# Method for Viewing Document Information on a Mobile Communication Device

**[0001]** The following is directed in general to displaying content on mobile communication devices, and more particularly to a method for viewing document information about a document, on a mobile communication device, without having to retrieve the full document onto the device.

## Background of the Invention

**[0002]** Mobile communication devices are becoming increasingly popular for business and personal use due to a relatively recent increase in number of services and features that the devices and mobile infrastructures support. Handheld mobile communication devices, sometimes referred to as mobile stations, are essentially portable computers having wireless capability, and come in various forms. These include Personal Digital Assistants (PDAs), cellular phones and smart phones. While their reduced size is an advantage to portability, bandwidth and processing constraints of such devices present challenges to the downloading and viewing of documents, such as word processing documents, tables and images. Also, as a result of their enhanced levels of functionality and computing power, handheld mobile communication devices are increasingly susceptible to attack by computer viruses.

**[0003]** Computer hackers commonly use email attachments as virus carriers to attack corporate network-connected computers. Therefore, email attachments are often identified as presenting a security threat for corporate networks. In order to protect such networks, many corporations and organizations use sophisticated systems to safely handle email attachments. One of the more common corporate approaches is to employ document management systems. One feature of such systems is that they usually rename email attachments with a common extension, for example ".tmp".

**[0004]** When the user of a mobile device receives an email with renamed attachments it is difficult for the user to determine which attachment is of interest based on file names alone. For example, if a mobile device user receives an email with attachments named 0001.tmp, 0002.tmp and 0003.tmp, and only one of them is a MS WORD® document that is of interest, the user is unable to identify the document from the common file extensions. The normal recourse in such a situation is to retrieve the document contents for all

attachments from the remote document server, and successively review the documents in order to identify the desired one.

**[0005]** However, the downloading of an entire document from the server to a mobile communication device consumes a large amount of bandwidth, especially when the document is large. In addition, viewing even a portion of such a downloaded document on the device consumes substantial device CPU/memory/battery resources.

## Summary of the Invention

**[0006]** According to an aspect of the invention, a method is provided for viewing document information on a mobile communication device (e.g. type, creation time, etc), without having to retrieve the full document onto the device. The solution is client - server based. The client is the mobile device attachment viewing application and the server is the document (attachment) handling process on a remote machine. This method includes two operational steps: server document information construction and delivery, and document information display on the mobile device.

**[0007]** By using the method set forth herein, a user is able to identify a document of interest, without retrieving the document content from the server for each attachment in an email. This minimizes bandwidth usage and provides an enhanced on-demand attachment viewing experience. Also, eliminating unnecessary document content transmission to the device minimizes device power consumption.

**[0008]** Additional aspects and advantages will be apparent to a person of ordinary skill in the art, residing in the details of construction and operation as more fully hereinafter described and claimed, reference being had to the accompanying drawings.

## Brief Description of the Drawings

**[0009]** A detailed description of the preferred embodiment is set forth in detail below, with reference to the following drawings, in which:

Figure 1 is a block diagram of a network environment in which the preferred embodiment may be practiced;

Figure 2 is a tree diagram showing the basic structure of a Document Object Model (DOM) used in the preferred embodiment;

Figure 3 shows the top-level of the DOM structure in Figure 2;

Figure 4 shows an exemplary DOM structure for a word processing document;

Figure 5 shows an exemplary DOM structure for a table document;

Figure 6 shows an exemplary DOM structure for a word processing document containing an image subdocument;

Figure 7 is a flowchart showing document information construction and delivery according to the preferred embodiment; and

Figures 8A and 8B show static and dynamic display, respectively, of document information on a mobile communication device according to the preferred embodiment.

**Detailed Description of the Preferred Embodiment**

**[0010]** With reference to Figure 1, network environment 10 is shown in which the preferred embodiment may be practiced. Network environment 10 includes mobile devices 12 communicating via a wireless network 14 to a server 28 for downloading document attachments to the mobile devices 12. While only one server 28 is shown for illustration purposes, a person of skill in the art will understand that network environment 10 could have many such servers for hosting web sites or graphic download sites, providing access to picture files such as JPEG, TIFF, BMP, PNG, SGI, MP4, MOV, GIF, SVG, etc. As would be understood by one of ordinary skill in the art, wireless network 14 may be a GSM/GPRS, CDPD, TDMA, iDEN Mobitex, DataTAC network, or a future network such as EDGE or UMTS, or a broadband network such as Bluetooth and variants of 802.11.

