



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
90/011,492	02/15/2011	5,966,702	13557.112021	8223

25226 7590 06/06/2011
MORRISON & FOERSTER LLP
755 PAGE MILL RD
PALO ALTO, CA 94304-1018

EXAMINER

ART UNIT PAPER NUMBER

DATE MAILED: 06/06/2011

Please find below and/or attached an Office communication concerning this application or proceeding.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

DO NOT USE IN PALM PRINTER

(THIRD PARTY REQUESTER'S CORRESPONDENCE ADDRESS)

KING & SPALDING LLP

1180 Peachtree Street

Atlanta, Georgia 30309

MAILED

JUN 06 2011

CENTRAL REEXAMINATION UNIT

EX PARTE REEXAMINATION COMMUNICATION TRANSMITTAL FORM

REEXAMINATION CONTROL NO. 90/011,492.

PATENT NO. 5,966,702.

ART UNIT 3992.

Enclosed is a copy of the latest communication from the United States Patent and Trademark Office in the above identified *ex parte* reexamination proceeding (37 CFR 1.550(f)).

Where this copy is supplied after the reply by requester, 37 CFR 1.535, or the time for filing a reply has passed, no submission on behalf of the *ex parte* reexamination requester will be acknowledged or considered (37 CFR 1.550(g)).

Office Action in Ex Parte Reexamination	Control No. 90/011,492	Patent Under Reexamination 5,966,702	
	Examiner MARY STEELMAN	Art Unit 3992	

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

- a ☒ Responsive to the communication(s) filed on 28 April 2011. b ☐ This action is made FINAL.
c ☒ A statement under 37 CFR 1.530 has not been received from the patent owner.

A shortened statutory period for response to this action is set to expire 2 month(s) from the mailing date of this letter. Failure to respond within the period for response will result in termination of the proceeding and issuance of an *ex parte* reexamination certificate in accordance with this action. 37 CFR 1.550(d). **EXTENSIONS OF TIME ARE GOVERNED BY 37 CFR 1.550(c).** If the period for response specified above is less than thirty (30) days, a response within the statutory minimum of thirty (30) days will be considered timely.

Part I THE FOLLOWING ATTACHMENT(S) ARE PART OF THIS ACTION:

- | | |
|---|---|
| 1. <input type="checkbox"/> Notice of References Cited by Examiner, PTO-892. | 3. <input type="checkbox"/> Interview Summary, PTO-474. |
| 2. <input checked="" type="checkbox"/> Information Disclosure Statement, PTO/SB/08. | 4. <input type="checkbox"/> _____. |

Part II SUMMARY OF ACTION

- 1a. ☒ Claims 1,5-7,11-13,15 and 16 are subject to reexamination.
1b. ☒ Claims 2-4,8-10,14 and 17-23 are not subject to reexamination.
2. ☐ Claims _____ have been canceled in the present reexamination proceeding.
3. ☐ Claims _____ are patentable and/or confirmed.
4. ☒ Claims 1,5-7,11-13,15 and 16 are rejected.
5. ☐ Claims _____ are objected to.
6. ☐ The drawings, filed on _____ are acceptable.
7. ☐ The proposed drawing correction, filed on _____ has been (7a) ☐ approved (7b) ☐ disapproved.
8. ☐ Acknowledgment is made of the priority claim under 35 U.S.C. § 119(a)-(d) or (f).
 a) ☐ All b) ☐ Some* c) ☐ None of the certified copies have
 1 ☐ been received.
 2 ☐ not been received.
 3 ☐ been filed in Application No. _____.
 4 ☐ been filed in reexamination Control No. _____.
 5 ☐ been received by the International Bureau in PCT application No. _____.
 * See the attached detailed Office action for a list of the certified copies not received.
9. ☐ Since the proceeding appears to be in condition for issuance of an *ex parte* reexamination certificate 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.
10. ☐ Other: _____

cc: Requester (if third party requester)

DETAILED ACTION

Reexamination

Claims 1, 5-7, 11-13, 15, and 16 of USPN 5,966,702 to Fresko et al. (hereinafter '702) (file date 10/31/1997, issue date 10/12/1999) are under reexamination. The Reexamination control number is 90/011,492.

