는 자바 애플리케이션, 즉 자식2 자바 애플리케이션을 실행하도록 요청하는 기능으로서, 본 출원안에 의해서 정의된 기능이다.

애플리케이션 관리자는 startApp(child1, Bound, c2URL)을 수신하면, 실행 유형 정보가 제2 실행 유형 정보이므로 pauseXlet() 기능을 이용하여 자식1 자바 애플리케이션의 실행 상태를 일시 정지 상태로 변경하도록 제어한다.

또한 initXlet()을 이용하여 자식2 자바 애플리케이션을 초기화하고, startXlet()을 이용하여 자식2 자바 애플리케이션을 시작한다.
한편 애플리케이션 관리자는 자식2 자바 애플리케이션의 실행 도중에 자식2 자바 애플리케이션으로부터 자식2 자바 애플리케이션의 소멸 요청을 수신할 수 있다.

`destroyApp(child2)`는 child2, 즉 자식2 자바 애플리케이션의 소멸을 요청하는 기능으로서, 본 출원인에 의해서 정의된 기능이다.

애플리케이션 관리자는 `destroyApp(child2)`를 수신하면, `destroyXlet()`을 이용하여 자식2 자바 애플리케이션을 소멸시키도록 제어한다.

`destroyXlet()`은 자바 애플리케이션을 소멸시키도록 제어하는 기능으로서, 본 출원인에 의해서 정의된 기능이다.

애플리케이션 관리자는 일시 정지 상태로 변경되었던 자식1 자바 애플리케이션의 실행 상태를 다시 속행 상태로 변경한다. 이를 위하여 애플리케이션 관리자는 `startXlet()`을 이용하여 자식1 자바 애플리케이션을 시작한다.

한편 애플리케이션 관리자는 자식1 자바 애플리케이션의 실행 도중에 자식1 자바 애플리케이션으로부터 자식1 자바 애플리케이션의 소멸 요청을 수신할 수 있다.

`destroyApp(child1)`는 child1, 즉 자식1 자바 애플리케이션의 소멸을 요청하는 기능으로서, 본 출원인에 의해서 정의된 기능이다.

애플리케이션 관리자는 `destroyApp(child1)`를 수신하면, `destroyXlet()`을 이용하여 자식1 자바 애플리케이션을 소멸시키도록 제어한다.
에플리케이션 관리자는 일시 정지 상태로 변경되었던 루트 자바 에플리케이션의 실행 상태를 다시 실행 상태로 변경한다. 이를 위하여 에플리케이션 관리자는 `startXlet()`을 이용하여 루트 자바 에플리케이션을 시작한다.

도 6은 참조로 하면 사용자 창에서 1 자바 에플리케이션과 2 자바 에플리케이션 관계가 다수 설정될 수 있다. 즉 각각 1 자바 에플리케이션도 각각 2 자바 에플리케이션을 실행하도록 요청할 수 있다.

도 7은 본 발명에 따른 자바 에플리케이션 관리 방법에 있어서 에플리케이션 관리자가 제1 실행 정보를 이용하여 자바 에플리케이션의 실행 상태를 변경하는 구상의 예를 나타내는 도면이다.

에플리케이션 관리자는 전술한 에플리케이션 관리자 API(도 1의 237) 기능을 구현하는 프로그램이다.

우선 루트 자바 에플리케이션(“루트”로 표시됨)은 에플리케이션 관리자에게 A 자바 에플리케이션(“A”로 표시됨)의 실행 요청을 전송한다.

A 자바 에플리케이션의 실행 요청은 `startApp(root, Standard, aURL)`로 표시된다.

`startApp(root, Standard, aURL)`는 root, 즉 루트 자바 에플리케이션의 Standard, 즉 제1 실행 유형 정보를 기초로 확득 정보 aURL에 의해서 확득되는 자바 에플리케이션, 즉 A 자바 에플리케이션을 실행하도록 요청하는 기능으로서, 본 출원안에 의해서 정의된 기능이다.
애플리케이션 관리자는 startApp(root, Standard, aURL)을 수신하면, 실행 유형 정보가 제1 실행 유형 정보이므로 destroyXlet() 기능을 이용하여 루트 자바 애플리케이션의 실행 상태를 소멸 상태로 변경하도록 제어한다.

또한 initXlet()을 이용하여 A 자바 애플리케이션을 초기화하고, startXlet()을 이용하여 A 자바 애플리케이션을 시작한다.

한편 애플리케이션 관리자는 A 자바 애플리케이션의 실행 도중에 A 자바 애플리케이션으로부터 A 자바 애플리케이션의 소멸 요청을 수신할 수 있다.

destroyApp(A)는 A, 즉 A 자바 애플리케이션의 소멸을 요청하는 기능으로서, 본 출원인에 의해서 정의된 기능이다.

애플리케이션 관리자는 destroyApp(A)를 수신하면, destroyXlet()을 이용하여 A 자바 애플리케이션을 소멸시키도록 제어한다.

애플리케이션 관리자는 소멸 상태로 변경되었던 루트 자바 애플리케이션의 실행 상태를 시작 상태로 변경한다. 이를 위하여 애플리케이션 관리자는 startXlet()을 이용하여 루트 자바 애플리케이션을 시작한다.

한편 루트 자바 애플리케이션("루트"로 표시됨)은 애플리케이션 관리자에게 B 자바 애플리케이션("B"로 표시됨)의 실행 요청을 전송한다.

B 자바 애플리케이션의 실행 요청은 startApp(root, Standard, bURL)로 표시된다.

startApp(root, Standard, bURL)는 root, 즉 루트 자바 애플리케이션이
Standard, 즉 제1 실행 유형 정보를 기초로 획득 경로 bURL에 의해서 획득되는 자바 애플리케이션, 즉 B 자바 애플리케이션을 실행하도록 요청하는 기능으로서, 본 출원인에 의해서 정의된 기능이다.

[0146] 애플리케이션 관리자는 startApp(root, Standard, bURL)을 수신하면, 실행 유형 정보가 제1 실행 유형 정보이므로 destroyXlet() 기능을 이용하여 루트 자바 애플리케이션의 실행 상태를 소멸 상태로 변경하도록 한다.

[0147] 또한 initXlet()을 이용하여 B 자바 애플리케이션을 초기화하고, startXlet()을 이용하여 B 자바 애플리케이션을 시작한다.

[0148] 한편 애플리케이션 관리자는 B 자바 애플리케이션의 실행 도중에 B 자바 애플리케이션으로부터 B 자바 애플리케이션의 소멸 요청을 수신할 수 있다.

[0149] destroyApp(B)는 B, 즉 B 자바 애플리케이션의 소멸을 요청하는 기능으로서, 본 출원인에 의해서 정의된 기능이다.

[0150] 애플리케이션 관리자는 destroyApp(B)를 수신하면, destroyXlet()을 이용하여 B 자바 애플리케이션을 소멸시키도록 한다.

[0151] 애플리케이션 관리자는 소멸 상태로 변경되었던 루트 자바 애플리케이션의 실행 상태를 시작 상태로 변경한다. 이를 위하여 애플리케이션 관리자는 startXlet()을 이용하여 루트 자바 애플리케이션을 시작한다.

[0152] 한편 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서 애플리케이션 관리자가 제3 실행 정보를 이용하여 자바 애플리케이션의 실행 상태를 변경하는 구
성은 전술한 도 6의 구성을 참고할 수 있으므로 생략한다.

다만 도 6에서 `pauseX1et()` 기능 대신에 자바 애플리케이션은 백그라운드 상태에서 실행하도록 제어하는 `background()` 기능을 사용하는 점과 백그라운드 상태에서 실행되도록 변경하는 기능을 사용하는 점이 차이점이다.

도 8은 본 발명에 따른 자바 애플리케이션 관리 방법의 다른 예시적인 흐름도이다.

우선 제1 자바 애플리케이션으로부터 제2 자바 애플리케이션의 실행 요청을 수신한다(S210).

단계 S210을 통하여 제2 자바 애플리케이션의 실행 요청을 수신하면, 제1 자바 애플리케이션의 실행 상태를 일시 정지 상태로 변경하도록 제어한다(S230).

단계 S210을 통하여 제2 자바 애플리케이션의 실행 요청을 수신하면, 단계 S230과 병렬적으로 또는 단계 S230을 수행한 이후에, 제2 자바 애플리케이션을 초기화하고 실행하도록 제어한다(S250).

즉 제2 자바 애플리케이션을 실행하여 사용자에게 제공한다.

단계 S250을 통하여 제2 자바 애플리케이션이 실행되는 도중에, 제2 자바 애플리케이션으로부터 제2 자바 애플리케이션의 소멸 요청을 수신하는 경우, 제2 자바 애플리케이션을 소멸시키도록 제어하고 일시 정지된 제1 자바 애플리케이션의 실행 실패를 속행 상태로 변경하도록 제어한다(S270).
도 8은 참조로 설명되는 실시에는 실행 유형 정보를 별도로 포함하지 않고 자동적으로 제2 자바 애플리케이션의 실행시 제1 자바 애플리케이션의 실행을 일시 정지시키고 제2 자바 애플리케이션의 소멸시 일시 정지된 제1 자바 애플리케이션의 실행을 수행하도록 구성되는 점을 제외하면 전술한 도 2를 참조로 하는 실시예에서와 유사하므로 상세한 설명을 생략한다.

도 9는 본 발명에 따른 자바 애플리케이션 관리 방법의 또 다른 예시적인 효과도이다.

우선 제1 자바 애플리케이션으로부터 제2 자바 애플리케이션의 실행 요청을 수신한다(S310).

단계 S310을 통하여 제2 자바 애플리케이션의 실행 요청을 수신하면, 제1 자바 애플리케이션의 실행 상태를 백그라운드 실행 상태로 변경하도록 제어한다(S330).

단계 S310을 통하여 제2 자바 애플리케이션의 실행 요청을 수신하면, 단계 S330과 병렬적으로 또는 단계 S330을 수행한 이후에, 제2 자바 애플리케이션을 초기화하고 실행하도록 제어한다(S350).