**[0011]** A connection to a fixed service requires special considerations, and may require special permission as authorized through a Network Access Point (NAP) 16. For generic services, such as web access, a proxy-gateway or Network Address Translator (NAT) 18 may be provided so that a network operator can control and bill for the access. NATs 18 enable management of a limited supply of public Internet addresses for large populations of wireless mobile devices. Solutions offered by a proxy-gateway or NAT 18 often involve a complex infrastructure, and thus may be managed by value-added service providers (VASPs), which provide, for instance, WAP gateways, WAP proxy gateway solutions, multi-media messaging servers (MMS) and Internet Multi-Media Services (IMS).

**[0012]** Private Intranet services 26 may require an associated Private Intranet Proxy Gateway 24 for accessing content on server 28. Such private services include WML access to corporate mail systems, HTML access to CRM databases, or any other services that deliver information as formatted data with links and URLs embedded. As shown, it is possible that a private service 26 may be connected directly to the wireless network 14, as opposed to being connected via Internet 20.

**[0013]** Referred to throughout this document, for the purpose of describing the preferred embodiment, is the structure of a Document Object Model (DOM) for a document attachment to be viewed on a mobile device 12.

**[0014]** The attachment server 28 uses a file-parsing distiller in the preferred embodiment, for a specific document type, to build an in-memory Document Object Model (DOM) structure representing an attachment of that document type. The document DOM structure is stored in a memory cache of server 28, and can be iterated bi-directionally.

**[0015]** As shown in Figure 2, the graph-based document DOM structure consists of nodes and leaves. The nodes serve as the parents of leaves and nodes, while leaves are end points of a branch in the graph. Each node and leaf can have a set of attributes to specify its own characteristics. For example, a paragraph node can contain attributes to specify its alignment, style, entry of document TOC, etc. In addition, each of the nodes and the leaves has a unique identifier, called a DOM ID, to identify itself in the document DOM structure.

**[0016]** The document DOM structure is divided into three parts: top-level, component and references. The top level refers to the document root structure, while the main document is constructed in the component and the references represent document references to either internal or external sub-document parts. The following paragraphs examine each part in detail.

**[0017]** The root node of a document DOM structure, referred to as "Document", contains several children nodes, referred to as "Contents", which represent different aspects of the document contents. Each "Contents" node contains one or multiple "Container" nodes used to store various document global attributes. The children of the "Container" nodes are components, which store the document structural and navigational

information. When the attachment server 28 builds the DOM structure for an attachment file for the first time, the top-level structure is a single parent-child chain as shown in Figure 3:

**[0018]** Three types of components are defined by the attachment server 28: text components, table components and image components, which represent text, tables and images in a document, respectively. The text and table components are described in detail below, and the image component structure is identical.

**[0019]** A component consists of a hierarchy of command nodes. Each command represents a physical entity, a property, or a reference defined in a document. For the text component, the physical entity commands are page, section, paragraph, text segments, comments, footnote and endnote commands, which by name define the corresponding entity contained in a document. The property commands for the text component are font, text color, text background color, hyperlink start/end and bookmark commands. The text component has only one reference command, referred to as the text reference command, which is used to reference a subdocument defined in the main body of a document. Usually, the children of a text component are page or section command nodes that, in turn, comprise a set of paragraph command nodes. The paragraph command can contain one or multiple nodes for the remaining command types.

**[0020]** Using the following sample text document, the corresponding document DOM structure is shown in Figure 4:

| |
|---|
| First paragraph.<br>Second paragraph with **bold** and red text. |

**[0021]** As Figure 4 demonstrates, the section command, which is the child of the text component, consists of two paragraph commands. The first paragraph command contains one text segment command and the text content for that paragraph is added as an attribute to the text segment command. The second paragraph command has a relatively more complex structure, as the text properties in the paragraph are much richer. Each time a text property (font, text color, etc) changes, a corresponding text property command is created and the change value is added to that command as an attribute. The subsequent text segment command records the text with the same text property as an attribute. As document structure gets richer and more complex, more commands of corresponding types

are created and the document properties are added as attributes to those commands.

