

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

U.S. Pat. No.: 7,103,640

Issue Date: September 5, 2006

Appl. No.: 09/661,222

Filing Date: September 13, 2000

Pat. Title: NETWORK DISTRIBUTED TRACKING
WIRE TRANSFER PROTOCOL

Control No.: To be assigned

Mail Stop "*Ex Parte* Reexam"
Attn: Central Reexamination Unit
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

**REQUEST FOR *EX PARTE* REEXAMINATION OF
U.S. PATENT NO. 7,103,640**

Dear Commissioner:

Pursuant to 35 U.S.C. § 302 and 37 C.F.R. § 1.510, the undersigned, on behalf of Amazon Web Services, Inc. (the "Requester") requests *ex parte* reexamination of claims 17, 18, and 24 (the "Challenged Claims") of U.S. Patent No. 7,103,640 ("the '640 Patent," Exhibit 1), issued on September 5, 2006. The '640 Patent is assigned to Kove IO, Inc. ("Patent Owner").

Ex Parte Patent Reexamination Filing Requirements

Pursuant to 37 C.F.R. § 1.510(b)(1), this Request provides statements pointing out each substantial new question of patentability based on material, non-cumulative reference patents and printed publications for the Challenged Claims of the '640 Patent (*see* Section II). In particular, this Request raises two substantial new questions of patentability for obviousness under 35 U.S.C § 103 based on: (1) Oracle Corporation's manual *Oracle Names Administrator's Guide*,

Release 2.0 (“OracleNamesAdminGuide”) in view of Joseph B. Greene’s book *Oracle DBA Survival Guide* (“OracleSG”) and U.S. Patent No. 5,777,989 (“McGarvey”); and (2) OracleNamesAdminGuide in view of OracleSG and McGarvey and further in view of U.S. Patent No. 5,301,286 (“Rajani”).

Pursuant to 37 C.F.R. § 1.510(b)(2), this Request requests reexamination of the Challenged Claims of the ’640 Patent, and provides a detailed explanation of the pertinence and manner of applying the cited references to the Challenged Claims (*see* Section III).

Pursuant to 37 C.F.R. § 1.510(b)(3), the Request provides copies of every patent or printed publication relied upon or referred to in the statement pointing out each substantial new question of patentability or in the detailed explanation of the pertinence and manner of applying the cited references (*see* Exhibits 2-5).

Pursuant to 37 C.F.R. § 1.510(b)(4), the Request provides a copy of the ’640 Patent as Exhibit 1, along with a copy of any disclaimer, certificate of correction, and reexamination certificate issued corresponding to the patent.

Pursuant to 37 C.F.R. § 1.510(b)(5), the attached Certificate of Service indicates that a copy of this Request in its entirety has been served on Patent Owner at the following address of record for Patent Owner, in accordance with 37 C.F.R. § 1.33(c):

Crowell/BGL
P.O. Box 10395
Chicago, IL 60610

Also submitted herewith is the fee set forth in 37 C.F.R. § 1.20(c)(1).

Pursuant to 37 C.F.R. § 1.510(b)(6), Requester hereby certifies that the statutory estoppel provisions of 35 U.S.C. § 315(e)(1) and 35 U.S.C. § 325(e)(1) don’t prohibit Requester from filing this *ex parte* patent reexamination request.

TABLE OF CONTENTS

I.	Introduction	1
II.	Substantial New Questions Of Patentability	3
A.	'640 PATENT OVERVIEW	3
B.	LISTING OF PRIOR ART PATENTS AND PRINTED PUBLICATIONS.....	5
C.	STATEMENT SETTING FORTH EACH SUBSTANTIAL NEW QUESTION OF PATENTABILITY	5
D.	CO-PENDING LITIGATION	7
E.	ESTOPPEL	8
III.	Detailed Explanation Of The Pertinence And Manner Of Applying The Prior Art References To Every Claim For Which Reexamination Is Requested.....	8
A.	ORACLE OVERVIEW.....	8
B.	LEVEL OF SKILL IN THE ART.....	13
C.	CLAIM CONSTRUCTION.....	13
D.	OVERVIEW OF THE PRIOR ART.....	15
1.	OracleNamesAdminGuide (Ex. 3) Overview.....	15
2.	OracleSG (Ex. 3) Overview	31
3.	McGarvey (Ex. 4) Overview	35
4.	Rajani (Ex. 5) Overview	37
E.	SNQ 1: INDEPENDENT CLAIMS 17 AND 18 ARE OBVIOUS BASED ON ORACLENAMESADMINGUIDE IN VIEW OF ORACLESG AND MCGARVEY.	41
1.	Reasons to Modify OracleNamesAdminGuide in View of OracleSG and McGarvey.	42
2.	OracleNamesAdminGuide in View of OracleSG and McGarvey Renders Obvious Independent Claim 18.....	47

3.	OracleNamesAdminGuide in View of OracleSG and McGarvey Renders Obvious Independent Claim 17.	88
F.	SNQ 2: DEPENDENT CLAIM 24 ARE OBVIOUS BASED ON ORACLE NAMES ADMIN- GUIDE IN VIEW OF ORACLESG AND MCGARVEY AND FURTHER IN VIEW OF RAJANI.	92
1.	OracleSG’s Teachings Render Obvious Using a Hash Table.	92
2.	Reasons to Modify OracleNamesAdminGuide in View of OracleSG and McGarvey and Further in View of Rajani	94
3.	OracleNamesAdminGuide in View of OracleSG and McGarvey and Further in View of Rajani Renders Obvious Dependent Claim 24.	97
G.	SECONDARY CONSIDERATIONS.....	102
IV.	Conclusion	105

TABLE OF EXHIBITS

Exhibit	Description
1	U.S. Patent No. 7,103,640 to Overton et al., filed on September 13, 2000, issued on September 5, 2005 (“the ’640 Patent”)
2	<i>Oracle Names Administrator’s Guide, Release 2.0</i> (Oracle Corp., Redwood City, CA), published 1996 (“OracleNamesAdminGuide”)
3	Joseph B. Greene, <i>Oracle DBA Survival Guide</i> , 1st ed. (Sams Publishing, Indianapolis, Ind.), ISBN: 0672306816, published October 1, 1995 (“OracleSG”) (divided into Part 1 (front cover through page 326) and Part 2 (page 327 through the back cover))
4	U.S. Patent No. 5,777,989 to John McGarvey, filed December 19, 1995, issued July 7, 1998 (“McGarvey”)
5	U.S. Patent No. 5,301,286 to Purshotam Rajani, filed January 2, 1991, issued April 5, 1994 (“Rajani”)
6	Declaration of Joseph B. Greene Under 37 C.F.R. § 1.132 (“Greene Decl.”)
7	U.S. Patent No. 7,233,978 to Overton et al., filed on June 1, 2001, issued on June 19, 2007 (“the ’978 Patent”)
8	U.S. Patent No. 7,814,170 to Overton et al., filed on February 13, 2006, issued on October 12, 2010 (“the ’170 Patent”)
9	Prosecution History of the ’640 Patent (“’640 History”)
10	Prosecution History of the ’978 Patent (“’978 History”)
11	Prosecution History of the ’170 Patent (“’170 History”)
12	Departments of Informatics, <i>Oracle V2</i> , Virtual Exhibitions in Informatics, University of Klagenfurt (September 30, 2019), http://cs-exhibitions.uni-klu.ac.at/index.php?id=403 (last accessed October 4, 2021)
13	<i>Oracle7 Server Concepts, Release 7.3</i> (Oracle Corp., Redwood City, CA), published 1996
14	George Koch & Kevin Loney, <i>Oracle: The Complete Reference, Electronic Edition</i> (Oracle Press), ISBN: 0078822858, published 1996 (excerpts)
15	Kevin Loney, <i>Oracle DBA Handbook, 7.3 Edition</i> (McGraw-Hill Osborne Media) ISBN: 0078822890, published 1997 (excerpts)

Exhibit	Description
16	<i>Kove IO, Inc. v. Amazon Web Services, Inc.</i> , 1:18-cv-08175 (N.D. Ill.), Kove’s Memorandum in Opposition to Amazon Web Services, Inc.’s Motion for Judgment on the Pleadings, D.I. 145 (May 29, 2020)
17	<i>Kove IO, Inc. v. Amazon Web Services, Inc.</i> , 1:18-cv-08175 (N.D. Ill.), Joint Claim Construction Chart (Updated), D.I. 382 (August 11, 2021)
18	Declaration of Sylvia D. Hall-Ellis, Ph.D. Under 37 C.F.R. § 1.132 Regarding Publication of <i>Oracle Names Administrator’s Guide, Release 2.0</i> (Exhibit 2)
19	Library of Congress Catalog Listing for Joseph B. Greene, <i>Oracle DBA Survival Guide</i> , 1st ed. (Sams Publishing, Indianapolis, Ind.), ISBN: 0672306816 (Ex. 3)
20	<i>Amazon Web Services, Inc., v. Kove IO, Inc.</i> , IPR2020-00276, Paper 11 (Patent Owner’s Preliminary Response) (PTAB March 19, 2020)
21	<i>Amazon Web Services, Inc., v. Kove IO, Inc.</i> , IPR2020-00276, Paper 23 (Decision) (PTAB March 19, 2020)
22	Oracle - Investor Relations – FAQ, Frequently Asked Questions, https://investor.oracle.com/faq/default.aspx (last accessed October 4, 2021)
23	<i>Kove IO, Inc. v. Amazon Web Services, Inc.</i> , 1:18-cv-08175 (N.D. Ill.), Kove IO, Inc.’s Final Enforceability and Validity Contentions (August 13, 2020)
24	NSN 7610-01-448-8459, 7610014488459, ISO Group Defense and Aerospace Supply Chain Partner, https://www.iso-group.com/NSN/7610-01-448-8459 (last accessed on October 12, 2021)

USE OF EMPHASIS IN QUOTATIONS

All emphases in quotations and exhibit citations have been added, unless otherwise indicated.

APPLICABLE STATUTES

References to 35 U.S.C. §§102 and 103 are to the pre-AIA versions applicable to the ’640 Patent.