Notice Regarding Certain Reexamination Issues

The patent owner is reminded of the continuing responsibility under 37 CFR 1.565(a), to apprise the Office of any litigation activity, or other prior or concurrent proceeding, involving this patent under reexamination throughout the course of this reexamination proceeding. The third party requester is also reminded of the ability to similarly apprise the Office of any such activity or proceeding throughout the course of this reexamination proceeding. See MPEP §§ 2207, 2282 and 2286.

Extensions of time under 37 CFR 1.136(a) will not be permitted in these proceedings because the provisions of 37 CFR 1.136 apply only to "a Patent Applicant" and not to parties in a reexamination proceeding. Additionally, 35 U.S.C. 305 requires that reexamination proceedings "will be conducted with special dispatch" (37 CFR 1.550(a)). Extension of time in ex parte reexamination proceedings are provided for in 37 CFR 1.550(c).

Information Disclosure Statement

IDS received 04/28/2011 has been considered. Citations lacking dates are lined through. Court proceedings have been considered. However, the citations to court proceedings do not meet the

Art Unit: 3992

requirements of 37 CFR 1.98 and have been lined through. Where patents, publications, and other such items of information are submitted by a party (patent owner or requester) in compliance with the requirements of the rules, the requisite degree of consideration to be given to such information will be normally limited by the degree to which the party filing the information citation has explained the content and relevance of the information. The initials of the examiner placed adjacent to the citations on the form PTO /SB /08A and 08B or its equivalent, without an indication to the contrary in the record, do not signify that the information has been considered by the examiner any further than to the extent noted above. See MPEP 2256.

Prior Art References

USPN 5,815,718 to Tock (file date 05/30/1996, issue date 09/29/1998, hereinafter, the "Tock patent" or '718, qualifies as a 102(e) reference.

USPN 5,613,120 to Palay, et al., (file date 10/20/1994, issue date 03/18/1997, hereinafter, the "Palay patent" or '120), qualifies as a 102(a) reference.

Claim Rejections - 35 USC § 102

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 –

Art Unit: 3992

(a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Rejections

Claims 1, 5-7, 11-13, 15, and 16 are rejected under 35 U.S.C. 102(e) as being anticipated by USPN 5,815,718 to Tock. See Request (02/15/2011) at pp. 10-11, 13, and Exhibit 4.

Per **claim 1**, Tock discloses (1: 41-47) a method of pre-processing class files: "...an offline class loader that is used to produce an executable module whose classes are preloaded into memory without requiring runtime dynamic loading...the offline class loader modifies (pre-processing) the existing class structures to accommodate static loading."

Tock discloses (FIG. 10, 8: 40-46) determining a plurality of duplicated elements in a plurality of class files: "...offline class loader proceeds to eliminate duplicate constants (determining plurality of duplicated elements)...For each class file (plurality of class files) (step 806), each entry in the class' constant pool is scanned for duplicate constants (step 812). Referring to FIG. 10, duplicate constants are detected by using a hash table."

Tock discloses (8: 45-55) forming a shared table comprising said plurality of duplicated elements: "Referring to FIG. 10, duplicate constants are detected by using a hash table (shared table). The hash value of the constant is determined by an appropriate hashing function (step 1102). A check is made to determine whether the hash value is contained in the hash table (step 1104). If the hash value exists in the hash table, then the constant is a duplicate and the entry is deleted from the constant pool by altering the constant's entry in the map table to reflect the memory location of the existing constant (step 1106). Otherwise, the constant's hash value and memory location are stored in the hash table (step 1108)."

Tock discloses (5: 38-50; 5: 29-34) removing said duplicated elements from said plurality of class files to obtain a plurality of reduced class files: ""The output of the offline class loader 302 can consist of two files: a constant pool file containing the constant data for the entire application; and an updated classfile (for each of said plurality of class files, class file is reduced by removing duplicates and representing as a single instance in constant pool file) containing the

Art Unit: 3992

class data structures and class members..." "The offline class loader also performs a number of optimizations in order to produce a more compact representation of the executable code (duplicated elements removed). For example, the constant pool that is associated with each class is combined for all the classes residing in the application."

