

# Court Declines to Use Intrinsic Evidence to Narrow Construction or Find Indefiniteness for Terms whose Ordinary Meanings are Clear

*Ancora Technologies, Inc. v. Apple, Inc.* (March 3, 2014)

Presented by Jason Rhodes  
BSKB Electrical Dept. Lecture, March 12, 2014



## Table of Contents

- Background
- Procedural History
- Parties' Arguments
- Fed. Cir. Holding 1: "Program" Construed too Narrowly
- Fed. Cir. Holding 2: "Volatile Memory" Not Indefinite
- Discussion



# Background



# Background

- Ancora owns US Pat. No. 6,411,941 & asserts it against Apple
- Method of preventing unauthorized software use
- Checks whether a software program is operating within a license, and stops the program (or takes other remedial action) if it is not
- Prior methods of checking license coverage of software either vulnerable to hacking, required additional hardware (expensive & inconvenient), or otherwise limited distribution
- To overcome these problems, 941 patent uses a memory space associated with a computer's basic input/output system (BIOS) to store the encrypted license information used for verification

# Background

- While contents of BIOS memory space may be modified, it requires unusually high level of programming expertise to do so
- Also, risk of accidentally damaging BIOS & rendering computer inoperable is “too high of a risk for the ordinary software hacker to pay”
- Thus, patented method has following advantages:
  1. Only requires existing computer hardware (eliminating expense & inconvenience of add'l hardware),
  2. Stores verification information in space that is harder & riskier for hackers to tamper with

## Background - Claim Language (of only asserted claim)

1. A method of restricting software operation within a license for use with a computer including an erasable, non-volatile memory area of a BIOS of the computer, and a volatile memory area; the method comprising the steps of:
  - selecting a program residing in the volatile memory,
  - using an agent to set up a verification structure in the erasable, non-volatile memory of the BIOS, the verification structure accommodating data that includes at least one license record,
  - verifying the program using at least the verification structure from the erasable non-volatile memory of the BIOS, and
  - acting on the program according to the verification.

# Procedural History



# Procedural History

- Ancora sued Apple in U.S. District Court for the Northern District of California, alleging that products running Apple's iOS operating system infringed 941 patent
- 2 issues decided by dist. court:
  1. Whether "program" is limited to application programs, or whether it also covers operating systems,
  2. Whether "volatile memory" & "non-volatile memory" are definite under 35 USC 112(b)

# Procedural History

- Issue 1: dist. court agreed w/ Apple & construed “program” as being limited to application programs i.e. programs relying on OS in order to run
  - Based on this interpretation, Ancora stipulated to summary judgment of non-infringement
- Issue 2: dist. court agreed w/ Ancora that terms “volatile memory” and “non-volatile memory” are not indefinite under 35 USC 112(b)
- Ancora appealed dist. court’s construction of “program,” while Apple cross-appealed dist. court’s holding that claim terms are definite

## Procedural History – Federal Circuit

- Before Chief Judge Rader, Circuit Judges Taranto & Chen
- Decided March 3, 2014
- Reviews claim construction and indefiniteness *de novo*
  - Dist. court's claim construction given no deference
  - See *Lighting Ballast Control LLC v. Philips Electronics N.A. Corp.* (decided February 21, 2014), where Fed. Cir. reconfirmed holding of *Cybor Corp. v. FAS Techs., Inc.* (138 F.3d 1448 (Fed. Cir. 1998))

# Parties' Arguments



# Parties' Arguments – Issue 1

## Ancora

- “Program” should not be construed in way that excludes operating systems from the class of programs whose license is verified by claimed method

## Apple

- “Program” should be limited to applications based on prosecution history (applicants' statements distinguishing over prior art, examiner's reasons for allowance)
- Spec describes that the software performing license verification is “*a priori* running in the computer” when the program to be verified is loaded into memory

# Parties' Arguments – Issue 2

## Apple

- Spec refers to a hard disk as example of “volatile memory” w/o further explanation. This renders both “volatile memory” and “non-volatile memory” indefinite

- Col. 1, ll. 19-21 (in “Background of the Invention” section)