I. INTRODUCTION

Claims 17, 18, and 24 of the '640 Patent should be rejected for obviousness under 35 U.S.C. § 103. The combination of OracleNamesAdminGuide, OracleSG, and McGarvey renders obvious independent claims 17 and 18. And the further teachings of Rajani renders obvious the additional elements of claim 24, which depends from claim 18.

More particularly, the '640 Patent is directed to techniques for locating data stored in a distributed network.¹ To do so, the patent describes storing the network addresses of data in a location server, then having clients query the location server to retrieve the network address of the desired data.² OracleNamesAdminGuide discloses the Oracle7 distributed data storage system that uses the same techniques described in the '640 Patent.

Like the '640 Patent, OracleNamesAdminGuide answers the question in a distributed data storage system: “where is my data?” OracleNamesAdminGuide describes a technique that separates the “where” (i.e., location of the data repository) from the “what” (i.e., the data itself).³ OracleNamesAdminGuide does this by using specialized servers called Names Servers, which serve as intermediaries between clients requesting data and the corresponding data storage servers.⁴ A client seeking to locate data on OracleNamesAdminGuide’s network will query a Names Server to retrieve the data’s network address, then use that address to connect to the entity housing the data. OracleNamesAdminGuide illustrates this network architecture in Figure 3-11 below.⁵

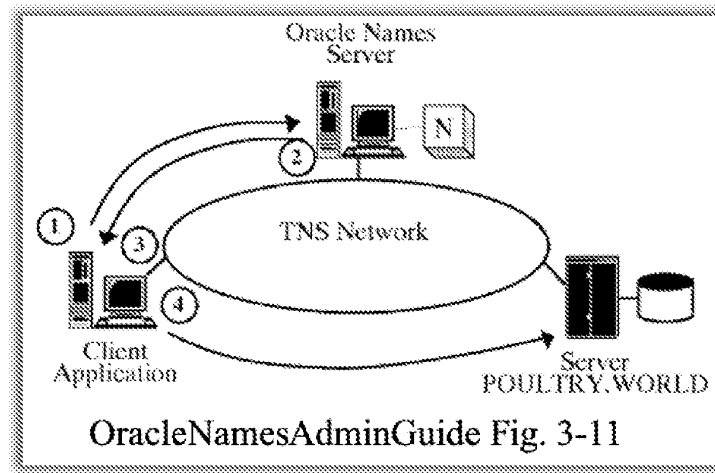
¹ See '640 Patent at Abstract.

² See *id.* at Abstract, 2:12-27.

³ OracleNamesAdminGuide at 20, 23, 33, 43-44, 52-53.

⁴ See *id.* at 20, 22.

⁵ Steps 1-4 appear in OracleNamesAdminGuide’s original Figure 3-11.



At step ①, the client requests the location of data uniquely identified by the POULTRY.WORLD identifier. OracleNamesAdminGuide uses its networking utility, SQL*Net, to send the client's request to the Names Server. At step ②, the Names Server receives the request, looks up the location information associated with the POULTRY.WORLD identifier, then sends that location information to client. The client receives the location information from the Names Server at step ③, then the client uses that information to build a connection request to the server housing the data. Finally, at step ④, the client contacts the server and establishes a client-server connection.⁶

Before the '640 claims were allowed, the Examiner wasn't presented with and didn't consider the OracleNamesAdminGuide, OracleSG, McGarvey, or Rajani references, which describe routine networking techniques like redirect messaging and hashing. Accordingly, this Request raises substantial new question of patentability ("SNQs") for claims 17, 18, and 24.

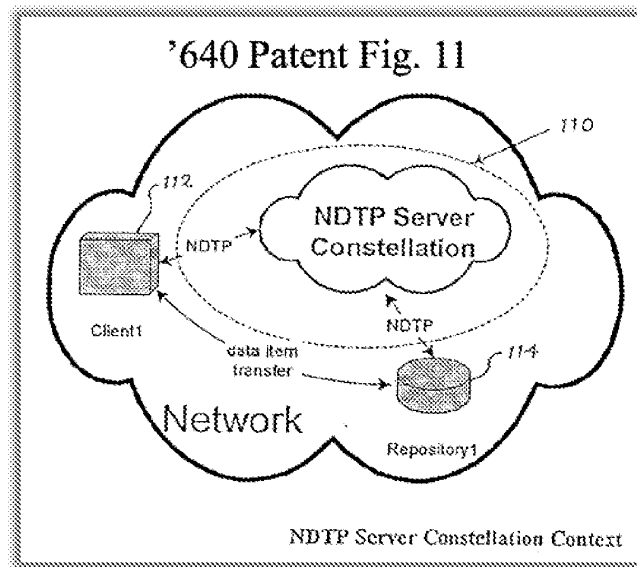
⁶ See *id.* at 52-53; see also *id.* at 128-129 (describing location information syntax includes host name, port, and database instance along with the connection information).

II. SUBSTANTIAL NEW QUESTIONS OF PATENTABILITY

Requester respectfully requests reexamination for claims 17, 18, and 24 of the '640 Patent under 35 U.S.C. § 302 and 37 C.F.R. § 1.510.

A. '640 Patent Overview

The '640 Patent was filed on September 13, 2000.⁷ The '640 Patent is directed to well-known techniques for locating data stored in a distributed networking environment.⁸ To do so, the patent describes using an identifier associated with the location of a respective data entity.⁹ And the patent uses data location servers to store location information associated with those identifiers as shown below in Figure 11.¹⁰



⁷ See '640 Patent at cover.

⁸ See *id.* at Abstract.

⁹ See *id.* at Abstract, 2:12-27.

¹⁰ See *id.* at 16:53-17:7.

The '640 Patent states that it was known in the art that distributed databases allow physical storage of a database to exist across multiple locations and that, to access such data, a client would send a request to the server.¹¹ And the patent acknowledges that hashing algorithms were known in the art to index location tables, such as the “hashpjw function presented by Aho, Sethi and Ullman in their text *Compilers, Principles, Techniques and Tools*.”¹²

During the original examination, the Examiner twice rejected the claims based on prior art. This prior art showed it was known for a distributed data system to have a client requesting the location of data associated with an identifier, and a location server retrieving the location information for the entity based on the identifier and sends that information to the client.¹³ And this prior art showed it was known that “use of a [] hashing operation in a computer networking environment was well known in the art at the time the invention was made.”¹⁴

In response, Applicant added to the claims the limitation that, if the location server lacks location information for the requested data, the location server will send a *redirect message* to the client.¹⁵ The amended claims further recite that the redirect message includes information useable by the client to locate another location server having the location information for the request entity.¹⁶

¹¹ See *id.* at 1:34-36, 1:56-65.

¹² *Id.* at 15:1-20; Greene Decl. (Ex. 6) ¶¶ 28-29.

¹³ See '640 History at 440-449, 488-498 (12/25/2004, 02/09/2005 Office Actions).

¹⁴ See *id.* at 447-448 (02/24/2004 Office Action).

¹⁵ See *id.* at 514-526 (08/09/2005 Amendments).

¹⁶ See *id.*

Below, Requester establishes that the limitations of issued claims 17, 18, and 24 were already known in the art, as shown by OracleNamesAdminGuide's distributed database storage system and other art.

B. Listing of Prior Art Patents and Printed Publications

Pursuant to 37 C.F.R. § 1.510(b)(3), this Request provides a copy of every patent or printed publication relied upon or referred to herein:

- **Exhibit 2:** *Oracle Names Administrator's Guide, Release 2.0* (Oracle Corp., Redwood City, CA), published 1996 ("OracleNamesAdminGuide");
- **Exhibit 3:** Joseph B. Greene, *Oracle DBA Survival Guide*, 1st ed. (Sams Publishing, Indianapolis, Ind.), ISBN: 0672306816, published October 1, 1995 ("OracleSG");
- **Exhibit 4:** U.S. Patent No. 5,777,989 to John McGarvey, filed December 19, 1995, issued July 7, 1998 ("McGarvey"); and
- **Exhibit 5:** U.S. Patent No. 5,301,286 to Purshotam Rajani, filed January 2, 1991, issued April 5, 1994 ("Rajani").

A Form SB-08 and copies of the foregoing references are submitted herewith.

C. Statement Setting Forth Each Substantial New Question of Patentability

For claims 17 and 18, this Request presents a substantial new question of patentability based on obviousness under 35 U.S.C. § 103 over OracleNamesAdminGuide in view of OracleSG and McGarvey ("SNQ 1"). For claim 24, this Request presents a substantial new question of patentability based on obviousness under 35 U.S.C. § 103 over OracleNamesAdminGuide in view of OracleSG and McGarvey and further in view of Rajani ("SNQ 2").

Requester submits that the prior art references raise a new "substantial question of patentability" because "the teaching of the (prior art) patents and printed publications is such that a reasonable examiner would consider the teaching to be important in deciding whether or not the

claim is patentable.”¹⁷ For example, these references, discussed in further detail in Section III, when considered as a combination in view of their shared teachings, teach each limitation of the claims, including:

- OracleNamesAdminGuide discloses, or renders obvious in view of OracleSG and McGarvey, claim 17’s “in response to receiving a data location request from a client to retrieve a location string associated with an identification string provided in the data location request, transmitting a redirect message to the client if the identification string is not associated with a location string at the data location server, wherein the redirect message contains information for use by the client to calculate a location of a different data location server in the plurality of data location servers, wherein the different data location server contains the location string,” and a similar limitation in claim 18; and
- OracleNamesAdminGuide renders obvious alone or in view of OracleSG, McGarvey, and Rajani, claim 24’s “the location information comprises a portion of a hash table distributed over the plurality of data location servers.”

Further, these prior art references are new because the “same question of patentability as to the claim has not been decided by the Office in an earlier concluded examination or review of the patent.”¹⁸ None of these references were raised, considered, or formed the basis of a rejection during the original prosecution of the ’640 Patent.