Tock discloses (FIG. 8B; 10: 29-32; 5: 38-50) a multi-class file comprising said plurality of reduced class files. Tock's class loader can output the constant pool file and updated class file as a single file. "Lastly, the offline class loader outputs the universal constant pool, an updated classfile containing the class data structures and the indicators specifying the memory storage requirements, as well as a special boot time indicator (step 930)." "The output of the offline class loader 302 can consist of two files: a constant pool file containing the constant data for the entire application; and an updated classfile containing the class data structures and class members. The data in both of these files is formatted as data definitions, where each definition specifies a bytecode and an offset indicating a memory location. The updated class file will include the memory storage indicators which will indicate in which type of memory storage device a particular set of bytecodes is to reside. However, the method and system described herein is not limited to producing these two files. Other file configuration can be used including, but not limited to, a single file containing all the related class data."

Per **claims 5 and 6**, Tock discloses (8: 40-46; 9: 27-29; FIG. 10) determining one or more constants shared between two or more class files and forming a shared constant table comprising said one or more constants shared between said two or more class files. Tock describes scanning

Art Unit: 3992

each class's constant pool for duplicate constants to be eliminated and merging the constants into a universal constant pool. "Next, the offline class loader proceeds to eliminate duplicate constants. This is performed in order to combine the constant pools of all the classes in a space efficient manner. For each class file (step 806), each entry in the class' constant pool is scanned for duplicate constants (step 812). Referring to FIG. 10, duplicate constants are detected by using a hash table." "Once space is allocated for the universal constant pool, each entry from the various class constant pools is merged into the universal constant pool (step 902)."

Per **claim 7**, Tock discloses (1: 4-7) a computer program product: "The present invention relates generally to object-oriented computer systems having classes that are dynamically loaded at runtime, and particularly to a system and method for a subset of the classes in a read-only memory."

Tock discloses (1: 41-47; 4: 6-34) a computer usable medium having computer readable program code embodied therein for pre-processing class files, using an offline class loader: "In summary, this disclosure pertains to an offline class loader that is used to produce an executable module whose classes are preloaded into memory without requiring runtime dynamic loading. The executable module, nevertheless, contains a class structure that is tailored for runtime dynamic loading. Thus, the offline class loader modifies the existing class structures to accommodate static loading." "Referring to FIG. 1, a server computer typically includes one or more processors 112, a communications interface 116, a user interface 114, and memory 110. Memory 110 stores...an operating system 118...an Internet communications manager program or

Art Unit: 3992

other type of network access procedures 120... a compiler 122... a source code repository 124 ... a class file repository 128... one or more class libraries 131 containing class files... an offline class loader 132 ... an assembler 134 which produces an object file representing the class members, class data structures, and memory storage indicators in a format that is recognizable for the linker.. a linker 136 for determining the memory layout for a set of preloaded classes and for resolving all symbolic references.. a browser 138...and one or more data files 146 for use by the server." (emphasis added) See limitations further addressed in claim 1 above.

The limitations of **claims 11 and 12** are addressed in claims 5 and 6 above, respectively.

Per **claim 13**, Tock's description (FIG. 1; 4: 6-34) of a server computer includes an apparatus, including a processor, memory, class files stored in memory, processes executing on said processor and a shared table, as addressed in claim 1 above. See Tock FIG. 8B; 5: 29-50; 8: 40-46; 10: 29-32.

Per **claim 15**, Tock discloses (8: 40-46; 10: 29-32) that the duplicated elements comprise elements of constant pools of respective class files, said shared table comprising a shared constant pool. Tock describes eliminating duplicate constants and combining constant pools into a universal constant pool. "Next, the offline class loader proceeds to eliminate duplicate constants. This is performed in order to combine the constant pools of all the classes in a space efficient manner. For each class file (step 806), each entry in the class' constant pool is scanned for duplicate constants (step 812). Referring to FIG. 10, duplicate constants are detected by

Art Unit: 3992

using a hash table." "Lastly, the offline class loader outputs the universal constant pool, an updated class file containing the class data structures and the indicators specifying the memory storage requirements, as well as a special boot time indicator (step 930)."