Software based products have been developed to validate authorized software usage by writing a license signature onto the computer's volatile memory (e.g. hard disk). These

- Col. 3, ll. 4-9 (in “Summary of the Invention” section)

An important advantage in utilizing non-volatile memory such as that residing in the BIOS is that the required level of system programming expertise that is necessary to intercept or modify commands, interacting with the BIOS, is substantially higher than those needed for tampering with data residing in volatile memory such as hard disk. Furthermore,

# Parties' Arguments – Issue 2

## Apple

- Col. 4, ll. 49-53 (in “Summary” section)

According to one, non-limiting, preferred embodiment of  
50 the present invention, the first non-volatile memory area is  
a ROM section of a BIOS; the second non-volatile memory  
area is a E<sup>2</sup>PROM section of a BIOS; and the volatile  
memory is a RAM e.g. hard disk and/or internal memory of  
the computer.

- Because “a hard disk is a quintessential example of non-volatile memory” and “the specification does not explain *how* a hard disk can fall into the category of volatile memory...or what characteristics differentiate volatile from non-volatile memory...a person of ordinary skill would not know what falls within the scope of the claims.”

# Fed. Cir. Holding 1: “Program” Construed too Narrowly



## Fed. Cir. Holding 1: “Program” Construed too Narrowly

- Claim term should be given its ordinary meaning in the pertinent context, unless patentee has made clear its adoption of a different definition or otherwise disclaimed that meaning
- Apple does not seriously dispute that ordinary meaning of “program” encompasses both operating systems & applications running on them (& other types of programs)

## Fed. Cir. Holding 1: “Program” Construed too Narrowly

- Claims themselves point against narrowing of term “program” to application programs
- Claim 1 recites a “method of restricting software operation” & only refers to restricted software as a “program” – never mentions “application”
- In contrast, indep. claim 18 (not asserted) recites “method for accessing an **application** software program,” and repeatedly refers to “**application** software program”
- Although claim 18 is not dependent, & claim differentiation is often of limited importance, this difference in terminology reinforces Ancora’s interpretation of “program”

## Fed. Cir. Holding 1: “Program” Construed too Narrowly

- Nothing in spec clearly narrows the term “program” to applications
- Specification refers to the software to-be-verified as “software program,” “software,” & “program” without limitation to particular type
- Spec only refers to such programs as applications in clearly identified **examples** i.e., “specific **non-limiting** example” and/or “preferred embodiment”
- Such examples are not sufficient to redefine term from plain & ordinary meaning

## Fed. Cir. Holding 1: “Program” Construed too Narrowly

- Prosecution history statements cited by Apple were made by applicants to distinguish over 103 combination of references: Misra (US 6,189,146) & Ewertz (US 5,479,639)
- Misra describes OS-level software (application) that enforces licenses – sets up verification structure & license record similar to patent, but does not store it in BIOS
- Ewertz describes BIOS routine which stores different type of data in a non-erasable area of BIOS

## Fed. Cir. Holding 1: “Program” Construed too Narrowly

- Prosecution history statements by applicants:

[T]here is no suggestion or motivation to combine Misra and Ewertz in the manner suggested in the Office Action. BIOS is a configuration utility. Software license management *applications, such as the one of the present invention*, are operating system (OS) level programs.... BIOS and OS level programs are normally mutually exclusive.

...

[T]he present invention proceeds against conventional wisdom in the art. Using BIOS to store *application data* such as that stored in Misra’s local cache for licenses is not obvious. The BIOS area is not considered a storage area for *computer applications*. An ordinary skilled artisan would not consider the BIOS as a storage medium to preserve application data for at least two reasons.

...

## Fed. Cir. Holding 1: “Program” Construed too Narrowly

First, ... [a]n ordinary person skilled in the art makes use of OS features to write data to storage mediums. There is no OS support whatsoever to write data to the system BIOS. Therefore, an ordinary person skilled in the art would not consider the BIOS as a possible storage medium....

Second, no file system is associated with the BIOS.... This is further evidence that OS level *application programmers* would not consider the BIOS as a storage medium for license data.