Overall, this Request provides the detailed mapping and explanation regarding the combination of references and raises a SNQ because the Office hasn’t previously considered the teaching of the combinations of references in SNQ 1 or SNQ 2.

¹⁷ MPEP 2242.

¹⁸ *Id.*

Further, Mr. Greene's declaration (Ex. 6) reinforces the Request's analyses. Mr. Greene has more than 30 years of professional experience implementing, managing, and analyzing Oracle products. Based on this experience, he has published six books on Oracle's database management system and its associated tools and utilities, including OracleSG.¹⁹ Mr. Greene concludes that the combination of OracleNamesAdminGuide in view of OracleSG and McGarvey renders obvious claims 17 and 18, and that the combination of OracleNamesAdminGuide in view of OracleSG and McGarvey and further in view of Rajani renders obvious claim 24.²⁰

D. Co-Pending Litigation

Requester is currently engaged in a pending litigation concerning the '640 Patent, namely *Kove IO, Inc. v. Amazon Web Services, Inc.*, 1:18-cv-08175 (N.D. Ill.). Requester is unaware of any pending prosecution concerning the '640 Patent.

In Reexamination Control No. 90/014,552, Aneeta Mishra, who has no relationship with the instant Requester, filed a request for *ex parte* reexamination of claims 17, 18, and 24 of the '640 Patent. The USPTO denied that request on September 9, 2020. None of the prior art references presented in this Request were before the USPTO in the 90/014,552 reexamination.

U.S. Patent No. 7,233,978 ("978 Patent," Ex. 7) claims to be a continuation-in-part of the application that issued as the '640 Patent. In IPR2020-00276, Requester filed a petition seeking *inter partes* review of '978 claims 1, 3, 6, 10, 14, 17, 23, 24, 30, and 31. The Board denied

¹⁹ A full description of Mr. Greene's professional background and qualifications can be found at paragraphs 4-13 of his declaration.

²⁰ See Greene Decl. ¶¶ 46-47, 116, 117, 126.

this petition on June 16, 2020 in Paper 23. None of the prior art references presented in this Request were before the Board in IPR2020-00276.²¹

E. Estoppel

The statutory estoppel provisions of 35 U.S.C. § 315(e)(1) and 35 U.S.C. § 325(e)(1) don't prohibit Requester from filing this *ex parte* reexamination request.

III. DETAILED EXPLANATION OF THE PERTINENCE AND MANNER OF APPLYING THE PRIOR ART REFERENCES TO EVERY CLAIM FOR WHICH REEXAMINATION IS REQUESTED

In accordance with 37 C.F.R. § 1.510(b)(2), Requester provides the following detailed explanation of the pertinence and manner of applying the prior art to the Challenged Claims.

A. Oracle Overview

Oracle Corporation develops and manufactures commercial database systems. Oracle released its first commercially available database in 1979, Oracle v2.²² By 1992, Oracle had released Oracle version 7.²³

Mr. Greene details how Oracle7 included full functionality for distributed databases.²⁴ For example, the *Oracle7 Server Concepts* manual explains the concept of a distributed database as follows:

A distributed database is a network of databases managed by multiple database servers that appear to a user as a single logical database. The data of all databases

²¹ *Contra In re Vivint, Inc.*, 2021 WL 4448620, at *2, *8 (Fed. Cir. Sept. 29, 2021) (ordering USPTO to dismiss an *ex parte* reexamination request that simply “repackaged” arguments raised in a prior IPR petition). In the instant Request, none of the relied upon prior art references or obviousness arguments were presented previously in an IPR or reexamination challenging the '640 Patent or any related patent.

²² See Greene Decl. ¶ 14 (citing Departments of Informatics' Oracle v2 timelines (Ex. 12)).

²³ See Greene Decl. ¶ 15.

²⁴ See *id.* ¶¶ 15-16.

in the distributed database can be simultaneously accessed and modified. The primary benefit of a distributed database is that the data of physically separate databases can be logically combined and potentially made accessible to all users on a network.²⁵

To operate as a distributed database system, Mr. Greene explains that a system includes a management utility stored on each client and each database server for sending requests to a database instance and for returning responses and data from the database to the client over the network. An Oracle7 system may be hosting dozens of database instances on any given database server. So to make a connection, the management utility must know the client's and database server's IP address or "host name," the ports used, the database instance, and the transport protocol.²⁶

Further, Mr. Greene explains that the Oracle7 system included two advances—namely, the expansion of location transparency and the use of hashing in distributed databases.²⁷ In distributed databases, location transparency hides the actual location of data from the users, allowing them to "see through" the location and internal storage information to the data itself. Location transparency is an important feature for maintaining and updating a distributed database on an ongoing basis. As the book *Oracle: The Complete Reference* details:

Over the life span of an application, its data will very likely move from one database to another, or from one host to another. Therefore it will simplify maintenance of an application if the exact physical location of a database object is shielded from the user (and the application).²⁸

²⁵ Ex. 13 (Oracle7 Server Concepts) at 71-72.

²⁶ See Greene Decl. ¶ 17.

²⁷ See *id.* ¶ 18.

²⁸ Ex. 14 (Oracle: The Complete Reference) at 5.

And Mr. Greene details that, in 1996, Oracle introduced further distributed database capabilities, including the release of Oracle version 7.3, the Oracle tools and utilities SQL*Net release 2.3.3, and Oracle Names release 2.0.²⁹

SQL*Net is the name for Oracle's add-on networking utility. This utility provides the network underpinnings for distributed databases.³⁰ The 1994 *Oracle DBA Handbook* explains that SQL*Net makes it "possible to create synonyms that give applications true network transparency; the user who submits the query will not know the location of the data that is used to resolve it."³¹ SQL*Net Client runs on each client and interfaces with applications running on the client. SQL*NET Listener runs on database servers, listens on multiple network ports and IP addresses for client requests, and returns requested information to clients.³²

Oracle Names is the moniker for Oracle's distributed database name server utility, and provides a directory of databases on Names Servers.³³ Names Servers are intermediaries that provide connection information, including location, to clients requesting data from corresponding data storage servers. Using Oracle Names, a client seeking data requests the location of the data from a Names Server, receives a location response from the Names Server, then requests the data itself from the data storage server identified in the location response.³⁴

²⁹ See Greene Decl. ¶ 19.

³⁰ See *id.* ¶ 20.

³¹ See Ex. 15 (Oracle DBA Handbook) at 13.

³² See Greene Decl. ¶ 20.

³³ At times, those in the field refer to Oracle Names as "Oracle*Names." See *id.* ¶ 21 n.6.

³⁴ See *id.* ¶ 21.

With Oracle Names, Mr. Greene explains that all tasks for managing distributed databases are handled using a global naming service, available to all hosts in a network. As the book *Oracle: The Complete Reference* explains:

This changes the way in which database links are resolved. Before, when a database link was specified, the database first looked at the user's private database links. If none with the matching name was found, then the available public database links were checked.

Oracle*Names adds an additional level to this. Now, if the first two checks do not return a match for the database link name, then the Oracle*Names server's list of global database link names is searched for the database link. If the link name is found there, Oracle*Names will return the link's specifications and resolve the query.

This greatly simplifies the administration of *location transparency*—the ability to mask the physical location of data—in a distributed environment. The information related to remote data access is now stored in a central location. The impact of this is felt every time an object's location is changed. For example, if there were multiple links using explicit connections to a single remote database, then a pre-Oracle*Names modification to the user's password would require dropping and re-creating multiple database links.³⁵

Oracle maintains a separate manual for Oracle Names release 2.0, which is titled *Oracle Names Administrator's Guide, Release 2.0* (Exhibit 2).³⁶

Mr. Greene explains that Oracle Names and SQL*Net allow customers to create unique customer identifiers associated with a data set's network address, and to use those identifiers to query Names Servers to locate where in the system the data set resides.³⁷

³⁵ See Ex. 14 (Oracle: The Complete Reference) at 8.

³⁶ See OracleNamesAdminGuide at 4.

³⁷ See Greene Decl. ¶ 22.

Like most relational database management system (“RDBMS”) products at the time, Oracle7 included hashing capability. Hashing was a known approach at the time to improve performance by processing certain types of queries more quickly.³⁸ The *Oracle DBA Handbook* explains Oracle7’s use of this hashing capability as follows:

In Oracle7, a second type of cluster is supported. Hash clusters use hashing functions on the row’s cluster key to determine the physical location where the row should be stored. This will yield the greatest performance benefit for queries of the form shown in the following listing.

```
SELECT name
FROM employee
WHERE empno = 123;
```

In this example, the EMPLOYEE table is queried for an exact match for the EMPNO column. If that table was part of the hash cluster, and EMPNO was part of the cluster key, then the database could use the hashing function to quickly determine where the data was physically located.³⁹

And as the *Oracle DBA Handbook* illustrates, it was understood in the art that hashing reduced the processing cost and time for query results:

Queries in which a column’s value is compared to an exact value are called equivalence queries. In Oracle7, a new type of cluster, called a hash cluster, is made available. This type of cluster stores a row in a specific location based on its value in the cluster key column. Every time a row is inserted, its cluster key value is used to determine which block it should be stored in; this same logic can be used during queries to quickly find data blocks that are needed for retrieval.⁴⁰

³⁸ See *id.* ¶¶ 23-24.

³⁹ Ex. 15 (Oracle DBA Handbook) at 5-6.

⁴⁰ *Id.* at 9.

Today, Oracle is up to version 21.⁴¹ Patent Owner didn't provide any Oracle publications to the Patent Office during the original examination of the '640 Patent.

B. Level of Skill in the Art

At the time of the purported invention, a POSITA would have had a bachelor's degree in computer science (or a similar field) and at least two years of experience working with database management software systems. Alternatively, a POSITA would have had a Master's degree or similar post-graduate studies in the field of database management software systems with fewer years of experience.⁴²

C. Claim Construction

In the pending district court matter, Patent Owner has stated that the '640 Patent expired on April 8, 2020.⁴³ Because the '640 Patent has expired, the claims should be given their ordinary and customary meaning as set forth in *Phillips v. AWH Corporation*. The MPEP states:

During reexamination . . . claims are given the broadest reasonable interpretation consistent with the specification and limitations in the specification are not read into the claims.... In a reexamination proceeding involving claims of an expired patent, claim construction pursuant to the principle set forth by the court in *Phillips* ... (words of a claim 'are generally given their ordinary and customary meaning' as understood by a person of ordinary skill in the art in question at the time of the invention) should be applied since the expired claim are not subject to amendment."⁴⁴

Thus, this Request applies the *Phillips* claim construction standard.