**[0022]** The table component has the same three types of commands as the text component, but different command names. The document DOM structure for the sample table document below is shown in Figure 5:

| | |
|---|---|
| Cell One | Cell Two |
| Cell Three | Cell Four |

**[0023]** As shown in the Figure 5, the table component has physical entity type commands of table, tablerow and tablecell, where the tablecell command can contain all available commands for the text component. In the example above, the first child TableRow command of the table command has an attribute "Index" defined by value of 0. This indicates that the indicated table row is the first one defined in the table. The attribute of the leftmost table cell command in Figure 5 has the same meaning.

**[0024]** A document sometimes contains subdocuments, for example images, tables, text boxes etc. The DOM structure set forth herein uses a reference command to point to the graph of such subdocuments. Thus, for the following sample document, the attachment server 28 generates the DOM structure shown in Figure 6:

This document has subdocument of images like this one .

Second paragraph contains the same image .

**[0025]** The structure shown in Figure 6 is identical to that discussed above in connection with Figures 4 and 5, except for the attributes of the two reference commands. The attachment server 28 constructs the image in "Sample Three" as a separate image component, which contains all of the image data in its own DOM hierarchy. In the DOM structure for the main document, the values of the "Ref" attributes of those two reference commands point to the image component, as indicated by the dashed lines, such that the DOM structure connects together all parts of the document.

**[0026]** Having described the document DOM structure used to implement an embodiment of the invention, a detailed discussion will now be provided of the document information construction, delivery and display function or method according to the preferred embodiment.

**[0027]** With reference to Figure 7, after receiving an email with renamed attachments on a mobile device 12 (step 30), the user can send a request to the server 28 for the associated document information. Once the server receives such a request, it initially constructs only the top level of the document DOM structure for the attachment (step 32), as discussed above in connection with Figure 3. Construction of the top level of the document DOM structure is a very fast operation, thereby minimizing wait time for the user. The server 28 then examines the document binary data (step 34) to find the basic document information (i.e. type, author, creation time and date, modified time and date, format type, etc) for the document.

**[0028]** Specifically, the file is opened in binary mode and searched to locate a file signature. The signature of the file is stored either at the beginning or at the end of a file (usually the first or last tens to a few hundred of bytes), and is used to indicate identify the file type (i.e. document original format type, discussed in greater below). For example, the signature of the PDF document original format type, "%PDF", is contained in the first 4 bytes of the raw binary data of a PDF file. For other types of information, the binary file must be searched further.

**[0029]** Or, since MS Office® files are "storage" type files (rather than "stream" type files such as PDF and text file), which can contain sub-streams and sub-storage, the first 8 bytes need to be a fixed value. Therefore, after confirming that the file is "storage" type, the server 28 searches for a stream called "WordDocument" contained in the file to verify that a file is a MS Word® file.

**[0030]** Directly examining binary data ensures that any macro or operation in the attachment is not executed, thereby eliminating any chance of virus attacks and/or other security threats.

**[0031]** After retrieving all of the available document information stored in the file, the server 28 adds the retrieved information as attributes to the root component of the DOM structure (step 36).

**[0032]** It will be appreciated that, compared to retrieval of the entire document contents, the document information search and construction process of Figure 7 is much simpler and quicker, especially for large documents, since the server 28 usually does not have to parse deep into the file to locate the document information. After the document

information is constructed, the server 28 sends a response back to the client device 12 (step 38) over a standard transportation channel.

**[0033]** After the client device 12 receives the requested document information for an email attachment, it displays the information to the user according to type. Specifically, server 28 indicates the document original format type in five categories: Archives, Documents, Spreadsheets, Presentations and Images. These five categories are each represented by a unique icon displayed on the screen of the mobile device 12 which, according to the preferred embodiment, are as follows:

| Icon | Attachment Type | Supported Sub document types |
|---|---|---|
|  | Archives | ZIP Archives. Only used in Attachment List Screen |
|  | Documents | MS Word, Adobe PDF, Corel WordPerfect, ASCII Text, HTML |
|  | Spreadsheets | MS Excel |
|  | Presentations | MS PowerPoint |
|  | Images | BMP, PNG, GIF, TIFF, JPEG |

**[0034]** The server 28 also preferably sends the document format subtype to the mobile device 12. For example, MS Word® and Adobe® PDF are both categorized as type "documents", which is further specified by the server 28 using the different subtypes, as indicated above. In addition to the document type, other document information, such as size, creation time, last modified time and author are also sent to the client device 12.