- Statements characterize the software *implementing the method* as an application, not the software *to be verified*
  - Applicants were merely arguing that verifying software differs from prior art in that it both (1) operates as application, and (2) uses BIOS level for data storage & retrieval

## Fed. Cir. Holding 1: “Program” Construed too Narrowly

- “Application programmers” refers to the programmers of the license-verifying software, not software to be verified
- Apple argues that the statement “Using BIOS to store **application** data...” means that license record must belong to application. Not necessarily. It could be referring to data being written by license-verifying application
  - In any event, such statement does not rise to level of disclaimer regarding nature of to-be-verified software

## Fed. Cir. Holding 1: “Program” Construed too Narrowly

- Apple also points to Examiner’s reasons for allowance as supporting its position. These reasons state:

[T]he closest prior art systems, singly or collectively, do not teach *licensed programs* running at the OS level interacting with a program verification structure stored in the BIOS to verify the program using the verification structure and having a user act on the program according to the verification.
- Statement is at worst a slip. Indisputably, the verifying software interacts w/ verification structure, not the verified software
- In any event, this was not applicants’ statement
  - See *Salazar v. Procter & Gamble Co.*, 414 F.3d 1342, 1345 (Fed. Cir. 2005) holding that remarks in the examiner’s statement of reasons for allowance are insufficient to limit claim scope

## Fed. Cir. Holding 1: “Program” Construed too Narrowly

- Apple argues that spec describes verifying software to be “a priori running in the computer” when program to be verified is loaded into memory (col. 2, ll. 10-19)

<sup>10</sup> Having placed the encrypted license record in the second non-volatile memory (e.g. the E<sup>2</sup>PROM), the process of verifying a license may be commenced. Thus, when a program is loaded into the memory of the computer, a so called license verifier application, that is a priori running in  
<sup>15</sup> the computer, accesses the program under question, retrieves therefrom the license record, encrypts the record utilizing the specified unique key (as retrieved from the ROM section of the BIOS) and compares the so encrypted record to the encrypted records that reside in the E<sup>2</sup>PROM. In the case of

- This description is part of what is merely a “non-limiting example” that is “by no means binding” (col. 1, ll. 44-45; col. 2, ll. 60-61)

## Fed. Cir. Holding 1: “Program” Construed too Narrowly

Conclusion: District court **erred** in construing “program” to mean “a set of instructions for software **applications** that can be executed by a computer”

# Fed. Cir. Holding 2: “Volatile Memory” Not Indefinite



## Fed. Cir. Holding 2: “Volatile Memory” Not Indefinite

- Under 35 USC 112(b), claim must be “sufficiently definite to inform the public of the bounds of the protected invention, i.e., what subject matter is covered by the exclusive rights of the patent.” *Halliburton Energy Servs., Inc. v. M-I LLC*, 514 F.3d 1244, 1249 (Fed. Cir. 2008)
- Supreme Court is currently considering how to refine formulations to apply definiteness requirement
  - Granted cert in *Nautilus, Inc. v. Biosig Instruments, Inc.* (Sup. Ct. No.13-369), re “insolubly ambiguous” standard
- However, in this case, indefiniteness challenge can be rejected without awaiting Sup. Ct.’s clarification

## Fed. Cir. Holding 2: “Volatile Memory” Not Indefinite

- Here, claim language & prosecution history leave no reasonable uncertainty about boundaries of terms at issue, even though certain parts of spec might be confusing when read in isolation
- Most important – there’s no dispute that “volatile memory” and “non-volatile memory” have meaning that is clear, settled & objective in content
- Everyone agrees that, to POSITA, “volatile memory is memory whose data is not maintained when the power is removed and a non-volatile memory is memory whose data is maintained when power is removed”

## Fed. Cir. Holding 2: “Volatile Memory” Not Indefinite

- This meaning gives public firm understanding of scope of claim terms
- This understanding is not supplanted by fact that spec refers 3 times to hard disk as example of volatile memory
- All sides agree that hard disk maintains data when power is removed & thus is not normally referred to as “volatile memory”
- However, terms at issue have so clear an ordinary meaning that POSITA would not be looking for clarification in spec
  - No facial ambiguity or obscurity in claim term
  - No claim even refers to a hard disk

## Fed. Cir. Holding 2: “Volatile Memory” Not Indefinite

- Spec nowhere purports to set out definition for “volatile” or “non-volatile” memory, & nothing in it reads like disclaimer of the clear ordinary meaning
- Under current claim construction law, a clear ordinary meaning is not properly overcome by a few passing references which do not amount to redefinition or disclaimer
- Also, skilled artisan would appreciate that passages at issue have possible meaning not starkly irreconcilable with clear meaning of claim terms

## Fed. Cir. Holding 2: “Volatile Memory” Not Indefinite

- Only example of volatile memory claimed is RAM (dep. claim 11)
- Well known that computer’s hard disk is routinely used as “virtual” memory to provide temporary storage when there’s insufficient RAM to complete an operation
- It is undisputed that virtual memory data becomes inaccessible through *usual means* once power is removed (even if it can still be found by more sophisticated means)
- This explanation finds support in spec., “the volatile memory is a **RAM** *e.g. hard disk* and/or internal memory of the computer”
  - Suggests that patentee only meant to refer to hard disk in capacity of supplemental memory in conjunction with main RAM, as opposed to completely redefining meaning of “volatile memory”

## Fed. Cir. Holding 2: “Volatile Memory” Not Indefinite

- Under the demanding standards for displacing as clear an ordinary meaning as exists in this case, POSITA could not have **reasonable** uncertainty about scope of the claims
- Also, prosecution history eliminates any reasonable basis to think patentee adopted different meaning than clear ordinary one

## Fed. Cir. Holding 2: “Volatile Memory” Not Indefinite

- In OA of 6/22/2001, Examiner explicitly referred to ordinary meaning of “non-volatile” to support 2 different grounds of rejection
  - In 102 rejection, said he was “relying on the standard definition of ‘non-volatile’ memory as memory that is maintained even when the power is removed from the storage system”
  - In 112, 2<sup>nd</sup> rejection, said that term “non-volatile” was being used in claim 1 in manner contrary to its usual meaning
    - “The term ‘non-volatile’ in claim 1 is used by the claim to exclude ‘hard disk,’ while it is accepted that a ‘hard disk’ is ‘non-volatile’ as it does not lose data when the power is removed from it” \*
- In response, Applicants **did not dispute** examiner’s understanding of “non-volatile,” but instead amended claim to restrict non-volatile memory to a memory area of the computer BIOS

## Fed. Cir. Holding 2: “Volatile Memory” Not Indefinite

- Apple argues that it is unclear whether claim 1 actually recites what “applicant regards as his invention” as required by 112(b)
- This requirement is distinct from 112(b)’s requirement that claim be sufficiently clear to be definite
- Here, Ancora embraces claim language’s clear & ordinary meaning – no evidence that applicants regarded their invention as something else
- Conversely, in case law cited by Apple, patentee agreed that claim language did not match what he regarded as his invention (and intrinsic record unambiguously showed this to be case)

## Fed. Cir. Holding 2: “Volatile Memory” Not Indefinite

Conclusion: District court was **correct** in rejecting Apple’s challenge to “volatile memory” and “non-volatile memory” as indefinite

# Discussion



# Discussion

- Is this at all similar to situation in *Teva Pharma. USA, Inc. v. Sandoz, Inc.* (723 F.3d 1363 (Fed. Cir. July 26, 2013))?
- In *Teva*, court found claim term “molecular weight” indefinite because, during the prosecution history of 2 related applications, applicants made contradictory statements as to how this weight is calculated
  - Statements were made in response to same type of rejections (indefiniteness) issued by same examiner
  - Statements caused examiner to withdraw both rejections, thus resulting in insoluble ambiguity

# Discussion

- Re interpretation of “program,” would outcome have been different under more deferential standard?
- Was public reasonably apprised of claim scope in view of “hard disk” examples?
- When inserting the hard disk examples, was the patent drafter really referring to its use as “virtual” memory?
- When drafting applications, should we avoid inserting “catchall” phrases with a bunch of examples? Or at least request inventor to give such a list special attention?

Thank you