⁴¹ See Greene Decl. ¶ 25.

⁴² See *id.* ¶¶ 38-42.

⁴³ See Ex. 16 (Memo in Opposition) at 8.

⁴⁴ MPEP 2258 (citing *In re Yamamoto*, 740 F.2d 1569 (Fed. Cir. 1984) and *Phillips v. AWH Corp.*, 415 F.3d 1303, 1316 (Fed. Cir. 2005)); Greene Decl. ¶¶ 43-45.

In the district court matter, Patent Owner has proposed constructions for certain claim terms and agreed to constructions for other terms.⁴⁵ For purposes of this reexamination only, Requester shows the prior art invalidates the Challenged Claims using the following constructions proposed or adopted by Patent Owner:

- “location” means “an encoding that is a member of a set of associations with an identifier in a location server, and that specifies where data pertaining to the entity identified by the identifier is stored” (claims 17-18, 24);
- “location server” means “a network-attached component that maintains a set of identifier/location mappings that are modified or returned in response to location request messages from clients” (claims 17-18, 24);
- “location information” means “information pertaining to one or more locations of data and/or the identities of one or more location servers” (claims 17-18, 24);
- “identifier” / “identifier string” / “identification string” means “a unique encoding that identifies an individual entity, and with which zero or more location strings are associated in a location server” (claims 17-18);
- “client” means “a network-attached component that initiates update or lookup of identifier/location mappings from a location server with location request messages” (claims 17-18, 24);
- “hash table” means “a data structure that stores values in a table, where values are stored and retrieved by applying a hash function to an input and using the function result as an index into the table” (claim 24); and
- “data location sewer network” means “data location server network” (claim 17).⁴⁶

Because the prior art establishes each Challenged Claim is obvious under Patent Owner’s proposed or adopted constructions, Requester has applied those for purposes of this reexamina-

⁴⁵ See Ex. 17 (Joint Claim Construction Chart) at 4-6.

⁴⁶ *Id.*

tion only. Requester doesn't concede the scope of any claim term in litigation or waive any argument in any proceeding that the claim terms are indefinite or the claims are invalid or unpatentable. For the purposes of this proceeding, Requester construes all other terms as having their ordinary and customary meanings. As shown below in Sections III.E-III.F, the prior art renders obvious the Challenged Claims under any reasonable construction of the claim terms.

D. Overview of the Prior Art

This Request establishes that the Challenged Claims are obvious based on the teachings of OracleNamesAdminGuide, OracleSG, McGarvey, and Rajani. During the original examination of the '640 Patent, the Examiner did not cite, consider, or rely on any of the references.

1. OracleNamesAdminGuide (Ex. 3) Overview

OracleNamesAdminGuide is a printed publication from the Oracle Corporation. On its face, OracleNamesAdminGuide shows that it was published by the Oracle Corporation in 1996 as Part No. A42482-1.⁴⁷ Oracle Corporation made OracleNamesAdminGuide accessible to the public on Oracle's website.⁴⁸ Mr. Greene confirms that, as of 1996, Oracle Corporation "was an established publisher that was well known to those interested in the field of database technologies."⁴⁹

⁴⁷ See OracleNamesAdminGuide at 4.

⁴⁸ See, e.g., https://docs.oracle.com/pdf/A42482_1.pdf (last accessed October 13, 2021).

⁴⁹ Greene Decl. ¶ 49; see also *Vidstream LLC v. Twitter, Inc.*, 981 F.3d 1060, 1065 (Fed. Cir. 2020) ("A reference will be considered publicly accessible if it was disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art exercising reasonable diligence[] can locate it.... When there is an established publisher there is a presumption of public accessibility as of the publication date.").

In Exhibit 18, Dr. Sylvia Hall-Ellis corroborates that OracleNamesAdminGuide was publicly accessible by at least May 7, 1997.⁵⁰ Dr. Hall-Ellis identifies a College of William & Mary library catalog entry that shows that OracleNamesAdminGuide was entered into that library's collection on May 7, 1997, where it was indexed by subject matter for public access.⁵¹ As Dr. Hall-Ellis explains, this catalog entry establishes that OracleNamesAdminGuide was publicly accessible no later than May 7, 1997.⁵²

Exhibit 24 further corroborates the public accessibility of OracleNamesAdminGuide. Exhibit 24 is a National Stock Number ("NSN") entry, which shows that the U.S. government assigned OracleNamesAdminGuide the NSN 7610-01-448-8459 on September 11, 1997.⁵³ The U.S. government assigns an NSN to publications, among other items in commerce, in order to process their procurement for the Department of Defense.⁵⁴

Thus, OracleNamesAdminGuide is a printed publication that is prior art under 35 U.S.C. § 102(b) because it published more than one year prior to the application date of the '640 Patent and under § 102(a) because authors other than the named inventors published it before the application date. And, even if Patent Owner were able to establish the '640 Patent is entitled to the

⁵⁰ See Ex. 18 (Hall-Ellis Decl.) ¶ 19. Dr. Hall-Ellis is an Adjunct Professor in the School of Information at San Jose State University in San Jose, California. She holds a Ph.D. in Library Science from the University of Pittsburgh. See *id.* at pp.12-14 (Attachment B).

⁵¹ See *id.* ¶ 31.

⁵² See *id.* ¶ 26.

⁵³ Ex. 24 at 2-3.

⁵⁴ See, e.g., https://www.dla.mil/Portals/104/Documents/J3LogisticOperations/Brochures/J3_nsnbooklet_200323.pdf?ver=2020-03-24-105311-683. As shown in Exhibit 24, OracleNamesAdminGuide was assigned a National Stock Number on September 17, 1997, corroborating the ongoing public accessibility of OracleNamesAdminGuide.

benefit of the earliest possible priority date of July 8, 1998, OracleNamesAdminGuide is §§ 102(a) and (b) prior art based on its publication date.

OracleNamesAdminGuide describes a real product, Oracle Names, which is a part of an Oracle7 database system. OracleNamesAdminGuide discloses a distributed database system, focusing on the Oracle Names utility that provides connection information for service locations residing on different clients and servers.⁵⁵ And OracleNamesAdminGuide explains that its Oracle7 system includes a Names Server that responds to a client's data requests using unique customer identifiers associated with the data's network address:

An Oracle Names Server *stores names and addresses for network services* such as databases or MultiProtocol Interchanges, database link definitions, and object aliases.

The Names Server stores the following types of objects, which are described below:

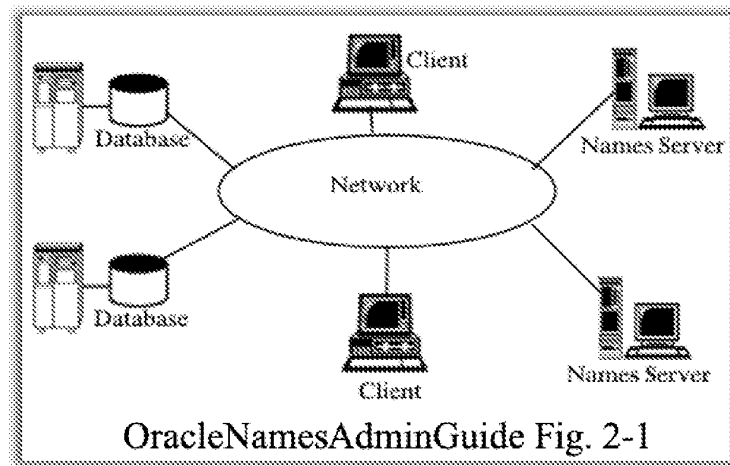
- database service names—*The service name is the global database name that is mapped to the SQL*Net connect descriptor*. The combination of a listener address and connect data make up a connect descriptor that locates a database service. Gateways to non-Oracle databases and Oracle RDB databases are also stored....
- V1 connect strings—A V1 *connect string can be mapped to a service name in the Oracle Names database*. The service name you enter acts as an alias for the SQL*Net V1 connect string, making it easy to access a server in a network that still uses SQL*Net V1....

After services have been defined in Oracle Names, *users only need to refer to them by name to connect to a database or network service*. The necessary exchange of details occurs between the requester and the Oracle Names Server.⁵⁶

⁵⁵ See OracleNamesAdminGuide at 5, 22, 52-54.

⁵⁶ *Id.* at 23-24.

Further, OracleNamesAdminGuide describes an Oracle7 system having multiple Names Servers functioning as a directory service storing addresses for all services on the network and making them available to clients wishing to make a connection.⁵⁷ As shown below in Figure 2-1, OracleNamesAdminGuide explains that an Oracle7 system supports multiple clients, databases, and Oracle Names database instances connected on a network.⁵⁸



OracleNamesAdminGuide discloses that servers include a “listener” application that listens for client requests and hands off requests to servers.⁵⁹

a) OracleNamesAdminGuide’s Names Servers

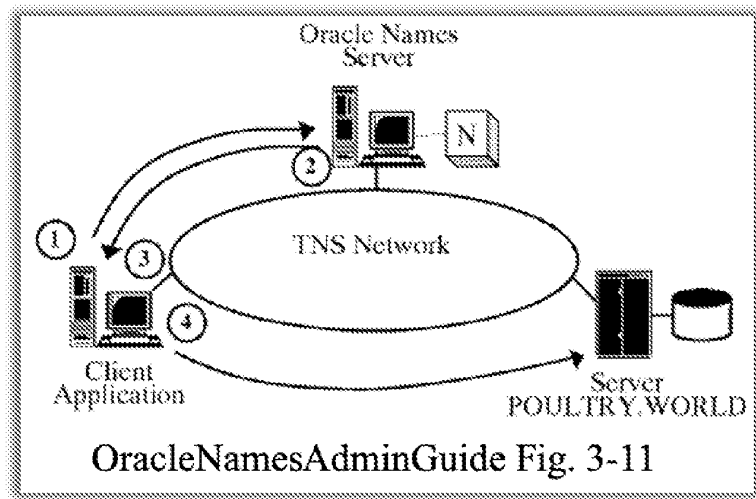
OracleNamesAdminGuide explains a “Names Server has a single purpose: to resolve, or assist in resolving, a client-initiated name request. It interprets the request, then looks up the name in its cache, or it calls a Names Server in another region. The response [is] passed back to

⁵⁷ See *id.* at 5, 22, 52-54.