즉 제2 자바 애플리케이션을 실행하여 사용자에게 제공한다.

단계 S350을 통하여 제2 자바 애플리케이션이 실행되는 도중에, 제2 자바 애플리케이션으로부터 제2 자바 애플리케이션의 소멸 요청을 수신하는 경우 제2 자바 애플리케이션을 소멸시키도록 제어하고 백그라운드 실행 상태이던 제1 자바 애플리
케이션의 실행 상태를 포어그라운드 실행 상태로 변경하도록 제어한다(S370).

[0167] 도 9를 참조로 설명되는 실시에는 실행 유형 정보를 별도로 포함하지 않고 자동적으로 제2 자바 애플리케이션의 실행시 제1 자바 애플리케이션을 백그라운드 에서 실행시키고 제2 자바 애플리케이션의 소멸시 백그라운드에서 실행된던 제1 자바 애플리케이션을 포어그라운드에서 실행하도록 구성되는 점을 제외하면 전술한 도 2를 참조로 하는 실시에 또는 도 8을 참조로 하는 실시에서와 유사하므로 상세한 설명을 생략한다.

[0168] 이상에서 설명한 바와 같이 본 발명에 따르면 제1 자바 애플리케이션은 실행 유형 정보를 기초로 다른 제2 자바 애플리케이션의 실행을 요청할 수 있고, 제2 자바 애플리케이션의 실행시 자신은 소멸 상태로 되거나 일시 정지 상태로 되거나 백그라운드 실행 상태로 변경될 수 있다.

[0169] 또한 제2 자바 애플리케이션이 소멸되는 경우에도, 제1 자바 애플리케이션은 제2 자바 애플리케이션에 의해서 설정되는 공유된 정보를 적용할 수 있다.

[0170] 따라서 서로 다른 개발 주체에서 제1 자바 애플리케이션과 제2 자바 애플리케이션이 개발되더라도, 사용자 측면에서는 제1 자바 애플리케이션과 제2 자바 애플리케이션을 서로 연동되는 하나의 자바 애플리케이션으로 간주될 수 있다. 따라서 자바 애플리케이션의 환경성을 높일 수 있다.

[0171] 이렇게 제1 자바 애플리케이션을 제조하는 서비스 제공자는 제2 자바 애플리케이션에 대한 획득 경로를 기초로 링크를 수행하는 것에 의해서 제2 자바 애플리케이션의 실행을 제어할 수 있다.
케이선의 기능을 사용자에게 제공할 수 있다.

또한 루트 자바 애플리케이션 뿐만 아니라 사용자 자바 애플리케이션도 제1 자바 애플리케이션일수 있다. 사용자 자바 애플리케이션에 제1 자바 애플리케이션인 경우, 사용자 자바 애플리케이션은 다른 사용자 자바 애플리케이션을 실행 유형 정보를 기초로 제2 자바 애플리케이션으로서 실행시키며, 자신은 소멸 상태로 되거나 일시 정지 상태로 되거나 백그라운드 실행 상태로 변경될 수 있다.

즉 사용자 자바 애플리케이션도 다른 사용자 자바 애플리케이션에 대한 횡단 경로를 기초로 링크를 수행하는 것에 의해서 다른 사용자 자바 애플리케이션의 기능을 사용자에게 제공할 수 있다.

따라서 예견대 자바 애플리케이션을 이용하여 TV 포털 서비스, 스마트 폴 애플리케이션, 블루레이 기반 애플리케이션을 제공하는 경우 유용하게 적용될 수 있다. 즉 하나의 서비스 개발자가 원하는 모든 기능을 자신이 자바 애플리케이션으로 구현하지 않더라도 다른 서비스 개발자가 구현한 자바 애플리케이션을 전수한 실행 유형 정보를 기초로 실행하는 것에 의해서 다양한 TV 포털 서비스, 스마트 폴 애플리케이션, 블루레이 기반 애플리케이션을 구현할 수 있다.

또한 본 방면에 따르면 사용자 장치의 시스템 자원을 효율적으로 관리할 수 있다. 사용자 장치의 시스템 자원 상태에 따라서 제1 자바 애플리케이션은 제1 실행 유형 정보, 제2 실행 유형 정보 또는 제3 실행 유형 정보를 기초로 제2 자바 애플리케이션을 실행시키며, 자신은 소멸 상태로 되거나 일시 정지 상태로 되거나 백그라운드 실행 상태로 변경될 수 있다.
또한 본 발명은 전술한 본 발명에 따른 자바 애플리케이션 관리 방법의 각 단계를 실현시키기 위한 프로그램을 기록한 컴퓨터로 관독 가능한 기록 매체를 제공한다.

컴퓨터로 관독 가능한 기록 매체는 컴퓨터 시스템에 의하여 읽혀질 수 있도록 데이터, 즉 코드 또는 프로그램 형태의 데이터가 저장되는 모든 종류의 기록 장치를 지칭한다. 이러한 컴퓨터로 관독 가능한 기록 매체는 예컨대 ROM, RAM 등의 메모리와, CD-ROM, DVD-ROM 등의 저장 매체, 자기 테이프, 플로피 디스크 등의 자기 저장 매체, 광 데이터 저장 장치 등이며, 예컨대 인터넷을 통한 전송 형태로 구현되는 경우도 포함한다. 또한 이러한 컴퓨터로 관독 가능한 기록 매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어 분산 방식으로 컴퓨터가 관독 가능한 데이터가 저장되고 실행될 수 있다.

그러나 이러한 컴퓨터로 관독 가능한 기록 매체에 대한 상세한 설명은 도 1 내지 도 9를 참조로 설명한 본 발명에 따른 자바 애플리케이션 관리 방법과 중복되지므로 생략한다.

비록 본 발명의 구성이 구체적으로 설명되었지만 이는 단지 본 발명을 예시적으로 설명한 것에 불과한 것으로, 본 발명이 속하는 기술분야에서 통상의 지식을 가지는 자라면 본 발명의 본질적인 특성에서 벗어나지 않는 범위 내에서 다양한 변형이 가능할 것이다.

따라서 본 명세서에 개시된 실시예들은 본 발명을 한정하기 위한 것이 아니
라 설명하기 위한 것이고, 이러한 실험에 의하여 본 발명의 사상과 범위가 한정되는 것은 아니다. 본 발명의 범위는 아래의 정구범위에 의해 해석되어야 하며, 그와 동등한 범위 내에 있는 모든 기술은 본 발명의 권리범위에 포함되는 것으로 해석되어야 할 것이다.

【산업상이용가능성】

이상 설명한 바와 같이, 본 발명에 따르면 제1 자바 애플리케이션은 실행 유형 정보를 기초로 다른 제2 자바 애플리케이션의 실행을 요청할 수 있고, 제2 자바 애플리케이션의 실행시 자신은 소멸 상태로 되거나 일시 정지 상태로 되거나 백그라운드 실행 상태로 변경될 수 있다.

또한 제2 자바 애플리케이션이 소멸되는 경우에도, 제1 자바 애플리케이션은 제2 자바 애플리케이션에 의해서 설정되는 공유된 정보를 적용할 수 있다.

따라서 서로 다른 개발 주체에서 제1 자바 애플리케이션과 제2 자바 애플리케이션이 개발되더라도, 사용자 측면에서는 제1 자바 애플리케이션과 제2 자바 애플리케이션을 서로 연동되는 하나의 자바 애플리케이션으로 간주할 수 있다. 따라서 자바 애플리케이션의 확장성을 높일 수 있다.

또한 본 발명에 따르면 사용자 정치의 시스템 자원을 효율적으로 관리할 수 있다. 사용자 정치의 시스템 자원 상태에 따라서 제1 자바 애플리케이션은 제1 실행 유형 정보, 제2 실행 유형 정보 또는 제3 실행 유형 정보를 기초로 제2 자바 애플리케이션의 실행을 요청하며, 자신은 소멸 상태로 되거나 일시 정지 상태로 되어
나 백그라운드 실행 상태로 변경될 수 있다.
【특허청구범위】

【청구항 1】

사용자 장치에서의 자바 애플리케이션 관리 방법으로서,

(a) 제1 자바 애플리케이션으로부터 실행 유형 정보를 포함하는 제2 자바 애플리케이션의 실행 요청을 수신하는 단계와,

(b) 상기 실행 유형 정보에 따라서 상기 제1 자바 애플리케이션의 실행 상태를 변경하도록 제어하는 단계와,

(c) 상기 제2 자바 애플리케이션을 초기화하고 실행하도록 제어하는 단계와,

(d) 상기 제2 자바 애플리케이션으로부터 상기 제2 자바 애플리케이션의 소멸 요청을 수신하는 경우 상기 제2 자바 애플리케이션을 소멸시키도록 제어하고, 상기 변경된 상기 제1 자바 애플리케이션의 실행 상태를 재변경하여 상기 제1 자바 애플리케이션이 실행되도록 제어하는 단계

를 포함하는 자바 애플리케이션 관리 방법.

【청구항 2】

제1항에 있어서,

상기 제1 자바 애플리케이션은 사용자 입력을 수신하는 사용자 인터페이스를 포함하는 것이고,

상기 제1 자바 애플리케이션은 상기 사용자 인터페이스를 통하여 상기 사용자 입력을 수신하면 상기 제2 자바 애플리케이션의 실행 요청을 생성하는 것인 자
바 애플리케이션 관리 방법.

【청구항 3】
제1항에 있어서,
상기 제2 자바 애플리케이션의 실행 요청은 상기 제2 자바 애플리케이션의 획득 경로를 포함하는 것인 자바 애플리케이션 관리 방법.

【청구항 4】
제3항에 있어서,
상기 단계 (c)는,
상기 획득 경로를 이용하여 상기 제2 자바 애플리케이션을 획득하는 단계
를 포함하는 것인 자바 애플리케이션 관리 방법.