**[0035]** The client device 12 displays the information in a static or dynamic fashion. With static display, the client device 12 displays the information on a static area, (e.g. title bar, etc.) of the device screen. With dynamic display, the client device 12 first caches the document information and then displays it using dynamic GUI elements, (e.g. pop-up message box, etc.) in response to a query from the user.

**[0036]** Figure 8A shows a static display of document type whereas Figure 8B shows both the static display of document type and a pop-up message box for dynamic representation of the remaining document information sent from the server 28.

**[0037]** In summary, the method of document information delivery and display according to the preferred embodiment allows a mobile device user to quickly determine if an attachment is of interest without having to retrieve the document content itself, thereby minimizing overall network bandwidth.

**[0038]** A person skilled in the art, having read this description of the preferred embodiment, may conceive of variations and alternative embodiments. All such variations and alternative embodiments are believed to be within the ambit of the claims appended hereto.

**Claims:**

1. A process for viewing document information on a mobile communication device relating to a document stored on a server, without downloading said document from the server to the mobile device, comprising:

transmitting a request for said document information from said mobile device to said server;

constructing a graph structure within said server, said graph structure containing said document information;

transmitting said document information to said mobile device; and

receiving and displaying said document information on said mobile device.

2. The process of claim 1, wherein said constructing further comprises creating a root node, and at least one child node for storing contents of said document, examining said document for said document information, retrieving said document information and adding said document information to said root node.

3. The process of claim 2, wherein said examining further comprises searching document binary data of said document to find said document information.

4. The process of claim 2 or claim 3, wherein said document information is added to said root node as an attribute.

5. The process of any one of claims 1 to 4, wherein said graph structure is a Document Object Model (DOM).

6. The process of any one of claims 1 to 5, wherein said displaying further comprises presenting said document information on a static screen area of said mobile device.

7. The process of claim 6, wherein said static screen area is a title bar.

8. The process of any one of claims 1 to 5, wherein said displaying further comprises caching and then presenting said document information using GUI elements on a dynamic screen area of said mobile device.

9. The process of claim 8, wherein said GUI elements include a pop-up message box.

10. A server process comprising:

constructing a graph structure of a document, said graph structure containing document information; and

transmitting said document information.

11. The server process of claim 10, wherein said constructing further comprises creating a root node, and at least one child node for storing contents of said document, examining said document for said document information, retrieving said document information and adding said document information to said root node.

12. The server process of claim 11, wherein said examining further comprises searching document binary data of said document to find said document information.

13. The server process of claim 11 or claim 12, wherein said document information is added to said root node as an attribute.

14. The server process of any one of claims 10 to 13, wherein said graph structure is a Document Object Model (DOM).

15. A mobile device process comprising:

transmitting a request for document information; and

receiving and statically or dynamically displaying said document information on said mobile device.

16. The mobile device process of claim 15, wherein said statically displaying further comprises presenting said document information on a static screen area of said mobile device where said static screen area is a title bar.

17. The mobile device process of claim 15, wherein said dynamically displaying further comprises caching and then presenting said document information using GUI elements on a dynamic screen area of said mobile device.

18. The mobile device process of claim 17, wherein said GUI elements include a pop-up message box.

19. The process according to any one of claims 1 to 19, wherein said document information includes any of (i) document original format type including any one or more of Archives, Documents, Spreadsheets, Presentations and Images represented on said mobile device by respective icons, (ii) format subtype for identifying a unique software application, and (iii) other document information including at least one of size, creation time, last modified time and author.

20. A system for viewing document information on a mobile communication device relating to a document stored on a server, without downloading said document from the server to the mobile device, said system comprising:

means for transmitting a request for said document information from said mobile device to said server;

means for constructing a graph structure within said server, said graph structure containing said document information;

means for transmitting said document information to said mobile device; and

means for receiving and displaying said document information on said mobile device.

21. A server comprising:

means for constructing a graph structure of a document, said graph structure containing document information; and

means for transmitting said document information.

22. A mobile device comprising:

means for transmitting a request for document information; and

means for receiving and statically or dynamically displaying said document information on said mobile device.