⁵⁸ See *id.* as 32-33.

⁵⁹ See *id.* at 32-33, 70; Greene Decl. ¶¶ 50-51.

the client.”⁶⁰ As shown below in Figure 3-11, OracleNamesAdminGuide describes exchanging messages that include ① the client issues a data query to Names Server, ② the Names Server receives the request, looks up the location of the data, then sends the database connection and location information to the client, ③ the client uses the location information data to build a connection request to the desired database, and ④ the client contacts the database server containing the desired data.⁶¹



And OracleNamesAdminGuide explains that large installations benefit from having a central directory service that captures connection information in one system for all clients to use.⁶² OracleNamesAdminGuide teaches storing connection information (e.g., host name, port, and database instance) associated with a client-side alias identifier.⁶³

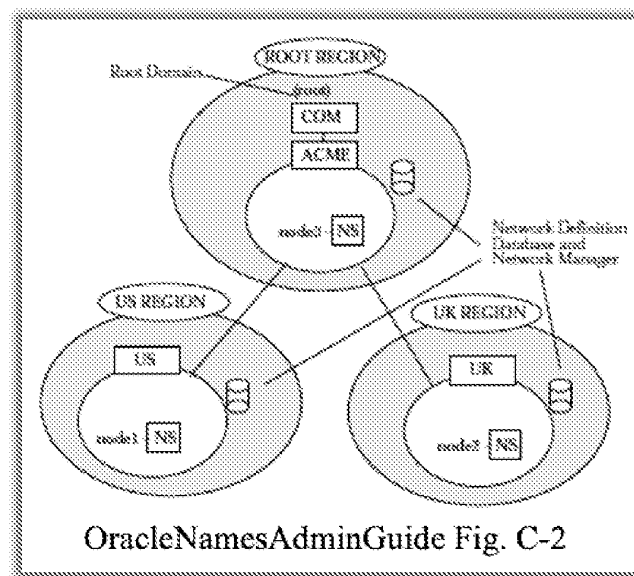
⁶⁰ OracleNamesAdminGuide at 52; *see also id.* at 202 (defining an Oracle Names Server as “[t]he name resolution software for efficiently answering name requests”).

⁶¹ *See id.* at 52-53.

⁶² *See id.* at 21-22.

⁶³ *See, e.g., id.* at 23, 52-53, 128-29.

Further, OracleNamesAdminGuide discloses various configurations of Names Servers acting as database directories for an Oracle7 system.⁶⁴ OracleNamesAdminGuide provides different distributed database configurations tailored to the particular needs of the Oracle7 user. For example, as shown below in Figure C-2, OracleNamesAdminGuide provides a hierarchical model in which the system delegates authority to two administrative regions authority controlled by a root region.⁶⁵ OracleNamesAdminGuide teaches that such an arrangement results in the respective US region's Names Servers containing different location information than the UK region's Names Server.⁶⁶



As discussed above in Section III.C, Patent Owner has asserted that the claim term “location server” means “a network-attached component that maintains a set of identifier/location mappings that are modified or returned in response to location request messages from clients.” A

⁶⁴ See *id.* at 178.

⁶⁵ See *id.* at 181-82.

⁶⁶ See *id.* at 185.

POSITA would have understood that OracleNamesAdminGuide teaches that each Names Server is a location server.⁶⁷ This is because each Names Server attaches to the network and maintains data location identifiers and returns those identifiers in response to query messages from clients. And OracleNamesAdminGuide explains that each Names Server is capable of modifying and updating data location identifiers.⁶⁸

*b) OracleNamesAdminGuide's SQL*Net Utility*

OracleNamesAdminGuide teaches that SQL*Net is Oracle's networking utility that enables remote data access between programs and databases. SQL*Net allows users to distribute applications and databases to different machines while continuing to communicate as if they were located on the same machine.⁶⁹ OracleNamesAdminGuide teaches that SQL*Net is an application that "works with the Oracle Server and enables two or more computers ... to exchange data through a network. SQL*Net supports distributed processing and distributed database capability. SQL*Net runs over and interconnects many communications protocols."⁷⁰ And OracleNamesAdminGuide teaches that, using SQL*Net, Names Servers communicate network address information to clients.⁷¹

⁶⁷ See Greene Decl. ¶¶ 52-54, 67-70.

⁶⁸ E.g., OracleNamesAdminGuide at 67 (explaining Names Servers store modified connect data), 101 (explaining administrator modifying data then refreshing Name Server).

⁶⁹ See *id.* at 29-30, 36, 53-54, 203.

⁷⁰ *Id.* at 203.

⁷¹ See *id.* at 20, 23, 52-53; Greene Decl. ¶ 55.

c) *OracleNamesAdminGuide' Location Information*

For a given data entity (e.g., a database), OracleNamesAdminGuide teaches that the entity's location information includes its host name,⁷² port, and database instance. OracleNamesAdminGuide explains that its Names Servers contain this location information to respond to a client's request to connect to a data entity.

As discussed above in Section III.C, Patent Owner has asserted that the claim term "location" means "an encoding that is a member of a set of associations with an identifier in a location server, and that specifies where data pertaining to the entity identified by the identifier is stored." Patent Owner has also asserted that the claim term "location information" means "information pertaining to one or more locations of data and/or the identities of one or more location servers."

OracleNamesAdminGuide explains that its SQL*Net uses location information to build a "connection descriptor," also called a connect string or dbstring. The client uses this information to establish a client-server connection.⁷³ OracleNamesAdminGuide explains that its connection descriptor is a "specially formatted description of the destination for a network connection. Connect descriptors are constructed using a set of keywords and values. They are mapped to *service names* to provide more convenient reference."⁷⁴ And OracleNamesAdminGuide explains that each "unique network location [is] used to identify a network object such as a database service,

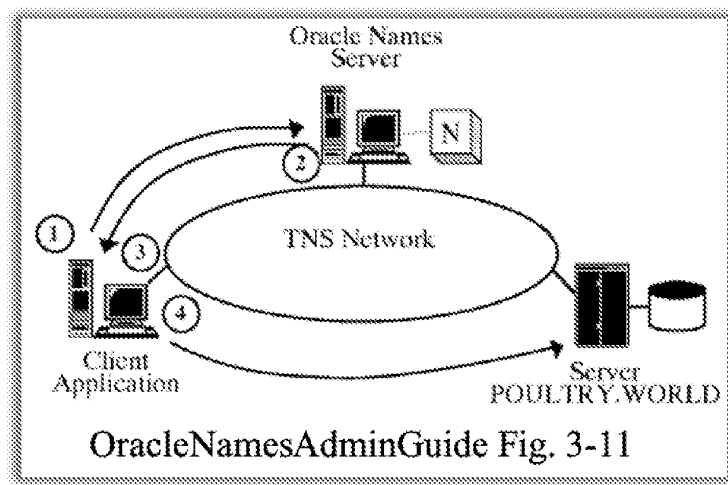
⁷² A POSITA would have understood that a "host name" may include a direct IP address or a host name that uses DNS to obtain an IP address. *See* Greene Decl. ¶ 56 n.35.

⁷³ *See* OracleNamesAdminGuide at 23 (explaining the prior version of Oracle Names used the term "connect string"), 100 (teaching prior Oracle versions refer to descriptors as "dbstring [which] is a database name defined in the Names Server").

⁷⁴ *Id.* at 200; *see also id.* at 202 (describing that a connect descriptor may include a system identifier ("SID"), which is a "unique name for an Oracle database instance.... The SID is included in the CONNECT DATA parts of the connect descriptors in a TNSNAMES.ORA file, and in the definition of the network listener in the LISTENER.ORA file").

client, Interchange, or Names Server. TNS addresses have a specific format. Addresses must be unique.”⁷⁵

As shown below in Figure 3-11, OracleNamesAdminGuide teaches that the client device sends a request for the POULTRY.WORLD database to the Names Server. The Names Server uses the identifier POULTRY.WORLD from the request to identify the location information (e.g., host name, port, and database instance) and sends this location information to the client. Using this location information, the client builds the connection descriptor associated with the POULTRY.WORLD database, then uses SQL*Net to make the connection.⁷⁶



Similarly, OracleNamesAdminGuide provides an exemplary request to the Names Server for connection and location information using “QUERY BONES.DEM.MEDICINE A.SMD” to the Oracle7 system.⁷⁷ Like the above POULTRY.WORLD example, OracleNamesAdminGuide

⁷⁵ *Id.* at 199; Greene Decl. ¶¶ 56-57.

⁷⁶ *See* OracleNamesAdminGuide at 52-53.