【청구항 5】
제1항에 있어서,
상기 실행 유형 정보는,
상기 제2 자바 애플리케이션의 실행 시 상기 제1 자바 애플리케이션을 소멸(destroy)시키고 상기 제2 자바 애플리케이션의 소멸 시 상기 제1 자바 애플리케이션을 다시 시작(start)하는 제1 실행 유형 정보와.

상기 제2 자바 애플리케이션의 실행 시 상기 제1 자바 애플리케이션의 실행을 일시 정지(pause)시키고 상기 제2 자바 애플리케이션의 소멸시 상기 일시 정지된 제1 자바 애플리케이션의 실행을 속행(unpause)하는 제2 실행 유형 정보와.
상기 제2 자바 애플리케이션의 실행시 상기 제1 자바 애플리케이션을 백그라운드(background)에서 실행시키고 상기 제2 자바 애플리케이션의 소멸시 상기 백그라운드에서 실행되던 상기 제1 자바 애플리케이션을 포어그라운드(foreground)에서 실행하는 제3 실행 유형 정보

중 어느 하나인 것인 자바 애플리케이션 관리 방법.

【청구항 6】

제5항에 있어서,

상기 단계 (b)는,

상기 실행 유형 정보가 상기 제1 실행 유형 정보이던 상기 제1 자바 애플리케이션의 실행 상태를 소멸 상태로 변경하도록 제어하고,

상기 실행 유형 정보가 상기 제2 실행 유형 정보이던 상기 제1 자바 애플리케이션의 실행 상태를 일시 정지 상태로 변경하도록 제어하고,

상기 실행 유형 정보가 상기 제3 실행 유형 정보이던 상기 제1 자바 애플리케이션의 실행 상태를 백그라운드 실행 상태로 변경하도록 제어하는 단계

를 포함하는 것인 자바 애플리케이션 관리 방법.

【청구항 7】

제5항에 있어서,

상기 단계 (d)는,

상기 실행 유형 정보가 상기 제1 실행 유형 정보이던 상기 제1 자바 애플리케이션...
케이션의 실행 상태를 시작 상태로 변경하도록 제어하고,

상기 실행 유형 정보가 상기 제2 실행 유형 정보이면 상기 제1 자바 에플리케이션의 실행 상태를 속행 상태로 변경하도록 제어하고,

상기 실행 유형 정보가 상기 제3 실행 유형 정보이면 상기 제1 자바 에플리케이션의 실행 상태를 포어그라운드 실행 상태로 변경하도록 제어하는 단계

를 포함하는 것인 자바 에플리케이션 관리 방법.

【정구항 8】

제5항에 있어서,

상기 실행 유형 정보가 상기 제2 실행 유형 정보 또는 상기 제3 실행 유형 정보로서 지정되지 않은 경우, 상기 실행 유형 정보는 상기 제1 실행 유형 정보인 것인 자바 에플리케이션 관리 방법.

【정구항 9】

제1항에 있어서,

상기 제1 자바 에플리케이션 및 상기 제2 자바 에플리케이션은 정보를 공유하는 것이고,

상기 단계 (d)는,

(d-1) 상기 제2 자바 에플리케이션에서 설정된 상기 공유된 정보를 상기 제1 자바 에플리케이션에 적용하도록 제어하는 단계

를 포함하는 것인 자바 에플리케이션 관리 방법.
【정책 항 10】

제9항에 있어서,

상기 단계 (d-1)은,

상기 제2 자바 애플리케이션과 상기 제1 자바 애플리케이션 사이의 봉신을 통하여 상기 공유된 정보를 적용하도록 제어하는 단계

를 포함하는 것인 자바 애플리케이션 관리 방법.

【정책 항 11】

제9항에 있어서,

상기 단계 (d-1)은,

상기 실행 유형 정보가 상기 제2 실행 유형 정보 또는 상기 제3 실행 유형 정보인 경우에만 상기 공유된 정보를 상기 제1 자바 애플리케이션에 적용하도록 제어하는 단계

를 포함하는 것인 자바 애플리케이션 관리 방법.

【정책 항 12】

제1항에 있어서,

상기 제1 자바 애플리케이션 또는 상기 제2 자바 애플리케이션은 Xlet 형태의 자바 애플리케이션인 것인 자바 애플리케이션 관리 방법.

【정책 항 13】

사용자 장치에서의 자바 애플리케이션 관리 방법으로서,
(a) 제1 자바 애플리케이션으로부터 제2 자바 애플리케이션의 실행 요청을 수신하는 단계와.

(b) 상기 제1 자바 애플리케이션의 실행 상태를 임시 정지 상태로 변경하도록 제어하는 단계와.

(c) 상기 제2 자바 애플리케이션을 초기화하고 실행하도록 제어하는 단계와.

(d) 상기 제2 자바 애플리케이션으로부터 상기 제2 자바 애플리케이션의 소멸 요청을 수신하는 경우 상기 제2 자바 애플리케이션을 소멸시키도록 제어하고 상기 임시 정지된 제1 자바 애플리케이션의 실행 상태를 속행 상태로 변경하도록 제어하는 단계.

를 포함하는 자바 애플리케이션 관리 방법.

【참고항 14】

제13항에 있어서.

상기 제1 자바 애플리케이션은 사용자 입력을 수신하는 사용자 인터페이스를 포함하는 것이고,

상기 제1 자바 애플리케이션은 상기 사용자 인터페이스를 통하여 상기 사용자 입력을 수신하면 상기 제2 자바 애플리케이션의 실행 요청을 생성하는 것인 자바 애플리케이션 관리 방법.

【참고항 15】

제13항에 있어서.
상기 제2 자바 애플리케이션의 실행 요청은 상기 제2 자바 애플리케이션의
취득 정보를 포함하는 것인 자바 애플리케이션 관리 방법.

【청구항 16】

제14항에 있어서,

상기 단계 (c)는,

상기 취득 정보를 이용하여 상기 제2 자바 애플리케이션을 취득하는 단계
를 포함하는 것인 자바 애플리케이션 관리 방법.

【청구항 17】

제13항에 있어서,

상기 제1 자바 애플리케이션 및 상기 제2 자바 애플리케이션은 정보를 공유
하는 것이고,

상기 단계 (d)는,

(d-1) 상기 제2 자바 애플리케이션에서 설정된 상기 공유된 정보를 상기 제1 자바 애플리케이션에 적용하도록 제어하는 단계
을 포함하는 것인 자바 애플리케이션 관리 방법.

【청구항 18】

제17항에 있어서,

상기 단계 (d-1)은,

상기 제2 자바 애플리케이션과 상기 제1 자바 애플리케이션 사이의 통신을
통하하여 상기 공유된 정보를 적응하도록 제어하는 단계

을 포함하는 것인 자바 에플리케이션 관리 방법.

【청구항 19】

제13항에 있어서.

상기 제1 자바 에플리케이션 또는 상기 제2 자바 에플리케이션은 Xlet 형태의 자바 에플리케이션인 것인 자바 에플리케이션 관리 방법.

【청구항 20】

사용자 장치에서의 자바 에플리케이션 관리 방법으로서.

(a) 제1 자바 에플리케이션으로부터 제2 자바 에플리케이션이 실행 요청을 수신하는 단계와,

(b) 상기 제1 자바 에플리케이션의 실행 상태를 백그라운드 실행 상태로 변경하도록 제어하는 단계와.

(c) 상기 제2 자바 에플리케이션을 초기화하고 실행하도록 제어하는 단계와.

(d) 상기 제2 자바 에플리케이션으로부터 상기 제2 자바 에플리케이션의 소멸 요청을 수신하는 경우 상기 제2 자바 에플리케이션을 소멸시키도록 제어하고 상기 백그라운드 실행 상태인 상기 제1 자바 에플리케이션의 실행 상태를 포어그라운드 실행 상태로 변경하도록 제어하는 단계

을 포함하는 자바 에플리케이션 관리 방법.
제품 연짜 : 2009-02-20

【청구항 21】

제20항에 있어서.

상기 제1 자바 에플리케이션은 사용자 입력을 수신하는 사용자 인터페이스를 포함하는 것이고,

상기 제1 자바 에플리케이션은 상기 사용자 인터페이스를 통하여 상기 사용자 입력을 수신하면 상기 제2 자바 에플리케이션의 실행 요청을 생성하는 일인 자바 에플리케이션 관리 방법.

【청구항 22】

제20항에 있어서.

상기 제2 자바 에플리케이션의 실행 요청은 상기 제2 자바 에플리케이션의 획득 결과를 포함하는 것인 자바 에플리케이션 관리 방법.

【청구항 23】

제22항에 있어서.

상기 단계 (c)는, 상기 획득 결과를 이용하여 상기 제2 자바 에플리케이션을 획득하는 단계

를 포함하는 것인 자바 에플리케이션 관리 방법.

【청구항 24】

제20항에 있어서.

상기 제1 자바 에플리케이션 및 상기 제2 자바 에플리케이션은 정보를 공유

55-44
하는 것이고,

상기 단계 (d)는,

(d-1) 상기 제2 자바 애플리케이션에서 설정된 상기 공유된 정보를 상기 제1 자바 애플리케이션에 적용하도록 제이하는 단계

을 포함하는 것인 자바 애플리케이션 관리 방법.

【청구항 25】

제24항에 있어서.

상기 단계 (d-1)은,

상기 제2 자바 애플리케이션과 상기 제1 자바 애플리케이션 사이의 통신을 통하여 상기 공유된 정보를 적용하도록 제이하는 단계

을 포함하는 것인 자바 애플리케이션 관리 방법.

【청구항 26】

제20항에 있어서.

상기 제1 자바 애플리케이션 또는 상기 제2 자바 애플리케이션은 Xlet 형태의 자바 애플리케이션인 것인 자바 애플리케이션 관리 방법.

【청구항 27】

제1항 내지 제26항 중 어느 한 항에 따른 자바 애플리케이션 관리 방법의 각 단계를 실현시키기 위한 프로그램을 기록한 컴퓨터로 만든 가능한 기록 매체.
【도면의 간단한 설명】

도 1은 본 발명에 따른 자바 애플리케이션 관리 방법이 실행되는 실행 환경을 나타내는 도면.

도 2는 본 발명에 따른 자바 애플리케이션 관리 방법의 예시적인 흐름도.

도 3은 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서 제1 자바 애플리케이션과 제2 자바 애플리케이션의 관계를 나타내는 도면.

도 4는 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서 실행 유형 정보에 따라서 제1 자바 애플리케이션의 실행 상태를 변경하도록 제어하는 단계를 보다 상세히 나타내는 흐름도.

도 5는 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서 실행 유형 정보에 따라서 제1 자바 애플리케이션의 실행 상태를 제정하도록 제어하는 단계를 보다 상세히 나타내는 흐름도.

도 6은 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서 애플리케이션 관리자가 제2 실행 정보를 이용하여 자바 애플리케이션의 실행 상태를 변경하는 구성의 예를 나타내는 도면.

도 7은 본 발명에 따른 자바 애플리케이션 관리 방법에 있어서 애플리케이션 관리자가 제1 실행 정보를 이용하여 자바 애플리케이션의 실행 상태를 변경하는 구성의 예를 나타내는 도면.

도 8은 본 발명에 따른 자바 애플리케이션 관리 방법의 다른 예시적인 흐름
도 9는 본 발명에 따른 자바 애플리케이션 관리 방법의 또 다른 예시적인 예를 보인다.

<table>
<thead>
<tr>
<th>번호</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>자바 애플리케이션 서버</td>
</tr>
<tr>
<td>200</td>
<td>사용자 장치</td>
</tr>
<tr>
<td>210</td>
<td>부트 프로그램</td>
</tr>
<tr>
<td>230</td>
<td>런타임 프로그램</td>
</tr>
<tr>
<td>231</td>
<td>X2let API</td>
</tr>
<tr>
<td>233</td>
<td>메소드 호출 API</td>
</tr>
<tr>
<td>235</td>
<td>자원 지원 API</td>
</tr>
<tr>
<td>237</td>
<td>애플리케이션 관리자 AP</td>
</tr>
<tr>
<td>239</td>
<td>디버그 지원 툴</td>
</tr>
<tr>
<td>290</td>
<td>자바 애플리케이션</td>
</tr>
</tbody>
</table>
【도 2】

시작

제1자바 애플리케이션으로부터 실행 유형 정보를 포함하는
제2자바 애플리케이션의 실행 요청을 수신

S110

실행 유형 정보에 따라서 제1자바 애플리케이션의 실행 상태를
변경하도록 제어

S130

특정 경로를 이용하여 제2자바 애플리케이션을 확득하고
제2자바 애플리케이션을 초기화하고 실행하도록 제어

S150

제2자바 애플리케이션의 소멸 요청을 수신하면, 제2자바
애플리케이션을 소멸시키도록 제어하고, 제1자바
애플리케이션의 실행 상태를 재변경하여 실행하도록 제어

S170

종료

【도 3】

루트의 실행 요청에
따른 자식1 실행

루트

자식1의 실행 요청에
따른 자식2 실행

자식1

자식2
【도 4】

S130 시작

설행 유형 정보 = 제1 실행 유형 정보?

아니오 S134
설행 유형 정보 = 제2 실행 유형 정보?

아니오 S137
설행 유형 정보 = 제3 실행 유형 정보?

아니오

예

S132
제1차례 예상케이선의 실행 상태를
소멸 상태로 변경

예

S135
제1차례 예상케이선의 실행 상태를
일시 정지 상태로 변경

예

S138
제1차례 예상케이선의 실행 상태를
백그라운드 실행 상태로 변경

S130 종료
【도 7】
【도 8】

시작

S210

제1 자바 애플리케이션으로부터
제2 자바 애플리케이션의 실행 요청을 수신

S230

제1 자바 애플리케이션의 실행 상태를
일시 정지 상태로 변경하도록 제어

S250

특정 경로를 이용하여 제2 자바 애플리케이션을 획득하고,
제2 자바 애플리케이션을 초기화하고 실행하도록 제어

S270

제2 자바 애플리케이션의 소멸 요청을 수신하면, 제2 자바
애플리케이션을 소멸시켜도록 제어하고, 제1 자바
애플리케이션의 실행 상태를 속행 상태로 변경하도록 제어,
공유 정보 적용

종료
【도 9】

시작

제1차례 애플리케이션으로부터
제2 차례 애플리케이션의 실행 요청을 수신

S310

제1 차례 애플리케이션의 실행 상태를
백그라운드 실행 상태로 변경하도록 제어

S330

특목 경로를 이용하여 제2 차례 애플리케이션을 확득하고,
제2 차례 애플리케이션을 초기화하고 실행하도록 제어

S350

제2 차례 애플리케이션의 소멸 요청을 수신하면, 제2 차례
애플리케이션을 소멸시키도록 제어하고, 제1 차례
애플리케이션의 실행 상태를 포어그라운드 실행 상태로
변경하도록 제어, 공유 정보 삭제

S370

종료
<table>
<thead>
<tr>
<th>APPLICATION NUMBER</th>
<th>FILING or 371(c) DATE</th>
<th>CRF ART UNIT</th>
<th>FEE REC'D</th>
<th>ATTY.DOCKET.NO</th>
<th>TOT CLAIMS</th>
<th>IND CLAIMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/656,872</td>
<td>02/18/2010</td>
<td>2192</td>
<td>597</td>
<td>30721U</td>
<td>22</td>
<td>3</td>
</tr>
</tbody>
</table>

CONFIRMATION NO. 2380

20529
THE NATH LAW GROUP
112 South West Street
Alexandria, VA 22314

Date Mailed: 03/10/2010

Receipt is acknowledged of this non-provisional patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. If an error is noted on this Filing Receipt, please submit a written request for a Filing Receipt Correction. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections.

Applicant(s)

Wonjang Baek, Seongnam-si, KOREA, REPUBLIC OF;
John Kim, Seoul, KOREA, REPUBLIC OF;

Assignment For Published Patent Application
DREAMER, Burbank, CA

Power of Attorney: The patent practitioners associated with Customer Number 20529

Domestic Priority data as claimed by applicant

Foreign Applications
REPUBLIC OF KOREA 10-2009-0014161 02/20/2009

Request to Retrieve - This application either claims priority to one or more applications filed in an intellectual property Office that participates in the Priority Document Exchange (PDX) program or contains a proper Request to Retrieve Electronic Priority Application(s) (PTO/SB/38 or its equivalent). Consequently, the USPTO will attempt to electronically retrieve these priority documents.

If Required, Foreign Filing License Granted: 03/08/2010

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is US 12/656,872

Projected Publication Date: 08/26/2010

Non-Publication Request: No

Early Publication Request: No
** SMALL ENTITY **
PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process simplifies the filing of patent applications on the same invention in member countries, but does not result in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at http://www.uspto.gov/web/offices/pac/doc/general/index.html.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, http://www.stopfakes.gov. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4158).

LICENSE FOR FOREIGN FILING UNDER
Title 35, United States Code, Section 184
Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as
set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign AssetsControl, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).
# Utility Patent Application Transmittal

(Only for new nonprovisional applications under 37 CFR 1.53(b))

**APPLICATION ELEMENTS**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Fee Transmittal Form (e.g., PTO/SB/17) (Submit an original and a duplicate for fee processing)</td>
</tr>
<tr>
<td>2.</td>
<td>Applicant claims small entity status. See 37 CFR 1.27.</td>
</tr>
<tr>
<td>3.</td>
<td>Specification [Total Pages 28] Both the claims and abstract must start on a new page (For information on the preferred arrangement, see MPEP 608.01(a))</td>
</tr>
<tr>
<td>4.</td>
<td>Drawing(s) (35 U.S.C. 113) [Total Sheets 9]</td>
</tr>
<tr>
<td>5.</td>
<td>Oath or Declaration [Total Sheets 2] a. Newly executed (original or copy) b. A copy from a prior application (37 CFR 1.63(d)) for continuation/divisional with Box 18 completed i. DELETION OF INVENTOR(S) Signed statement attached deleting inventor(s) name in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).</td>
</tr>
<tr>
<td>6.</td>
<td>Application Data Sheet. See 37 CFR 1.76</td>
</tr>
<tr>
<td>7.</td>
<td>CD-ROM or CD-R in duplicate, large table or Computer Program (Appendix) Landscape Table on CD</td>
</tr>
<tr>
<td>8.</td>
<td>Nucleotide and/or Amino Acid Sequence Submission (If applicable, items a. c. are required) a. Computer Readable Form (CRF) b. Specification Sequence Listing on i. CD-ROM or CD-R (2 copies); or ii. Paper c. Statements verifying identity of above copies</td>
</tr>
</tbody>
</table>

**ADDRESS TO:** Commissioner for Patents P.O. Box 1450 Alexandria VA 22313-1450

**ACCOMPANYING APPLICATION PARTS**

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.</td>
<td>Assignment Papers (cover sheet &amp; document(s)) Name of Assignee DREAMER</td>
</tr>
<tr>
<td>10.</td>
<td>37 CFR 3.73(b) Statement (when there is an assignee) Power of Attorney</td>
</tr>
<tr>
<td>11.</td>
<td>English Translation Document (if applicable)</td>
</tr>
<tr>
<td>12.</td>
<td>Information Disclosure Statement (PTO/SB/08 or PTO-1449) Copies of foreign patent documents, publications, &amp; other information</td>
</tr>
<tr>
<td>13.</td>
<td>Preliminary Amendment</td>
</tr>
<tr>
<td>14.</td>
<td>Return Receipt Postcard (MPEP 503) (Should be specifically itemized)</td>
</tr>
<tr>
<td>15.</td>
<td>Certified Copy of Priority Document(s) (if foreign priority is claimed)</td>
</tr>
<tr>
<td>17.</td>
<td>Other: Transmittal Letter, Check</td>
</tr>
</tbody>
</table>