Per **claim 16**, Tock discloses (FIG. 12; 3: 46-52; 10; 29-43) a virtual machine having a class loader and a runtime data area, said class loader configured to obtain and load said multi-class file into said runtime data area. Tock describes a boot time initiator that loads a multi-class file into the runtime data area. "Lastly, the offline class loader outputs the universal constant pool, an updated class file containing the class data structures and the indicators specifying the memory storage requirements, as well a special boot time initiator (step 930). Referring to FIG. 12, the preloadable executable module and boot time initiator 1220 are permanently stored in the read-only memory of a client computer. Each time the client computer is powered on or rebooted, the boot time initiator 1220 is automatically executed. Among other tasks, the boot time initiator copies all methods and data that must be resident in random access memory during execution to the random access memory locations assigned to them by the linker...executing a Java browser (virtual machine)."

Claims 1 5-7, 11-13, and 15 are rejected under 35 U.S.C. 102(a) as being anticipated by USPN 5,613,120 to Palay et al. See Request (02/15/2011) at pp. 11-12, 13, and Exhibit 5.

Art Unit: 3992

Per **claim 1**, Palay discloses (1: 11-14) a method of pre-processing class files by way of compiling and linking class files of object-oriented computer programs: "The present invention relates generally to software compilers and linkers, and more particularly to software compilers and linkers for compiling object-oriented computer programs."

Palay discloses (FIG. 6; 28: 39-61) determining a plurality of duplicated elements in a plurality of class files, identifying and removing duplicate class definitions and duplicate class symbol entries. "Two issues that must be addressed when doing this merge operation are what to do about duplicate members and the mapping of references from the relocations to the merged class instance table and the merged class symbol table. When merging the class definition tables 414, duplicate non-dynamic and internal dynamic class definitions are removed. Duplicate definitions of dynamic classes are considered to be an error. When merging class instance tables 416, duplicate entries are considered to be an error. Duplicate entries in the class symbol tables 408 are removed. There are no duplicate entries in the class relocation tables 412."

Palay discloses (28: 39-61) forming a shared table comprising said plurality of duplicated elements and creating merged tables as part of merging class information. "In step 606, the linker 112 reads in the object files 106, 108 and the shared libraries 110 and merges together the class information contained in these files." "Two issues that must be addressed when doing this

Art Unit: 3992

merge operation are what to do about duplicate members and the mapping of references from the relocations to the merged class instance table and the merged class symbol table. When merging the class definition tables 414, duplicate non-dynamic and internal dynamic class definitions are removed. Duplicate definitions of dynamic classes are considered to be an error. When merging class instance tables 416, duplicate entries are considered to be an error. Duplicate entries in the class symbol tables 408 are removed. There are no duplicate entries in the class relocation tables 412.”

Palay discloses (28: 54-56) removing said duplicated elements from said plurality of class files to obtain a plurality of reduced class files and identifying and removing duplicate class definitions and duplicate class symbol entries. “When merging the class definition tables 414, duplicate non-dynamic and internal dynamic class definitions are removed.”

Palay discloses (12: 11-14) a compiler that generates information about classes (plurality of class files) and how they are used. Such information is called information and is embedded in the object file produced by the compiler.” Abstract, “A linker links the object file (plurality of object files) potentially with at least one other object file or shared library to thereby generate an executable file or shared library.” Further actions performed by the linker, i.e., removing duplicates, necessarily results in reduced class files.

Art Unit: 3992

Palay discloses (7: 51-54) a multi-class file comprising said plurality of reduced class files and said shared table. Palay's linker creates a shared library in the form of an object file from the merged class information: "[T]he static linker 114 generates another shared library. As will be appreciated, a shared library is in the form of an object file." Also see FIG. 4 and Abstract.

Regarding **claims 5 and 6**, Palay discloses (28: 54-56) determining one or more constants shared between two or more class files and forming a shared constant table comprising one or more constants shared between two or more class files: Palay identifies duplicates among the shared data so that they can be removed. "When merging the class definition tables 414, duplicate non-dynamic and internal dynamic class definitions (constants) are removed."

Regarding **claim 7**, Palay discloses (33: 8-11; 1: 11-14) a computer program product / computer usable medium having computer readable program code embodied therein for pre-processing class files. The compiling and linking of object-oriented computer programs described in Palay involves pre-processing class files. "21. A computer program product comprising a computer readable medium having computer program logic recorded thereon for enabling a computer system to compile and link a source file, said computer program product comprising..." "The present invention relates generally to software compilers and linkers, and more particularly to software compilers and linkers for compiling object-oriented computer programs."