⁷⁷ *See id.* at 129.

teaches that the SQL*Net client sends a request to the Names Server for the desired database using the corresponding BONES.DEM.MEDICINE identifier. The Names Server uses the identifier BONES.DEM.MEDICINE from the request to identify the location information (e.g., host name, port, and database instance) and send this location information to the client. The client then uses this location information to connect to the server storing the BONES.DEM.MEDICINE database. OracleNamesAdminGuide details the resulting query metrics and location information sent to the client from the Name Server as follows:

Total response time:	0.04 seconds
Response status:	normal, successful completion
Authoritative answer:	yes
Number of answers:	1
Canonical name:	bones.dem.medicine
TTL:	1 day
Alias translations:	
from:	bones.dem.medicine
to:	bones.dem.medicine

Answers:

```
data type is 'a.smd'  
Syntax is ADDR: ... (DESCRIPTION= (ADDRESS= (COM-  
MUNITY=tcp) (PROTOCOL=TCP) (Host=cowboy)  
(Port=1522)) (CONNECT_DATA= (SID=rodeo)) [.]"78
```

This illustrates that OracleNamesAdminGuide's Names Server completed the response in an *a.smd* output format in 0.4 seconds, and labeled the response status as normal and successfully completed. And the Names Server responds (i.e., "answers") to the client's query by

⁷⁸ *Id.*

providing the location information for BONES.DEM.MEDICINE.⁷⁹ This location information includes the host name “cowboy,”⁸⁰ port 1522, and database instance “rodeo” along with the connection information (“protocol=TCP”).⁸¹

Using this location information, OracleNamesAdminGuide teaches that the client builds the connection descriptor associated with the BONES.DEM.MEDICINE database, then uses SQL*Net to connect to the database using one of the available network transport protocols, TCP/IP.⁸² And using the below annotated version of OracleNamesAdminGuide’s Figure 3-11, Mr. Greene demonstrates the sequence of events to resolve the client’s data query BONES.DEM.MEDICINE A.SMD.⁸³

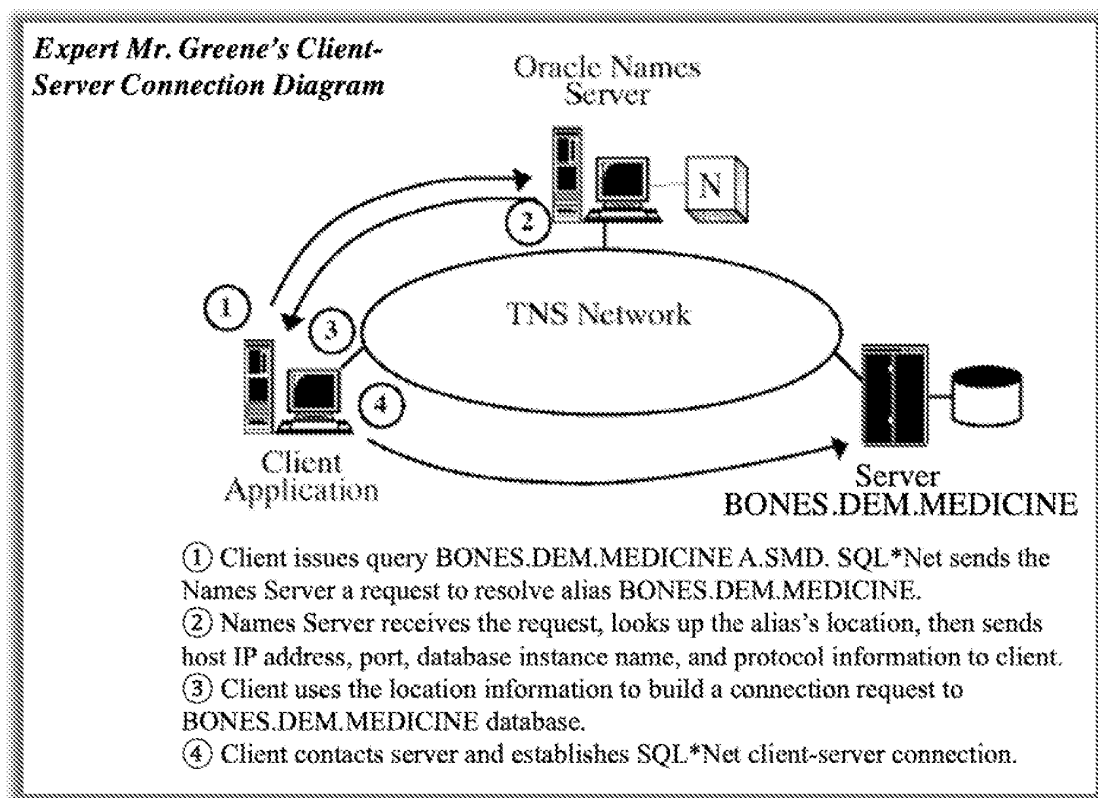
⁷⁹ *See id.*

⁸⁰ As Mr. Greene explains, the host name would be converted to an IP address using a DNS server before the connection string was compiled. Greene Decl. ¶¶ 58-60.

⁸¹ *See* OracleNamesAdminGuide at 129.

⁸² *See id.* at 39, 201 (describing other available transport protocols, like DECnet).

⁸³ Diagram from Mr. Greene’s Declaration. *See* Greene Decl. ¶ 61 (citing OracleNamesAdminGuide at 52, 129).



Based on these teachings, a POSITA would have understood that OracleNamesAdminGuide's connection descriptor is built using the claimed "location information" (i.e., a data encoding that specifies the storage location of data and that is associated with an identifier for the data (e.g., network-service-name-alias) in a Names Server).⁸⁴

d) OracleNamesAdminGuide's Identifier String

OracleNamesAdminGuide discloses associating location information (i.e., connection descriptor) to an identifier. As discussed above in Section III.C, Patent Owner has asserted the claim term "identification string" means "a unique encoding that identifies an individual entity, and with which zero or more location strings are associated in a location server."

⁸⁴ See Greene Decl. ¶ 63.

On the client-side of a client-server connection, OracleNamesAdminGuide uses a network-service-name-alias as an identifier string to easily refer to desired entities on the distributed database and to obtain connection information from the Names Server. For example, OracleNamesAdminGuide explains that aliases allow a user to “connect to a database or use a database link by using a simple name instead of having to specify the global database name or global database link name (that is, the fully-qualified name).”⁸⁵ As Mr. Greene explains, a POSITA would have understood that Oracle literature uses “alias” in other contexts, such as server-side aliases and object aliases.⁸⁶ Because those other types of aliases are different and distinct from aliases used on the client-side of an Oracle7 system, the Request and Mr. Greene refer to client-side identifiers as network-service-name-aliases.⁸⁷

As shown in the above example, OracleNamesAdminGuide’s client uses the network-service-name-alias BONES.DEM.MEDICINE to identify a specific database associated with location information (e.g., host name, port, database instance) in the Names Server. And as shown in that example, the network-service-name-alias BONES.DEM.MEDICINE refers to one, and only one, database in the Oracle7 system.⁸⁸

Based on these teachings, a POSITA would have understood that OracleNamesAdminGuide’s network-service-name-alias is the claimed “identifier string” because it is a unique en-

⁸⁵ OracleNamesAdminGuide at 101.

⁸⁶ See Greene Decl. ¶¶ 62-64.

⁸⁷ *Id.*

⁸⁸ As OracleSG explains, this single location address may correspond to a virtual address for multiple servers working together for load balancing. See OracleSG at 690-96.

coding that identifies an individual data entity (e.g., BONES.DEM.MEDICINE) and is associated with the location information (i.e., host name, port, and database instance used to build a connection descriptor) in a Names Server..

In an IPR proceeding concerning claims of the related '978 Patent, Patent Owner argued that the specification requires a “1-to-1 relationship between an identifier and an entity and [a] 1-to-many relationship between an identifier and locations.”⁸⁹ But Patent Owner’s argument lacks sufficient support in the disclosures of the '978 Patent or '640 Patent that would limit an entity to one, and only one, identifier, or that would require a one-to-many relationship between identifier and location. Patent Owner appears to base its 1-to-1 argument on the position that its construction’s use of the word “unique” means that it requires exclusivity. But this interpretation is inconsistent with the specification, which refer to multiple identification strings specifying an entity and that the relationships between identifiers, locations, and entities are dynamic:

The network distributed tracking wire transfer protocol includes two basic components: *identification strings for specifying the identity of an entity* in the distributed data collection, and location strings for specifying network locations of data associated with an entity. The protocol accommodates variable length identifier and location strings. *Relationships between identification strings and location strings can be dynamically and spontaneously manipulated thus allowing the corresponding data relationships also to change dynamically*, spontaneously, and efficiently.⁹⁰

⁸⁹ Ex. 20 (IPR2020-00276 Preliminary Resp.) at 26. The '978 Patent claims to be a continuation-in-part of the '640 Patent. *See* '978 Patent (Ex. 7) at Cover.

⁹⁰ '640 Patent at 2:15-24.

And indeed, in its IPR decision denying institution, the Board stated that it “disagree[s] based on the record before us with Patent Owner’s arguments that its construction requires as ‘a 1-to-1 relationship between an identifier and the entity that it is identifying.’”⁹¹

Yet even if Patent Owner’s 1-to-1 argument were correct, OracleNamesAdminGuide and OracleSG describe specific configurations in which an identifier identifies a single entity. For example, as shown in Figure 3-11, OracleNamesAdminGuide discloses that the identifier POULTY.WORLD identifies a single entity, the POULTRY.WORLD database.⁹² As another example, OracleNamesAdminGuide discloses that the identifier BONES.DEM.MEDICINE identifies a single entity, having location Host=cowboy, Port=1522, CONNECT_DATA=(SID=rodeo).⁹³

Similarly, even if Patent Owner’s 1-to-many argument were correct, OracleNamesAdminGuide and OracleSG describe specific configurations in which an identifier is associated with multiple locations. For example, OracleNamesAdminGuide and OracleSG explain the need for fault tolerance in an Oracle7 system in which the data entities are duplicated in the system.⁹⁴ In such a configuration, a POSITA would have understood an entity’s identifier has a 1-to-many relationship with two or more locations.

⁹¹ Ex. 21 (IPR2020-00276 Decision) at 12 (declining to construe “identifier,” “location,” and “location information” because decision didn’t turn on those constructions for ’978 Patent concerning art not at issue here).

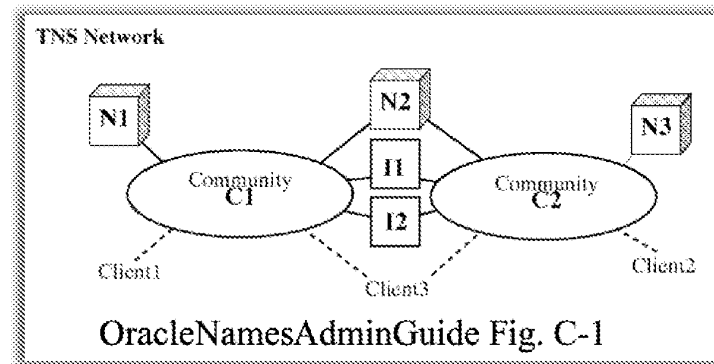
⁹² See OracleNamesAdminGuide at 52-53.

⁹³ See *id.* at 129.

⁹⁴ See *id.* at 32, 72; OracleSG at 152-56.

e) *OracleNamesAdminGuide's Redirect Messaging*

Using location and identifier strings, OracleNamesAdminGuide teaches that a client submits a query for the location of an entity (e.g., database) in an Oracle7 system having multiple Names Servers, such as in below Figure C-1. If Names Server N1 receives that query but lacks the location information to answer that query, OracleNamesAdminGuide teaches the option of N1 forwarding the request to N2. OracleNamesAdminGuide also teaches that it is often desirable to disable N1's forwarding function to lessen the workload of N1, especially when N1 is a performance bottleneck in cross-region request processing.⁹⁵ When the forwarding function is disabled, OracleNamesAdminGuide explains that N1 "simply tells the requestor the address of the Names Server that can answer the request."⁹⁶



Using this message from the N1 Names Server redirecting to another Names Server that can answer the request, a POSITA would have understood that the client then determines the location of another Names Server (N2 or N3) that has the requested location information.⁹⁷ And as shown in OracleNamesAdminGuide's Figure C-3, a POSITA would have understood the Usage

⁹⁵ E.g., OracleNamesAdminGuide at 137; see also *id.* at 119-20, 186-87.

⁹⁶ *Id.* at 137.

⁹⁷ See Greene Decl. ¶¶ 71-72.

Notes' warning about failure applies in the limited situation of a local client requesting data located in a foreign region that the client is unable to directly query.⁹⁸ Otherwise, all other Names Servers will return a redirect message to the client containing the address of another Names Server that can service the request, if the initial server lacks the information requested by the client.⁹⁹

2. OracleSG (Ex. 3) Overview

OracleSG is a printed publication authored by Mr. Greene.¹⁰⁰ On its face, OracleSG shows that Sams Publishing published OracleSG in 1995.¹⁰¹ Corroborating evidence establishes that Sams Publishing published OracleSG no later than December 31, 1995 in paper form.¹⁰² Mr. Greene confirms that, as of 1995, Sams Publishing “was an established publisher that was well known to those interested in the field of database technologies,” and he has “personal knowledge that Sams Publishing published physical copies of OracleSG no later than October 1995.”¹⁰³

⁹⁸ See OracleNamesAdminGuide at 186-87.

⁹⁹ See Greene Decl. ¶ 73.

¹⁰⁰ See *id.* ¶ 74.

¹⁰¹ See OracleSG at 27.

¹⁰² See Ex. 19 (Library of Congress catalog record showing 1995 publication date for OracleSG (Ex. 3)).

¹⁰³ Greene Decl. ¶ 74; see also *Vidstream LLC v. Twitter, Inc.*, 981 F.3d 1060, 1065 (Fed. Cir. 2020) (“A reference will be considered publicly accessible if it was disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art exercising reasonable diligence[] can locate it.... When there is an established publisher there is a presumption of public accessibility as of the publication date.”).

OracleSG is a printed publication that is prior art under 35 U.S.C. § 102(b) because it published more than one year prior to the application date of the '640 Patent and under § 102(a) because it published by others before that application date.¹⁰⁴

OracleSG describes the use of real products, namely Oracle versions 6 through 7.2 and related products. OracleSG “focuses on the job of the database administrator [and] describes Oracle technical topics... as background material to understand how to deal with problems and why things work the way they do.”¹⁰⁵ The Oracle7 system was “by its very nature, a complex system” and too vast to fit neatly into one volume. OracleSG helped address that problem.¹⁰⁶

Accordingly, OracleSG addresses key Oracle7 functionalities, including the distributed database itself, the SQL*Net utility, Oracle Names, queries, hashing, and optimization. OracleSG explains that “[p]art of the challenge faced by Oracle database administrators (DBAs) is to determine what parts of this Oracle world they must support and prepare themselves accordingly.”¹⁰⁷ OracleSG was designed to help meet that challenge, not by replacing the Oracle7 reference manuals, but as a companion piece to them. The book states: “[t]here are a number of things that change all of the time and you need these specific references. *Oracle DBA Survival Guide* provides the overall concepts and some of the glue that enables you to know which functions you should be performing and why.”¹⁰⁸

¹⁰⁴ Even if Patent Owner were able to establish that the '640 Patent is entitled to the benefit of the earliest possible priority date of July 8, 1998, OracleSG is §§ 102(a) and (b) prior art based on its publication date.

¹⁰⁵ OracleSG at 20.

¹⁰⁶ *Id.* at 21.

¹⁰⁷ *Id.* at 27.

¹⁰⁸ *Id.* at 20.

OracleSG explains that “the burden of providing the translation between the easy names and the exact server and instance addresses falls on the DBA.”¹⁰⁹ OracleSG emphasizes that it is important for a POSITA to keep up to date on new features and functions to ensure their system is working as well as possible, specifically referencing the significance of Oracle Names.¹¹⁰

Further, OracleSG provides insight into specific areas of technology. First and foremost is the relational database management system itself, which is designed “to store and retrieve information in a variety of formats.”¹¹¹ OracleSG also teaches the importance of secondary products and features relating to the Oracle RDBMS: “[Oracle7] implements a series of background processes to perform part of the work of storing and retrieving information. This also helps to reduce the complexity of each individual process and increase the speed of processing, especially on computers with multiple CPUs.”¹¹²

Referencing the below Figure 46.5, OracleSG teaches that SQL*Net is installed on each device in the system in order to connect the devices, clients, and servers.¹¹³ OracleSG explains that Oracle7 included the upgraded SQL*Net v2.0 “to provide better services to client-server environments.”¹¹⁴ As OracleSG teaches, SQL*Net v2.0 interfaced to the multi-threaded server to reduce processing capacity for idling users reading their outputted results to improve client-

¹⁰⁹ *Id.* at 87.

¹¹⁰ *See id.*; Greene Decl. ¶¶ 75-78.

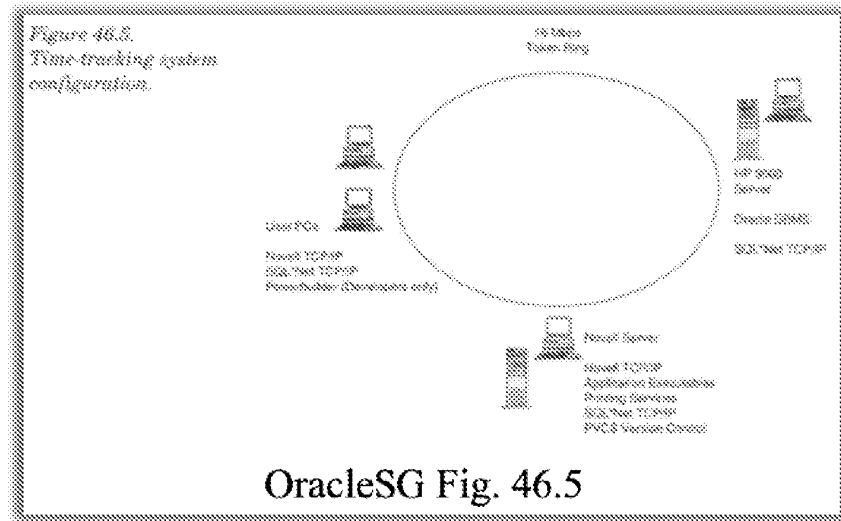
¹¹¹ OracleSG at 27.

¹¹² *Id.*

¹¹³ Greene Decl. ¶ 79.

¹¹⁴ OracleSG at 648.

server performance.¹¹⁵ The query remained a key function of Oracle7 and SQL enables those queries to locate data files.¹¹⁶



OracleSG further provides ways to improve the efficiency of the system, including the well-known use of hashing in database queries and the growingly popularity of “parallel everything.”¹¹⁷ As described in OracleSG, Oracle7 systems include rule-based and cost-based optimizers that list ways to improve the efficiency and functionalities available in database tables.¹¹⁸ OracleSG explains that these functionalities, such as hashing, can be used to ameliorate common concerns facing databases, including high transaction volume, disk space issues, and slow response times.¹¹⁹

¹¹⁵ See *id.*; see also *id.* at 172-73 (describing improved listener application and multi-thread server processes of SQL*Net version 2).

¹¹⁶ See *id.* at 152; Greene Decl. ¶¶ 79-81.

¹¹⁷ See OracleSG at 152, 583-84.

¹¹⁸ See *id.* at 633, 636.

¹¹⁹ See *id.* at 675.

As Mr. Greene explains, OracleSG provides the reader with these key features of the Oracle7 system and explains the considerations necessary to maximize its benefits.¹²⁰ As such, OracleSG acts as a conduit for the reader to piece together the various functions and features and implement the full-range of capabilities contained within the single product.¹²¹

3. McGarvey (Ex. 4) Overview

McGarvey is a U.S. Patent issued July 7, 1998 based on an application filed December 19, 1995.¹²² McGarvey is prior art under at least 35 U.S.C. §§ 102(a) and (b) because it is a patent issued more than one year prior to the application date of the '640 Patent.¹²³

McGarvey describes a distributed network configuration that uses well-known name resolution techniques.¹²⁴ McGarvey discloses a TCP/IP network having names servers 115-116 and client hosts 101-103 request network addresses from the names servers, as shown below in Figure 1.¹²⁵

¹²⁰ See Greene Decl. ¶¶ 79-87.