18. If a CONTINUING APPLICATION, check appropriate box, and supply the requisite information below and in the first sentence of the specification following the title, or in an Application Data Sheet under 37 CFR 1.76:

- [ ] Continuation
- [ ] Divisional
- [ ] Continuation-in-part (CIP) of prior application No.: Art Unit: 

**19. CORRESPONDENCE ADDRESS**

- [ ] The address associated with Customer Number: 20529 OR [ ] Correspondence address below

**Name**

**Address**

**City**

**State**

**Zip Code**

**Country**

**Telephone**

**Email**

**Signature**

**Name (Print/Type)** Jerald L. Meyer

**Date** February 8, 2010

**Registration No. (Attorney/Agent)** 41,194

This collection of information is required by 37 CFR 1.53(b). The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Word Version Copyright 2007 Forms in Word (www.formsintword.com)
**FEE TRANSMITTAL**

**For FY 2009**

- **Applicant claims small entity status. See 37 CFR 1.27**
- **Total Amount of Payment ($):** 637.00

**METHOD OF PAYMENT**
- Check [ ]
- Credit Card [ ]
- Money Order [ ]
- None [ ]
- Other (please identify): 
  - Deposit Account [ ]
  - Deposit Account Number: 14-0112
  - Deposit Account Name: THE NATH LAW GROUP

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

- Charge fees(s) indicated below
- Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17
- Credit any overpayments

**WARNING:** Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

**FEE CALCULATION**

1. **BASIC FILING, SEARCH, AND EXAMINATION FEES**
   - **FILING FEES**
   - **SEARCH FEES**
   - **EXAMINATION FEES**

<table>
<thead>
<tr>
<th>Application Type</th>
<th>Small Entity Fee ($)</th>
<th>Small Entity Fee ($)</th>
<th>Small Entity Fee ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Utility</td>
<td>330</td>
<td>165</td>
<td>540</td>
</tr>
<tr>
<td>Design</td>
<td>220</td>
<td>110</td>
<td>100</td>
</tr>
<tr>
<td>Plant</td>
<td>220</td>
<td>110</td>
<td>330</td>
</tr>
<tr>
<td>Reissue</td>
<td>330</td>
<td>165</td>
<td>540</td>
</tr>
<tr>
<td>Provisional</td>
<td>220</td>
<td>110</td>
<td>0</td>
</tr>
</tbody>
</table>

2. **EXCESS CLAIM FEES**
   - **Fee Description**
     - Each claim over 20 (including Reissues)
     - Each independent claim over 3 (including Reissues)
     - Multiple dependent claims

<table>
<thead>
<tr>
<th>Total Claims</th>
<th>Extra Claims Fee ($)</th>
<th>Fee Paid ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>22, 20 or HP</td>
<td>2</td>
<td>52</td>
</tr>
<tr>
<td>23 or HP</td>
<td>0</td>
<td>26</td>
</tr>
<tr>
<td>3 or HP</td>
<td>0</td>
<td>390</td>
</tr>
</tbody>
</table>

   | HP = highest number of total claims paid for, if greater than 20. |
   | Multiple Dependent Claims Fee ($) | Fee Paid ($) |
   | 195         | 0.00          |

3. **APPLICATION SIZE FEE**
   - If the specification and drawings exceed 100 sheets of paper (excluding electronically filed sequence or computer listings under 37 CFR 1.52(e)), the application size fee due is $270 ($135 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

<table>
<thead>
<tr>
<th>Total Sheets</th>
<th>Extra Sheets Fee ($)</th>
<th>Fee Paid ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>37</td>
<td>100</td>
<td>0</td>
</tr>
</tbody>
</table>

   | Number of each additional 50 or fraction thereof | Fee ($) | Fee Paid ($) |
   | 0          | 135                  | 0.00         |

4. **OTHER FEE($)**
   - Non-English Specification, $130 fee (no small entity discount)
   - Other (e.g., late filing surcharge): Assignment Recordation Fee

   - $40.00

**SUBMITTED BY**

- **Signature: Jerald L. Meyer**
- **Registration No.: 41,194**
- **Telephone: (703) 548-6284**
- **Date: February 18, 2010**

---

This collection of information is required by 37 CFR 1.136. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 30 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.
IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Wonjang BAEK, et al. Conf. No.: Not Yet Assigned

Appl. No.: Not Yet Assigned Examiner: Not Yet Assigned

Filed: February 18, 2010 Group Art Unit: Not Yet Assigned

For: METHOD FOR MANAGING JAVA APPLICATIONS

TRANSMITTAL LETTER

Commissioner for Patents
P.O. Box 1450
Alexandria, VA  22313-1450

Dear Commissioner:

Submitted herewith for filing in the U.S. Patent and Trademark Office is the following:

(1) PTO/SB/05 (Utility Patent Application Transmittal), one (1) page;

(2) PTO/SB/17 (Fee Transmittal), one (1) page;

(3) Application Data Sheet, five (5) pages;

(4) Utility Patent Application, thirty-seven (37) pages, consisting of:
    21 pages of Textual Specification,
    6 pages of 22 Claims,
    1 page for the Abstract, and
    9 sheets of Drawings;

(5) One (1) Executed Declaration and Power of Attorney, two (2) pages;

(6) One (1) Executed Assignment, two (2) pages, with a Recordation Cover Sheet, one (1) page, in favor of DREAMER of Burbank, California;
(7) Our check no. 09660 in the amount of $637.00 for the statutory filing ($165), search ($270), examination ($110), and excess claim ($52) fees for a small entity, as well as the recordation fee ($40); and

(8) Request for Early Notification of the Serial Number.

Please charge any fee deficiency or credit any overpayment to Deposit Account No. 14-0112.

Respectfully submitted,

THE NATH LAW GROUP

[Signature]

Jerald L. Meyer
Registration No. 41,194
Derek Richmond
Registration No. 45,771
Customer No. 20529

Date: February 18, 2010

THE NATH LAW GROUP
112 S. West Street
Alexandria, Virginia 22314
Tel: (703) 548-6284
Fax: (703) 683-8396

JLM/DR/bd
APPLICATION DATA SHEET

Application Information

Application Number: Not Yet Assigned
Filing Date: February 18, 2010
Application Type: Non-Provisional
Subject Matter: Utility
Suggested Classification:
Suggested Group Art Unit:
CD-ROM or CD-R?: None
Number of CD disks: 0
Number of copies of CDs: 0
Sequence submission?: No
Computer Readable Form (CRF): No
Number of copies of CRF: 0
Title: METHOD FOR MANAGING JAVA APPLICATIONS
Attorney Docket Number: 30721U
Request for Early Publication?: No
Request for Non-Publication?: No
Suggested Drawing Figure:
Total Drawing Sheets: 9
Small Entity?: Yes
Latin name:
Variety denomination name:

Page # 1
Initial 2/17/2010
Petition included?: No
Petition Type: N/a
Licensed U.S. Govt. Agency: N/a
Contract or Grant Numbers: N/a
Secrecy Order in Parent Appl.: No

**Applicant Information (1)**

Applicant Authority type: Inventor
Primary Citizenship Country: KR
Status: Full Capacity
Given Name: Wonjang
Middle Name: 
Family Name: BAEK
Name Suffix: 
City of Residence: Seongnam-si, Gyeonggi-do
State or Province of Residence: 
Country of Residence: KR
Street of Mailing address: 501-401 AreumMaeul Kunyoung Apts.
107-402, Imae-dong, Bundang-gu
City of mailing address: Seongnam-si, Gyeonggi-do
State/Province of mailing address: 
Country of mailing address: KR
Postal Code of mailing address: 463-731
Applicant Information (2)

Applicant Authority type:: Inventor
Primary Citizenship Country:: KR
Status:: Full Capacity
Given Name:: John
Middle Name::
Family Name:: KIM
Name Suffix::
City of Residence:: Seoul
State or Province of Residence::
Country of Residence:: KR
Street of Mailing address:: SK HUB SU B-1416, 650-4, Guro 1-dong,
                           Guro-gu
City of mailing address:: Seoul
State/Province of mailing address::
Country of mailing address:: KR
Postal Code of mailing address:: 152-720
Correspondence Information

Correspondence Customer Number:: 20529
Name:: THE NATH LAW GROUP
Street of mailing address:: 112 S. West Street
City of mailing address:: Alexandria
State or Province of mailing address:: Virginia
Country of mailing address:: United States of America
Postal or Zip Code of mailing address:: 22314
Phone number:: (703) 548-6284
Fax number:: (703) 683-8396
E-Mail address:: ip@nathlaw.com

Representative Information

Representative Customer Number:: 20529

Domestic Priority Information

Application No:: Continuity Type:: Prior Application No:: Filing Date

(MM/DD/YY)::
Foreign Priority Information

Application No.:: 10-2009-0014161
Country:: KR
Filing Date:: 02/20/09
Priority Claimed:: Yes

Assignee Information

Assignee name:: DREAMER
Street of mailing address:: 3500 W. Olive Avenue, Suite 990
City of mailing address:: Burbank
State/Province of mailing address:: CA
Country of mailing address:: US
Postal Code of mailing address:: 91505

Signature::

First Name:: Jerald L. Last Name:: Meyer
Registration No.:: 41194 Date (MM/DD/YY):: 02/8/10
METHOD FOR MANAGING JAVA APPLICATIONS

This application claims the benefit of Korean Patent Application No. 10-2009-0014161 filed on February 20, 2009, which is hereby incorporated for reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to a method for managing java applications executable in a user device.

2. Description of the Related Art

Conventional televisions or conventional DVD players are only capable of playing audio/video data. However, latest digital televisions and blu-ray players are capable of providing an environment for executing applications.

Latest mobile communication terminals are also capable of providing the environment.

The applications executable in user devices such as the digital televisions, blu-ray players and mobile communication terminal is usually created using Java language (hereinafter referred to as “Java application”) and provided to the user device in a byte code format.

Operations such as a start and an end of the application and an access to system
resources are carried out between the application and an operating system on which the application is executed.

An application model defines how the application should be managed and also defines roles of the operating system and the application.

Java Specifications such as J2ME (Java 2 Platform Micro Edition) and J2SE (Java 2 Platform Sun Edition) currently supports various application models. Applet and Xlet are examples of the application model.

Sun Microsystems provides Xlet for executing the java application in an embedded device such as the digital television. While Applet is an embedded application which is executed in a browser, Xlet is an embedded application executed in the embedded device.

Generally, one java application is executed and displayed on a single screen of the user device.

For instance, when another java application is selected for execution while a startup java application is executed and displayed, the startup java application is then halted and the selected java application is executed.

In other words, the java applications executed in the user device are independent of each other. Therefore, it is difficult to guarantee a continuity between the java applications because information between the independent java applications cannot be shared.
SUMMARY OF THE INVENTION

It is an object of the present invention to provide a method for managing java applications wherein an expandability for and a continuity between java applications are provided by changing states of the java applications in execution and sharing information between the java applications.

In order to achieve above-described objects of the present invention, there is provided a method for managing java applications executable in a user device, the method comprising steps of: (a) issuing a request for executing a second java application, the request being issued by a first java application and including a execution type information; (b) changing a state of the first java application according to the execution type information; (c) executing the second java application; and (d) killing the second java application when a request for killing the second java application is issued by the second java application and re-changing the state of the first java application changed in the step (b).

Preferably, the first java application generates the request for executing the second java application based on a user input received through a user interface.

Preferably, the request for executing the second java application further includes a path for obtaining the second java application.

Preferably, the step (c) comprises executing the second java application by obtaining the second java application at a location defined by the path.

Preferably, the execution type information designates one of: a first execution type
wherein the first java application is killed when the second java application is executed and the first java application is restarted when the second java application is killed; a second execution type wherein the first java application is paused when the second java application is executed and the first java application is unpaused when the second java application is killed; and a third execution type wherein the first java application is switched to background when the second java application is executed and the first java application is switched to foreground when the second java application is killed.

Preferably, the step (b) comprises: killing the first java application when the execution type information designates the first execution type; pausing the first java application when the execution type information designates the second execution type; and switching the first java application to background when the execution type information designates the third execution type.

Preferably, the step (d) comprises: restarting the first java application when the execution type information designates the first execution type; unpausing the first java application when the execution type information designates the second execution type; and switching the first java application to foreground when the execution type information designates the third execution type.

Preferably, the first java application and the second java application inter-relays a shared information.

Preferably, the first java application and the second java application inter-relays a
shared information when the execution type information designates one of the second execution type and the third execution type.

Preferably, each of the first java application and the second java application includes a java application of Xlet format.

There is also provided a method for managing java applications executable in a user device, the method comprising steps of: (a) issuing a request for executing a second java application, the request being issued by a first java application; (b) pausing the first java application; (c) executing the second java application; and (d) killing the second java application when a request for killing the second java application is issued by the second java application and unpausing the first java application paused in the step (b).

Preferably, the first java application generates the request for executing the second java application based on a user input received through a user interface.

Preferably, the request for executing the second java application further includes a path for obtaining the second java application.

Preferably, the step (c) comprises executing the second java application by obtaining the second java application at a location defined by the path.

Preferably, the first java application and the second java application inter-relays a shared information.

Preferably, each of the first java application and the second java application includes a java application of Xlet format.
There is also provided a method for managing java applications executable in a user device, the method comprising steps of: (a) issuing a request for executing a second java application, the request being issued by a first java application; (b) switching the first java application to background; (c) executing the second java application; and (d) killing the second java application when a request for killing the second java application is issued by the second java application and switching the first java application in background to foreground.

Preferably, the first java application generates the request for executing the second java application based on a user input received through a user interface.

Preferably, the request for executing the second java application further includes a path for obtaining the second java application.

Preferably, the step (c) comprises executing the second java application by obtaining the second java application at a location defined by the path.

Preferably, the first java application and the second java application inter-relays a shared information.

Preferably, each of the first java application and the second java application includes a java application of Xlet format.

**BRIEF DESCRIPTION OF THE DRAWINGS**

Fig. 1 is block diagram illustrating an environment for executing a method for managing a java application in accordance with the present invention.
Fig. 2 is a flow diagram illustrating a first embodiment of a method for managing a java application in accordance with the present invention.

Fig. 3 is a diagram illustrating a relationship between a first java application and a second java application in accordance with the present invention.

Fig. 4 is a flow diagram illustrating a change in a state of a first java application based on an execution type information in accordance with the present invention.

Fig. 5 is a flow diagram illustrating re-changing a state of a first java application based on an execution type information in accordance with the present invention.

Fig. 6 is a diagram illustrating a configuration wherein a state of a java application is changed based on a second execution type in accordance with the present invention.

Fig. 7 is a diagram illustrating a configuration wherein a state of a java application is changed based on a first execution type in accordance with the present invention.

Fig. 8 is a flow diagram illustrating a second embodiment of a method for managing a java application in accordance with the present invention.

Fig. 9 is a flow diagram illustrating a third embodiment of a method for managing a java application in accordance with the present invention.

**DETAILED DESCRIPTION OF THE INVENTION**

A method for managing a java application in accordance with the present invention will be described with reference to accompanied drawings.

Fig. 1 is block diagram illustrating an environment for executing a method for
managing a java application in accordance with the present invention.

Referring to Fig. 1, the environment includes a java application server 100 and a user device 200.

The java application server 100 stores a java application and transmits the java application to the user device 200 when requested by the user device 200.

The user device 200 executes the java application received from the java application server 100 to be provided to a user. The user device 200 may include a digital television, a set-top box, a blu-ray player or a mobile communication terminal.

The java application server 100 and the user device 200 are connected via a communication network such as the Internet and a wireless network.

Hereinafter, an application developer refers to a person who develops the java application which may be executed in the java application server 100 or the user device 200, the user refers to a person who uses the user device 200 and takes advantage of a service provided via the java application developed by the application developer.

The user device 200 comprises a boot program 210, a run-time program 230 and a java application 290.

The boot program 210 is stored within the user device 200 or in a storage medium accessible by the user device 200. The boot program 210 checks a status of the user device 200 and searches and obtains the suitable run-time program 230.

The run-time program 230 provides a run-time environment for the java
application 290.

The run-time program 230 includes an X2let API 231, a method invocation API 233, a resource support API 235, an application manager API 237 and a debug support tool 239.

The X2let API 231 developed by the Applicant is implemented by an enhanced Xlet in compliance with Java TV standard.

The X2let API 231 supports downloading of the java application 290 from the java application server 100 and execution of the downloaded java application 290.

The method invocation API 233 developed by the Applicant executes a method in the java application server 100 from the user device 200.

The resource support API 235 developed by the Applicant supports a memory-effective caching and a resource downloading. The resource support API 235 may handle resources such as an image, a byte array, a sound and a java class.

The application manager API 237 developed by the Applicant manages a life cycle of the java application 290. The java application 290 may be started, paused, unpause, switched to background or foreground and killed via the application manager API 237.

The debug support tool 239 237 developed by the Applicant supports debugging.

The java application 290 may be classified into a root java application 293 and a user java application 296.

The root java application 293 is a base application executed in the user device 200.
The root java application 293 is required to be executed in the user device 200. The root java application 293 is executed at a startup of the user device 200 and displayed on a screen of the user device 200. The root java application 293 may include a user interface for selecting the user java application 296 to be executed. The user interface may be implemented as a graphical user interface. The user may select the user java application 296 to be executed via the user interface.

The user java application 296 is selected by the user and then executed. The user java application 296 refers to the java application other than the root java application 293.

While the root java application 293 is created by a manufacturer of the user device 200 and distributed with the user device 200, the user java application 296 is created by the application developer and transmitted to the user device 200 through the java application server 100.

The method in accordance with the present invention will be described in detail hereinafter.

Fig. 2 is a flow diagram illustrating a first embodiment of the method for managing the java application in accordance with the present invention.

Referring to Fig. 2, a request for executing a second java application generated by a first java application is issued (S110).

The request includes an execution type information.

The first java application may be the root java application denoted as reference
numeral 293 in Fig. 1 or the user java application denoted as reference numeral 296 in Fig. 1. The second java application may be the user java application. It is preferable that each of the first java application and the second java application includes the java application of Xlet format executable in the user device 200.

Fig. 3 is a diagram illustrating a relationship between the first java application and the second java application in accordance with the present invention.

When the root java application issues the request for executing a first child java application, the root java application and first child java application correspond to the first java application and the second java application, respectively.

When first child java application issues the request for executing a second child java application, first child java application and the second child java application correspond to the first java application and the second java application, respectively.

The first java application may include the user interface for receiving a user input.

The user interface may be the graphical user interface for executing the second java application. When the user input is received while executing the first java application, the first java application generates and issues the request for executing the second java application.

The request includes the execution type information which designates an execution type.

The execution type includes one of a first execution type, a second execution type
and a third execution type.

In accordance with the first execution type, the first java application is killed when the second java application is executed and the first java application is restarted when the second java application is killed.

In accordance with the second execution type, the first java application is paused when the second java application is executed and the first java application is unpanded when the second java application is killed.

In accordance with the third execution type, the first java application is switched to background when the second java application is executed and the first java application is switched back to foreground when the second java application is killed.

In accordance with the present invention, a state of the first java application is determined when the second java application is executed.

When the execution type information does not designate certain the execution types, it may be regarded that the execution type information designates the first execution type.

That is, when the execution type information included in the request that does not designate the certain the execution type is issued, subsequent steps are carried out on an assumption that the execution type is the first execution type.

The request may also include a path for obtaining the second java application.

When the second java application is stored within the user device, the path may
represent a location of the second java application in the user device.

    When the second java application is stored in the java application server, the path may represent a location of the second java application in the java application server.

    The path may have a form of an URL (uniform resource locator).