Art Unit: 3992

Palay discloses (Abstract; 28: 50-61) computer readable program code configured to cause a computer to determine a plurality of duplicated elements in a plurality of class files. Palay describes identifying and removing duplicate class definitions and duplicate class symbol entries.

“A system and method for compiling and linking a source file is described. A compiler generates class information pertaining to object-oriented classes referenced in the source file. The class information is sufficient to enable a linker to resolve class definitions and to perform class relocation operations. The compiler also generates an object file from the source file. The object file includes the class information. The compiler generates the object file such that resolution of class definitions and performance of class relocation operations are delayed until operation of the linker. A linker links the object file potentially with at least one other object file or shared library to thereby generate an executable file or shared library. The linker uses the class information contained in the object file to resolve class definitions and to perform class relocation operations.” “Two issues that must be addressed when doing this merge operation are what to do about duplicate members and the mapping of references from the relocations to the merged class instance table and the merged class symbol table. When merging the class definition tables 414, duplicate non-dynamic and internal dynamic class definitions are removed. Duplicate definitions of dynamic classes are considered to be an error. When merging class instance tables 416, duplicate entries are considered to be an error. Duplicate entries in the class symbol tables 408 are removed. There are no duplicate entries in the class relocation tables 412.” See additional limitations addressed in the rejection of claim 1 above.

Art Unit: 3992

Claims 11 and 12 are computer program product versions of claims 5 and 6 above. See limitations addressed above.

Regarding **claim 13**, Palay discloses (FIG. 2) an apparatus comprising a processor and memory coupled to said processor. Palay discloses (6: 14-15; FIG. 4) a plurality of class files stored in said memory. The object files comprise class files. "A linker links the object file with other object files and shared libraries."

Palay discloses (7: 51-54; FIG. 4) a process executing on said processor said process configured to form a multi-class file. Palay's linker creates a shared library in the form of an object file from the merged class information. "[T]he static linker 114 generates another shared library. As will be appreciated, a shared library is in the form of an object file."

Palay discloses (28: 54-56) a plurality of reduced class files obtained from said plurality of class files by removing one or more elements that are duplicated between two or more of said plurality of class files. Palay describes identifying and removing duplicate class definitions and duplicate class symbol entries. "When merging the class definition tables 414, duplicate non-dynamic and internal dynamic class definitions are removed." The merging of class information and the removal of duplicates as expressly described in Palay necessarily would provide reduced class files.

Palay discloses (28: 50-61) a shared table comprising said duplicated elements. Palay describes creating merged tables as part of merging class information. "Two issues that must be addressed when doing this merge operation are what to do about duplicate members and the mapping of

Art Unit: 3992

references from the relocations to the merged class instance table and the merged class symbol table. When merging the class definition tables 414, duplicate non-dynamic and internal dynamic class definitions are removed. Duplicate definitions of dynamic classes are considered to be an error. When merging class instance tables 416, duplicate entries are considered to be an error. Duplicate entries in the class symbol tables 408 are removed. There are no duplicate entries in the class relocation tables 412.”

Regarding **claim 15**, Palay discloses (28: 54-56) that the duplicated elements comprise elements of constant pools of respective class files, said shared table comprising a shared constant pool. Palay describes eliminating duplicates and creating merged tables of class information. "When merging the class definition tables 414, duplicate non-dynamic and internal dynamic class definitions are removed (redundant constants removed).”

Since requester did not request reexamination of claims 2-4, 8-10, 14, and 17-23 and did not assert the existence of a substantial new question of patentability (SNQP) for such claims (see 35 U.S.C. § 311(b)(2); see also 37 CFR 1.915b and 1.923), such claims will not be reexamined. This matter was squarely addressed in *Sony Computer Entertainment America Inc., et al v. Jon W. Dudas*, Civil Action No. 1:05CV1447 (E.D.Va. May 22, 2006), Slip Copy, 2006 WL 1472462. (Not Reported in F.Supp.2d.) The District Court upheld the Office's discretion to not reexamine claims in an inter partes reexamination proceeding other than those claims for which reexamination had specifically been requested. The Court stated:

Art Unit: 3992

To be sure, a party may seek, and the PTO may grant, inter partes review of each and every claim of a patent. Moreover, while the PTO in its discretion may review claims for which inter partes review was not requested, nothing in the statute compels it to do so. To ensure that the PTO considers a claim for inter partes review, § 311(b)(2) requires that the party seeking reexamination demonstrate why the PTO should reexamine each and every claim for which it seeks review. Here, it is undisputed that Sony did not seek review of every claim under the '213 and '333 patents. Accordingly, Sony cannot now claim that the PTO wrongly failed to reexamine claims for which Sony never requested review, and its argument that AIPA compels a contrary result is unpersuasive.

The Sony decision's reasoning and statutory interpretation apply analogously to ex parte reexamination, as the same relevant statutory language applies to both inter partes and ex parte reexamination. 35 U.S.C. § 302 provides that the ex parte reexamination "request must set forth the pertinency and manner of applying cited prior ' art to every claim for which reexamination is requested" (emphasis added), and 35 U.S.C. § 303 provides that "the Director will determine whether a substantial new question of patentability affecting any claim of the patent concerned is raised by the request..." (Emphasis added). These provisions are analogous to the language of 35 U.S.C. § 311(b)(2) and 35 U.S.C. § 312 applied and construed in Sony, and would be construed in the same manner. As the Director can decline to reexamine non-requested claims in an inter partes reexamination proceeding, the Director can likewise do so in ex parte reexamination proceeding. See Notice of Clarification of Office Policy To Exercise Discretion in Reexamining Fewer Than All the Patent Claims (signed Oct. 5, 2006) 1311 OG 197 (Oct. 31, 2006). See also

Art Unit: 3992

MPEP § 2240, Rev. 5, Aug. 2006. Therefore, claims 2-4, 8-10, 14, and 17-23 will not be reexamined in this ex parte reexamination proceeding.

Conclusion

Amendment Proposed in Reexamination – 37 CFR 1.530(d) Patent owner is notified that any proposed amendment to the specification and/or claims in this reexamination proceeding must comply with 37 CFR 1.530(d)-(j), must be formally presented pursuant to 37 CFR 1.52(a) and (b), and must contain any fees required by 37 CFR 1.20(c).

In order to ensure full consideration of any amendments, affidavits or declarations, or other documents as evidence of patentability, such documents must be submitted in response to this Office action. Submissions after the next Office action, which is intended to be a final action, will be governed by the requirements of 37 CFR 1.116, after final rejection and 37 CFR 41.33 after appeal, which will be strictly enforced.

Any paper filed with the USPTO, i.e., any submission made, by either the Patent Owner or the Third Party Requester must be served on every other party in the reexamination proceeding, including any other third party requester that is part of the proceeding due to merger of the reexamination proceedings. As proof of service, the party submitting the paper to the Office must attach a Certificate of Service to the paper, which sets forth the name and address of the party served and the method of service. Papers filed without the required Certificate of Service may be denied consideration. See 37 CFR 1.550(f)

Art Unit: 3992

Please mail any communications to: Attn: Mail Stop "Ex Parte Reexam"
Central Reexamination Unit
Commissioner for Patents
P. O. Box 1450
Alexandria VA 22313-1450

Please FAX any communications to: (571) 273-9900
Central Reexamination Unit

Please hand-deliver any communications to: Customer Service Window
Attn: Central Reexamination Unit
Randolph Building, Lobby Level
401 Dulany Street
Alexandria, VA 22314

Registered users of EFS-Web may alternatively submit such correspondence via the electronic filing system EFS-Web, at <https://spportal.uspto.gov/authenticate/authenticateuserlocalepf.html>.

EFS-Web offers the benefit of quick submission to the particular area of the Office that needs to act on the correspondence. Also, EFS-Web submissions are "soft scanned" (i.e., electronically uploaded) directly into the official file for the reexamination proceeding, which offers parties the opportunity to review the content of their submissions after the "soft scanning" process is complete. Any inquiry concerning this communication or earlier communications from the examiner, or as to the status of this proceeding, should be directed to the Central Reexamination Unit at telephone number (571) 272-7705.

/Mary Steelman/

Primary Examiner

Central Reexam Unit 3992

Conferees:

