Technical Disclosure Commons

Defensive Publications Series

December 2022

RESOLVING SIZING ISSUES IN EXPORTING LARGE OBJECTS

Lei Liu

Follow this and additional works at: https://www.tdcommons.org/dpubs_series

Recommended Citation
Liu, Lei, "RESOLVING SIZING ISSUES IN EXPORTING LARGE OBJECTS", Technical Disclosure Commons, (December 28, 2022)
https://www.tdcommons.org/dpubs_series/5610

This work is licensed under a Creative Commons Attribution 4.0 License.
This Article is brought to you for free and open access by Technical Disclosure Commons. It has been accepted for inclusion in Defensive Publications Series by an authorized administrator of Technical Disclosure Commons.

RESOLVING SIZING ISSUES IN EXPORTING LARGE OBJECTS

ABSTRACT

A computing device (e.g., a smartphone, a smartwatch, smartglasses, smart headphones (including earbuds), a laptop computer, a tablet computer, a vehicle head unit, etc.) may execute an application programming interface (API) that outputs pairing information per a universal or general standard. For example, the API may process communication parameters uniquely specified by an original equipment manufacturer (OEM) to identify key attributes and features. The API may insert the identified key attributes and features into a custom specification having a common format. In this way, the API may generalize, abstract, or otherwise standardize communication parameters specifications that were originally uniquely defined. The custom specification may be compatible with various types of exporting, such as incremental exporting, block exporting, etc. These various types of exporting may reduce or eliminate sizing issues encountered when a computing device exports large amounts of data (e.g., objects).

DESCRIPTION

FIG. 1 below is a conceptual diagram illustrating a system 10 that includes a computing device 100 and a computing system 120. Examples of computing device 100 may include a cellular phone, a smartphone, a personal digital assistant (PDA), a laptop computer, a tablet computer, a portable gaming device, a portable media player, an e-book reader, a watch (including a so-called smartwatch), smart glasses, a gaming controller, a vehicle head unit, etc. Examples of computing system 120 may include one or more desktop computers, laptop computers, mainframes, servers, cloud computing systems, virtual machines, etc.

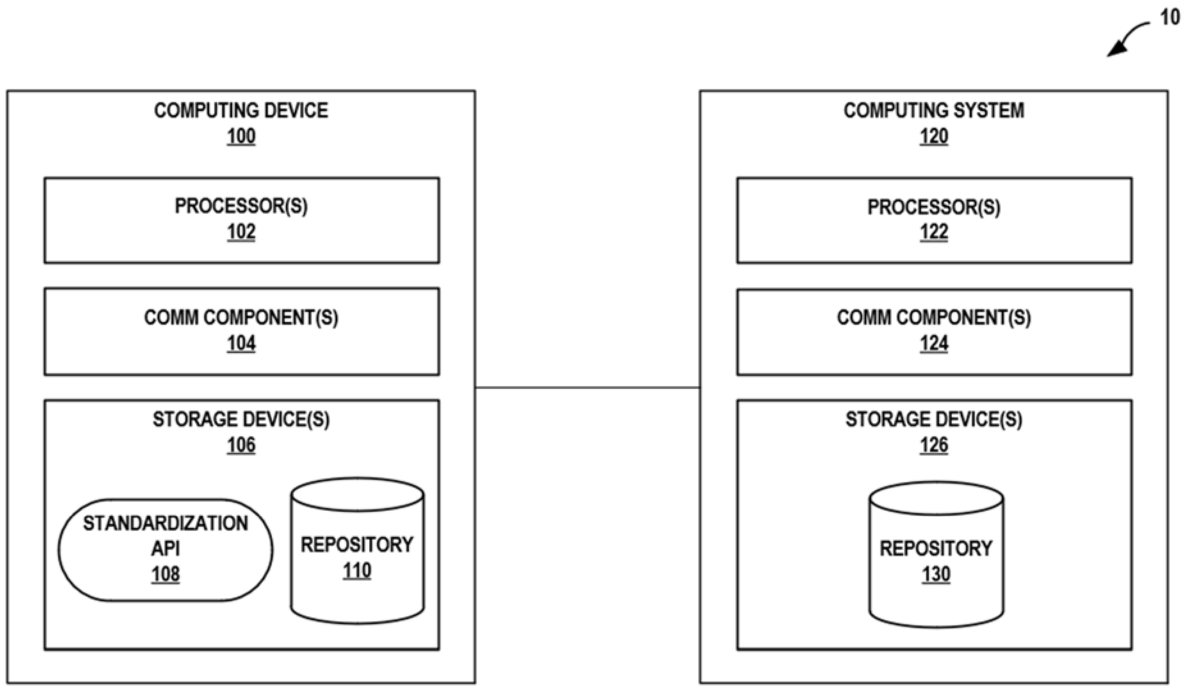

FIG. 1

Computing device 100 may include processors 102 configured to implement functionality and/or execute instructions associated with computing device 100. Examples of processors 102 may include one or more of an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), an application processor, a display controller, an auxiliary processor, a central processing unit (CPU), a graphics processing unit (GPU), one or more sensor hubs, and any other hardware configure to function as a processor, a processing unit, or a processing device.

Computing device 100 may include communication components 104 ("COMM components 104") configured to receive and transmit various types of information over a network. Examples of COMM components 104 may include an ultra-wideband (UWB) radio, a cellular radio, a third-generation (3G) radio, a fourth-generation (4G) radio, a fifth-generation (5G) radio, a Bluetooth® radio (or any other personal area network (PAN) radio), a near-field communication (NFC) radio, a WiFi® radio (or any other wireless local area network (WLAN) radio), etc.

Computing device 100 and may include storage devices 106 configured to store one or more computer-readable storage media. Storage devices 106 may be configured for long-term, as well as short-term storage of information, such as instructions, data, or other information used by computing device 100. In some examples, storage devices 106 may include non-volatile storage elements. Examples of such non-volatile storage elements include magnetic hard discs, optical discs, solid state discs, and/or the like. In other examples, in place of, or in addition to the non-volatile storage elements, the storage devices may include one or more so-called “temporary” memory devices, meaning that a primary purpose of these devices may not be long-term data storage. For example, the devices may comprise volatile memory devices, meaning that the devices may not maintain stored contents when the devices are not receiving power. Examples of volatile memory devices include random-access memories (RAM), dynamic random-access memories (DRAM), static random-access memories (SRAM), etc.

Computing system 120 may include one or more components substantially similar to the components of computing device 100. For example, computing system 120 may include processors 122, COMM components 124, and storage devices 126 configured to perform the techniques of this disclosure.

Two or more computing devices (e.g., computing device 100 and another computing device) may connect (e.g., pair) to communicate with each other, such as via a personal area network (PAN). When attempting to pair, the computing devices may transmit and receive device pairing information, such as communication parameters (e.g., device identity, device type, device properties, etc.), necessary for establishing (e.g., negotiating) communication. In general, each computing device may have communication parameters uniquely specified by the original equipment manufacturer (OEM) of the computing device. In other words, the communication parameters specification for a particular computing device may not comply with a universal or general standard. Additionally, the communication parameters specification may include a considerable amount of data. As a result, a computing device may experience sizing issues when, for example, exporting (e.g., to computing system 120) the communication parameters specifications of the various computing devices with which the computing device has paired.

In accordance with techniques of this disclosure, computing device 100 may execute a standardization application programming interface 108 ("standardization API 108") that outputs pairing information per a universal or general standard ("standard"). For example, standardization API 108 may process a uniquely defined communication parameters specification stored in a repository 110 to identify key attributes and features of the communication parameters specification. Standardization API 108 may insert (in a manner compliant with the standard) the identified key attributes and features into a custom specification having a common format. In this way, standardization API may generalize, abstract, or otherwise standardize communication parameters specifications that were originally uniquely defined. Computing device 100 may then export the custom specification to computing system 120, and computing system 120 may store the custom specification in a repository 130.

An advantage of standardizing the communication parameters specification as described above may include enabling various types of exporting, such as incremental exporting, block exporting, etc. These various types of exporting may reduce or eliminate the sizing issues encountered when computing device 100 exports device pairing information to computing system 120.

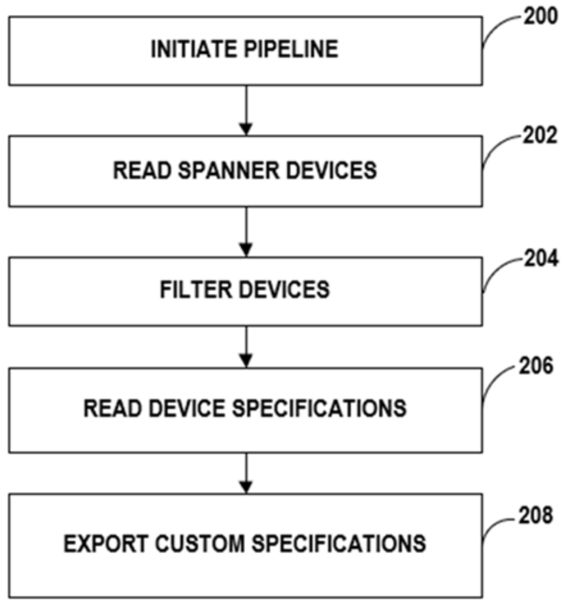

FIG. 2

FIG. 2 is a flowchart of example operations performed by computing device 100 shown in FIG. 1. As shown in FIG. 2, computing device 100 may initiate a pipeline (200). For example, computing device 100 may call an open-source distributed data collection service for transferring data from source to destination (e.g., a Flume pipeline). System 10 may apply a minimal information model to nearby pipelines with newly created, dedicated, and environment-specific access database file groups, network configurations, naming conventions, and public key exchange protocols. For example, no existing nearby pipeline practices, configurations, jobs, resources, and information may be shared or leaked to a Secure Shell File Transfer Protocol (SFTP) service or an OEM.

Computing device 100 may read all the spanning devices (202). Computing device 100 may filter the spanning devices based on, for example, an allowed and filtered list (e.g., selecting or reading all device identifiers) as well as all or specific locales (204). Computing device 100 may read the communication parameters specifications for the allowed devices (206).

Standardization API 110 may standardize the communication parameters specifications to generate corresponding custom specifications. In some examples, standardization API 110 outputs the custom specifications as capacitor (or other database storage format) files for internal support and tracking and/or text proto files on a client connection manager export format (e.g. file formats). Computing device 100 may then export the custom specifications to computing system 120 (208). System 10 may redact private keys (e.g., a device identifier) during exporting for security purposes.

It is noted that the techniques of this disclosure may be combined with any other suitable technique or combination of techniques. As one example, the techniques of this disclosure may be combined with the techniques described in U.S. Patent Application Publication No. 2018/0255445A1. In another example, the techniques of this disclosure may be combined with the techniques described in U.S. Patent Application Publication No. 2013/0217336A1. In yet another example, the techniques of this disclosure may be combined with the techniques described in U.S. Patent Application Publication No. 2020/0265202A1.