    Thereafter, the state of the first java application is changed according to the execution type information included in the request issued in the step S110 (S130).

    Fig. 4 is a flow diagram illustrating the step S130 in detail wherein the state of the first java application is changed based on the execution type information.

    Referring to Fig. 4, when the execution type designated in the execution type information is the first execution type, the first java application is killed (S131 and S132).

    When the execution type designated in the execution type information is the second execution type, the first java application is paused (S134 and S135).

    When the execution type designated in the execution type information is the third execution type, the first java application is switched to background (S137 and S138).

    Referring back to Fig. 2, the second java application is obtained and executed in parallel with or after carrying out the step S130 (S150).

    Thereafter, When a request for killing the second java application is issued while the second java application is executed, the second java application is killed and the first java application state of which is changed in the step S130 is re-changed (S170).

    Fig. 5 is a flow diagram illustrating re-changing the state of the first java
application based on the execution type information.

Referring to Fig. 5, when the execution type designated in the execution type information is the first execution type, the first java application is re-started (S171 and S172).

When the execution type designated in the execution type information is the second execution type, the paused first java application is unpaused (S174 and S175).

When the execution type designated in the execution type information is the third execution type, the first java application in the background is switched back to foreground (S177 and S178).

The first java application and the second java application may relay a shared information to each other, i.e., inter-relay the shared information.

Specifically, if the second java application executed in the step S150 obtains the shared information that is to be shared with the first java application, the second java application may relay the shared information to the first java application in the step S170, and the first java application may use the relayed information.

In one embodiment, when the first java application receives the user input and the received user input includes a certain information which corresponds to the shared information, the first java application transmits the certain information included in the user input to the second java application, and the second java application may use the certain information.
Similarly, when the second java application receives the user input and the received user input includes a certain information which corresponds to the shared information, the second java application transmits the certain information included in the user input to the first java application, and the first java application may use the certain information.

Particularly, the shared information may be inter-relayed when the execution type is the second execution type or the third execution type.

Fig. 6 is a diagram illustrating a configuration wherein the state of the java application is changed based on the second execution type.

An application manager shown in Fig. 6 implements the application manager API shown in Fig. 1.

Referring to Fig. 6, the root java application transmits a request for executing the first child java application to the application manager.

Here, the root java application and the first child java application correspond to the first java application and the second java application, respectively.

The request for executing the first child java application is denoted as startApp(root, Bound, c1URL) in Fig. 6.

startApp(root, Bound, c1URL) represents the request for executing, by the root java application (denoted as “root”), the first child java application located at a path “c1URL” according to the second execution type (denoted as “Bound”).
When the request `startApp(root, Bound, c1URL)` is received, the application manager pauses the root java application using `pauseXlet()` function.

The application manager also initializes the first child java application using `initXlet()` function, and starts the first child java application using `startXlet()` function.

Thereafter, the first child java application transmits a request for executing the second child java application to the application manager.

Here, the first child java application and the second child java application correspond to the first java application and the second java application, respectively.

The request for executing the second child java application is denoted as `startApp(child1, Bound, c2URL)` in Fig. 6.

`startApp(child1, Bound, c2URL)` represents the request for executing, by the first child java application (denoted as "child1"), the second child java application located at a path "c2URL" according to the second execution type (denoted as "Bound").

When the request `startApp(child1, Bound, c2URL)` is received, the application manager pauses the root java application using `pauseXlet()` function.

The application manager also initializes the second child java application using `initXlet()` function, and starts the second child java application using `startXlet()` function.

When `destroyApp(child2)` which is a request for killing the second child java application is received from the second child java application, the application manager kills the second child java application using `destroyXlet()` function.
When the second child java application is killed, the application manager unpauses the paused first child java application. In other words, the application manager starts the first child java application using startXlet() function.

When destroyApp(child1) which is a request for killing the first child java application is received from the first child java application, the application manager kills the first child java application using destroyXlet() function.

When the first child java application is killed, the application manager unpauses the paused root java application. In other words, the application manager starts the root java application using startXlet() function.

Fig. 7 is a diagram illustrating a configuration wherein the state of the java application is changed based on the first execution type.

Referring to Fig. 7, the root java application transmits a request for executing a java application "A" to the application manager.

The request for executing the java application "A" is denoted as startApp(root, Standard, aURL) in Fig. 7.

startApp(root, Standard, aURL) represents the request for executing, by the root java application (denoted as "root"), the java application "A" located at a path "aURL" according to the first execution type (denoted as "Standard").

When the request startApp(root, Standard, aURL) is received, the application manager kills the root java application using destroyXlet() function.
The application manager also initializes the java application "A" using initXlet() function, and starts the java application "A" using startXlet() function.

When destroyApp(A) which is a request for killing the java application "A" is received from the java application "A", the application manager kills the java application "A" using destroyXlet() function.

When the java application "A" is killed, the application manager restarts the root java application killed by destroyXlet() function. In other words, the application manager starts the root java application using startXlet() function.

Thereafter, the root java application transmits a request for executing a java application "B" to the application manager.

The request for executing the java application "B" is denoted as startApp(root, Standard, bURL) in Fig. 7.

startApp(root, Standard, bURL) represents the request for executing, by the root java application (denoted as "root"), the java application "B" located at a path "bURL" according to the first execution type (denoted as "Standard").

When the request startApp(root, Standard, bURL) is received, the application manager kills the root java application using destroyXlet() function.

The application manager also initializes the java application "B" using initXlet() function, and starts the java application "B" using startXlet() function.

When destroyApp(B) which is a request for killing the java application "B" is
received from the java application "B", the application manager kills the java application "B" using destroyXlet() function.

When the java application "B" is killed, the application manager restarts the root java application killed by destroyXlet() function. In other words, the application manager starts the root java application using startXlet() function.

A configuration for changing the state of the java application based on the third execution type is similar to that shown in Fig. 6.

When the state of the java application is changed based on the third execution type, background() function for switching the java application to the background is used in place of pauseXlet() function, and the java application in the background is then switched to foreground.

Fig. 8 is a flow diagram illustrating a second embodiment of the method for managing the java application in accordance with the present invention.

Referring to Fig. 8, a request for executing the second java application generated by the first java application is issued (S210).

Thereafter, the first java application is paused (S230).

Thereafter, the second java application is obtained and executed in parallel with or after carrying out the step S230 (S250).

When a request for killing the second java application is issued while the second java application is executed, the second java application is killed and the first java
application paused in the step S230 is unpaused (S270).

In accordance with the second embodiment, the execution type information is not included in the request for executing the second java application. The second embodiment is identical to the first embodiment except that the execution type information is not included in the request. Therefore, a detailed description is omitted.

Fig. 9 is a flow diagram illustrating a third embodiment of the method for managing the java application in accordance with the present invention.

Referring to Fig. 9, a request for executing the second java application generated by the first java application is issued (S310).

Thereafter, the first java application is switched to background (S330).

Thereafter, the second java application is obtained and executed in parallel with or after carrying out the step S330 (S350).

When a request for killing the second java application is issued while the second java application is executed, the second java application is killed and the first java application in the background is switched to foreground (S370).

In accordance with the third embodiment, the execution type information is not included in the request for executing the second java application. The third embodiment is identical to the second embodiment except that the first java application is switched to background instead of being paused. Therefore, a detailed description is omitted.

In accordance with the present invention, because the first java application may be
associated with the second java application, a use of the java application for the user is facilitated, and the java applications developed by different application developers can be associated with each other in order to improve an expandability. For instance, the application developer may create the first java application in a manner that the first java application includes a path for the second java application in order for the user to take advantage of the second java application.

Particularly, the present invention may be applied to base applications of TV portal services, smart phones and blu-ray players employing the java application.

Various services may be provided based on the execution type information and the path even when the base application does not include every function.

Moreover, the first java application and the second java application inter-relays the shared information in order to guarantee a continuity therebetween.

While the present invention has been particularly shown and described with reference to the preferred embodiment thereof, it will be understood by those skilled in the art that various changes in form and details may be effected therein without departing from the spirit and scope of the invention as defined by the appended claims.
What is claimed is:

1. A method for managing java applications executable in a user device, the method comprising steps of:
   
   (a) issuing a request for executing a second java application, the request being issued by a first java application and including a execution type information;
   
   (b) changing a state of the first java application according to the execution type information;
   
   (c) executing the second java application; and
   
   (d) killing the second java application when a request for killing the second java application is issued by the second java application and re-changing the state of the first java application changed in the step (b).

2. The method in accordance with claim 1, wherein the first java application generates the request for executing the second java application based on a user input received through a user interface.

3. The method in accordance with claim 1, wherein the request for executing the second java application further includes a path for obtaining the second java application.

4. The method in accordance with claim 3, wherein the step (c) comprises executing the second java application by obtaining the second java application at a location defined
by the path.

5. The method in accordance with claim 1, wherein the execution type information designates one of:

   a first execution type wherein the first java application is killed when the second java application is executed and the first java application is restarted when the second java application is killed;

   a second execution type wherein the first java application is paused when the second java application is executed and the first java application is unpaused when the second java application is killed; and

   a third execution type wherein the first java application is switched to background when the second java application is executed and the first java application is switched to foreground when the second java application is killed.

6. The method in accordance with claim 5, wherein the step (b) comprises:

   killing the first java application when the execution type information designates the first execution type;

   pausing the first java application when the execution type information designates the second execution type; and

   switching the first java application to background when the execution type information designates the third execution type.
7. The method in accordance with claim 5, wherein the step (d) comprises:

restarting the first java application when the execution type information designates
the first execution type;

unpausing the first java application when the execution type information designates
the second execution type; and

switching the first java application to foreground when the execution type
information designates the third execution type.

8. The method in accordance with claim 1, wherein the first java application and the
second java application inter-relays a shared information.

9. The method in accordance with claim 7, wherein the first java application and the
second java application inter-relays a shared information when the execution type
information designates one of the second execution type and the third execution type.

10. The method in accordance with claim 1, wherein each of the first java application
and the second java application includes a java application of Xlet format.

11. A method for managing java applications executable in a user device, the method
comprising steps of:
(a) issuing a request for executing a second java application, the request being issued by a first java application;

(b) pausing the first java application;

(c) executing the second java application; and

(d) killing the second java application when a request for killing the second java application is issued by the second java application and unpausing the first java application paused in the step (b).

12. The method in accordance with claim 11, wherein the first java application generates the request for executing the second java application based on a user input received through a user interface.

13. The method in accordance with claim 11, wherein the request for executing the second java application further includes a path for obtaining the second java application.

14. The method in accordance with claim 13, wherein the step (c) comprises executing the second java application by obtaining the second java application at a location defined by the path.

15. The method in accordance with claim 11, wherein the first java application and the second java application inter-relays a shared information.
16. The method in accordance with claim 11, wherein each of the first java application and the second java application includes a java application of Xlet format.

17. A method for managing java applications executable in a user device, the method comprising steps of:

   (a) issuing a request for executing a second java application, the request being issued by a first java application;

   (b) switching the first java application to background;

   (c) executing the second java application; and

   (d) killing the second java application when a request for killing the second java application is issued by the second java application and switching the first java application in background to foreground.

18. The method in accordance with claim 17, wherein the first java application generates the request for executing the second java application based on a user input received through a user interface.

19. The method in accordance with claim 17, wherein the request for executing the second java application further includes a path for obtaining the second java application.

20. The method in accordance with claim 19, wherein the step (c) comprises executing the second java application by obtaining the second java application at a location defined
by the path.

21. The method in accordance with claim 17, wherein the first java application and the second java application inter-relays a shared information.

22. The method in accordance with claim 17, wherein each of the first java application and the second java application includes a java application of Xlet format.
ABSTRACT OF DISCLOSURE

The present invention relates to a method for managing java applications executable in a user device. The present invention provides an expandability for and a continuity between java applications by changing states of the java applications in execution and sharing information between the java applications.
Fig. 1
START

issue request for executing 2\textsuperscript{nd} java application including execution type information \textsuperscript{\textapprox} S110

change state of 1\textsuperscript{st} java application according to execution type information \textsuperscript{\textapprox} S130

obtain and execute 2\textsuperscript{nd} java application \textsuperscript{\textapprox} S150

kill 2\textsuperscript{nd} java application and re-change state of 1\textsuperscript{st} java application \textsuperscript{\textapprox} S170

END

Fig. 2
execute child1 according to request of root

execute child2 according to request of child1

Fig. 3
START

S131  execution type information designates 1st execution type?

YES S132
kill 1st java application

NO

S134  execution type information designates 2nd execution type?

YES S135
pause 1st java application

NO

S137  execution type information designates 3rd execution type?

YES S138
switch 1st java application to background

NO

END

Fig. 4
Fig. 5
Fig. 6
Fig. 7
START

issue request for executing 2nd java application

pause 1st java application

obtain and execute 2nd java application

kill 2nd java application and unpause 1st java application

END

Fig. 8
START

issue request for executing 2nd java application \[S310\]

switch 1st java application to background \[S330\]

obtain and execute 2nd java application \[S350\]

kill 2nd java application and switch 1st java application in background to foreground \[S370\]

END

Fig. 9
DECLARATION FOR PATENT APPLICATION

As a below-named inventor(s), I/we hereby declare that:

My/Our residence(s), post office address(es) and citizenship(s) is/are as stated below next to my/our name(s).

I/We believe I/we am/are the original inventor, first and sole (if only one name is listed below) or the original, first and joint inventors (if plural names are listed below) of the subject matter which is claimed, and for which a patent is sought on the invention entitled:

the specification of which: (check one)
[X] is attached hereto.

[ ] was filed on ____________, as Serial No. ________________

and was amended on ________________________________ (if applicable).

I/We hereby state that we have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I/We acknowledge the duty to disclose information which is material to the patentability of this application as defined by 37 CFR § 1.56.

I/We hereby claim foreign priority benefits under 35 U.S.C. § 119 of any foreign application(s) for patent or inventor's certificate listed below, and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Applications:

10-2009-0014161
(Application No.)
(Country) Rep. of Korea
(Day/Month/Year Filed) 20 / 2 / 2009
[X] [ ]

I/We hereby appoint the Practitioners associated with the following Customer Number:

Customer Number 20529

Direct Telephone Calls to:

Gary M. Nath
(703) 548-6284

Send Correspondence to:

Customer Number 20529

THE NATH LAW GROUP
112 South West Street
Alexandria, VA 22314
U.S.A.

I/We hereby claim the benefit under 35 U.S.C. § 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by 35 U.S.C. § 112, first paragraph, I/we acknowledge the duty to disclose material information as defined in 37 CFR § 1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

(U.S. Application Serial No.) (U.S. Filing Date) (Status--patented, pending, abandoned)

(U.S. Application Serial No.) (U.S. Filing Date) (Status--patented, pending, abandoned)
I/we hereby claim the benefit under 35 U.S.C. 119(e) of any United States provisional application(s) listed below:

Application Number(s) ___________________________ Filing Date ___________________________

We hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements are made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001 and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full name of sole or first inventor: BAEK, Wonjang
Inventor's Signature (Wonjang Bae) Date: 12/5/09
Country of Citizenship: Republic of Korea

Full name of second inventor: KIM, John
Inventor's Signature (John Kim) Date: 12/5/09
Residence: SK HUB SU B-1416, 650-4, Guro 1-dong, Guro-gu, Seoul 152-720, Republic of Korea
Country of Citizenship: Republic of Korea
Post Office Address: SK HUB SU B-1416, 650-4, Guro 1-dong, Guro-gu, Seoul 152-720, Republic of Korea
PATENT APPLICATION SERIAL NO.__________________________

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

02/19/2010 AWONDAF1 00000023 12656872
01 FC:2011 165.00 DP
02 FC:2111 270.00 DP
03 FC:2311 110.00 DP
04 FC:2202 52.00 DP

PTO-1556
(5/87)

<table>
<thead>
<tr>
<th>APPLICATION AS FILED – PART I</th>
<th>SMALL ENTITY</th>
<th>OTHER THAN SMALL ENTITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOR</td>
<td>(Column 1)</td>
<td>(Column 2)</td>
</tr>
<tr>
<td>BASIC FEE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(37 CFR 1.16(a), (b), or (c))</td>
<td>NUMBER FILLED</td>
<td>NUMBER EXTRA</td>
</tr>
<tr>
<td>SEARCH FEE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(37 CFR 1.16(k), (l), or (m))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXAMINATION FEE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(37 CFR 1.16(b), (p), or (q))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL CLAIMS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(37 CFR 1.16(i))</td>
<td>22</td>
<td>minus 20 =</td>
</tr>
<tr>
<td>INDEPENDENT CLAIMS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(37 CFR 1.16(h))</td>
<td>3</td>
<td>minus 3 =</td>
</tr>
<tr>
<td>APPLICATION SIZE FEE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(37 CFR 1.16(s))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* If the difference in column 1 is less than zero, enter “0” in column 2.

APPLICATION AS AMENDED – PART II

<table>
<thead>
<tr>
<th>AMENDMENT A</th>
<th>CLAIMS REMAINING AFTER AMENDMENT</th>
<th>HIGHEST NUMBER PREVIOUSLY PAID FOR</th>
<th>PRESENT EXTRA</th>
<th>SMALL ENTITY</th>
<th>ADDITIONAL FEE ($)</th>
<th>OTHER THAN SMALL ENTITY</th>
<th>ADDITIONAL FEE ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total (37 CFR 1.16(i))</td>
<td>Minus **</td>
<td>=</td>
<td>X =</td>
<td>OR</td>
<td>X =</td>
<td>OR</td>
</tr>
<tr>
<td></td>
<td>independent (37 CFR 1.16(h))</td>
<td>Minus ***</td>
<td>=</td>
<td>X =</td>
<td>OR</td>
<td>X =</td>
<td>OR</td>
</tr>
<tr>
<td></td>
<td>Application Size Fee (37 CFR 1.16(s))</td>
<td>N/A</td>
<td></td>
<td></td>
<td>OR</td>
<td>N/A</td>
<td>OR</td>
</tr>
<tr>
<td></td>
<td>FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))</td>
<td></td>
<td></td>
<td>TOTAL ADD'T FEE</td>
<td>OR</td>
<td>TOTAL ADD'T FEE</td>
<td>OR</td>
</tr>
</tbody>
</table>

AMENDMENT B

<table>
<thead>
<tr>
<th>CLAIMS REMAINING AFTER AMENDMENT</th>
<th>HIGHEST NUMBER PREVIOUSLY PAID FOR</th>
<th>PRESENT EXTRA</th>
<th>SMALL ENTITY</th>
<th>ADDITIONAL FEE ($)</th>
<th>OTHER THAN SMALL ENTITY</th>
<th>ADDITIONAL FEE ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total (37 CFR 1.16(i))</td>
<td>Minus **</td>
<td>=</td>
<td>X =</td>
<td>OR</td>
<td>X =</td>
<td>OR</td>
</tr>
<tr>
<td>Independent (37 CFR 1.16(h))</td>
<td>Minus ***</td>
<td>=</td>
<td>X =</td>
<td>OR</td>
<td>X =</td>
<td>OR</td>
</tr>
<tr>
<td>Application Size Fee (37 CFR 1.16(s))</td>
<td>N/A</td>
<td></td>
<td></td>
<td>OR</td>
<td>N/A</td>
<td>OR</td>
</tr>
<tr>
<td>FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))</td>
<td></td>
<td></td>
<td>TOTAL ADD'T FEE</td>
<td>OR</td>
<td>TOTAL ADD'T FEE</td>
<td>OR</td>
</tr>
</tbody>
</table>

* If the entry in column 1 is less than the entry in column 2, write “0” in column 3.
** If the "Highest Number Previously Paid For" in this space is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" in this space is less than 3, enter "3".

This collection of information is required by 37 CFR 1.16. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.