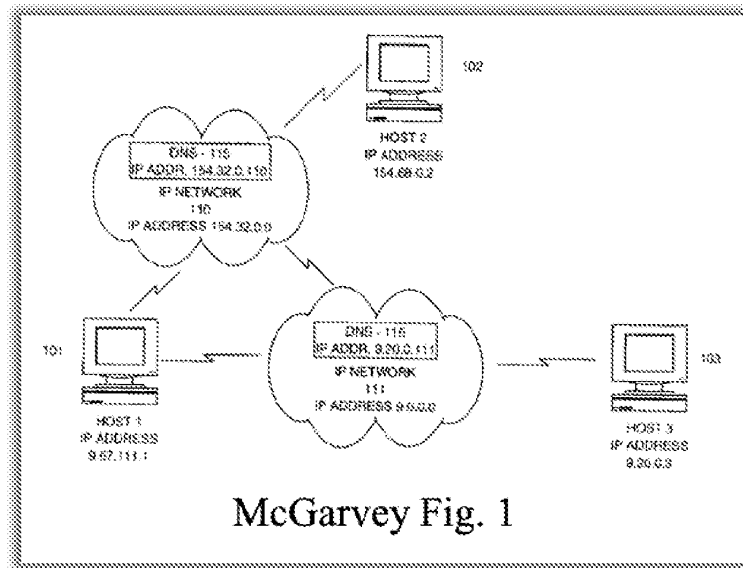
¹²¹ See *id.* ¶ 88.

¹²² See McGarvey (Ex. 4) at Cover.

¹²³ Even if Patent Owner were able to establish the '640 Patent is entitled to the benefit of the earliest possible priority date of July 8, 1998, McGarvey is § 102(a) prior art based on its issue date and § 102(e) prior art based on its filing date.

¹²⁴ See *id.* at Abstract, 1:40-49, 2:61-67.

¹²⁵ See *id.* at 3:60-63.



As discussed in the reference's Background of the Invention, McGarvey teaches that it was known in the art for a client to send a location request to a first names server. If the first names server doesn't have the requested location information, it returns a redirect message to the client. Based on that redirect message, the client sends its request to a second names server that returns the requested location information:

Address resolution must be done before a TCP/IP connection can be established with the destination host. The name resolution information may be stored locally in a hosts file on each host, but this approach is not practical for large networks. ***More commonly, this information is stored in the name resolution master file of a domain name server....***

A domain name consists of a label or of several labels connected by periods. ***Domain names are organized in a hierarchical tree of subdomains.*** For example, domain bbbbbb.aaaaa is a subdomain of aaaaa, and domain ccccc.bbbbbb.aaaaa is a subdomain of bbbbbb.aaaaa....

A particular name server may contain information on only a subset of the domain namespace, but the name server typically belongs to a hierarchy of name servers,

and if it does, either it communicates with other members of the hierarchy to attempt to resolve host names, *or it returns information referring the client to another server in the hierarchy, so that the client can redirect the query.*¹²⁶

In addition to teaching known name resolution techniques, McGarvey discloses an embodiment in which client 101 issues a location query to names server 115. If the server responds that the location is unknown, the client searches locally for the address of the location.¹²⁷

4. Rajani (Ex. 5) Overview

Rajani is a U.S. Patent issued April 5, 1994 based on an application filed January 2, 1991.¹²⁸ Rajani is prior art under at least 35 U.S.C. §§ 102(a) and (b) because it is a patent issued more than one year prior to the application date of the '170 Patent.

As Mr. Greene explains, Rajani teaches how to “quickly ascertain” the location of a file stored in a hierarchical memory configuration using a table of file names:

[E]ach time a file is stored in the backup memory its storage location in the backup memory as well as its name and version number are stored in a Table of file names. Thus, the location of a file stored in the back up memory can be quickly ascertained by referring to the Table of file names.¹²⁹

Mr. Greene further details how Rajani explains that using unique identifiers associated with file locations in a table of names helps to quickly locate files in a hierarchical system.¹³⁰

¹²⁶ *Id.* at 1:13-48.

¹²⁷ *See id.* at 5:3-32; Greene Decl. ¶¶ 90-92.

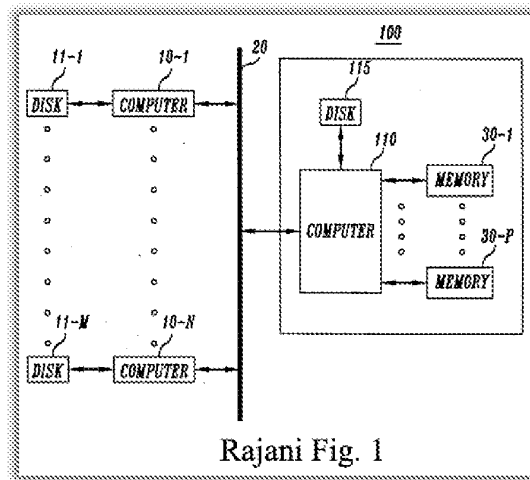
¹²⁸ *See* Rajani (Ex. 5) at Cover.

¹²⁹ Rajani at 1:29-34; Greene Decl. ¶ 94.

¹³⁰ Rajani at Abstract; Greene Decl. ¶ 95.

And Rajani explains that it was known in the art that locating data in such hierarchical configurations can be labor intensive, even for a table of file names.¹³¹ Rajani explains that, as the system grows, the number of entries and length of each identifier grows to compound the intensive labor required to find location data entries on the table of file names.¹³² Accordingly, Rajani describes the benefits of using hashing for the table of file names to “quickly ascertain” the location under these conditions.

Specifically, Rajani is directed to indexing location information for memory that is distributed over a network. As shown below in Figure 1, Rajani describes a data network 20 that connects multiple computers 10-1 to 10-N having its own respective memory storages 30-1 to 30-P.¹³³



Rajani describes using a hash function to organize data in storage across memory (30-1 to 30-P) and to determine the location of the data in memory. Specifically, Rajani teaches using a

¹³¹ Rajani at 1:18-23; Greene Decl. ¶ 96.

¹³² Rajani at 1:37-45; Greene Decl. ¶ 97.

¹³³ See *id.* at 1:6-9, 2:7-43; Greene Decl. ¶ 98.

hash function to convert each file's hierarchical pathname into a unique identifier, which is used to identify the file's location in the network:

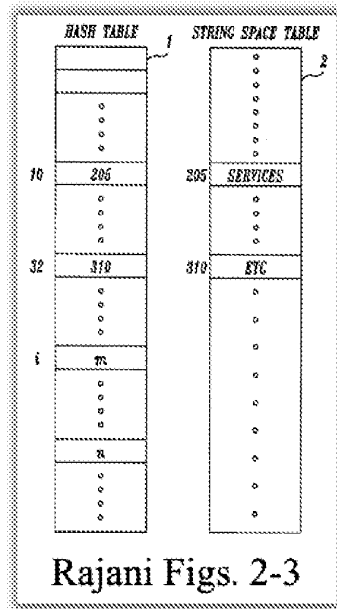
[A] method comprising the steps of ... storing each of said components in an available location of a first table of locations such that the address of said available location defines the unique identity value of the component stored thereat, and deriving for each of said components an associated hashing number and storing in a second table of locations at a location identified by said hashing number the address of the available location ¹³⁴

And as shown below in Figures 2-3, Rajani describes using a hash table to associate a network location to a file entity:

[R]eferring now to FIG. 2 and 3, computer 110 employs in the process of locating or creating a record respectively associated with a file that is either being stored or un-loaded from one memories 30-1 through 30-P a so-called hash Table (FIG. 2) and string space Table (FIG. 3). In the following discussion it will be assumed that a file is being stored in one of the memories 30-1 through 30-P, for example, memory 30-1, and that the path name of the file is /etc/services. It is also assumed that the originator of the file is computer 10-1. (It is to be understood of course that the ensuing discussion equally pertains to each of the computers 10 and equally pertains to a file having a different path name, e.g., /usr/data1, or a longer pathname, i.e., /usr/etcMbin/services.).¹³⁵

¹³⁴ Rajani at 8:53-60, Claim 1.

¹³⁵ *Id.* at 4:1-15.



To do so, Rajani teaches using a unique index (e.g., S-index) to facilitate the hashing process and ease the locating burden:

Upon receipt of the file, or path name /etc/services, computer 110 maps the path name into the aforementioned unique identity which may be used to locate the file after it has been stored in memory 30-1. That is, computer 110 first converts each component of the pathname, i.e., etc and services, in accord with an aspect of the invention, into a unique identification comprising a predetermined number of bytes-illustratively four bytes. I call this unique identification an S index. In the present illustrative example of the invention, the S index for the components “etc” and “services” are assumed to be, for example, 310 and 205, respectively. Computer 10 generates each S index by first passing each component of the pathname through a conventional hashing algorithm to obtain a memory offset (i.e., $h = \text{hash}(\text{“string component”}) = i$). It is assumed herein that memory offsets of 32 and 10 are obtained as a result of respectively passing “etc” and then “services” through such a hashing algorithm.¹³⁶

¹³⁶ *Id.* at 4:16-34.

Further, Rajani teaches a respective computer will use the hash table to locate a file on the memory drive based on the S-index:

Computer 110 then retrieves from hash Table 1 the contents of the location defined by the offset of 32, which contents in the present illustrative example should be the S index for the component “etc”, i.e., the value of 310 as shown in FIG. 2... Computer 110 then similarly verifies that the value of 205 is the S index for the component “services”, as shown at location 10 of hash Table 1. Thus, the S index for a component in a pathname is the memory location in string space Table 2 containing that component.¹³⁷

While it shows using a hashing table to locate a preferred type of entity (e.g., a backup file), Rajani teaches a POSITA would be able to devise similar uses of hashing to achieve similar results based on the particular needs of the arrangement and file structure:

The foregoing is merely illustrative of the principles of the invention. Those skilled in the art will be able to devise numerous arrangements, which, although not explicitly shown or de-scribed herein, nevertheless embody those principles that are within the spirit and scope of the invention. For example, special mapping techniques would allow the D-node and H-node Tables to be combined into one Table for each of the computers 10. In addition, the collision among hashing numbers could be dealt with using a so-called secondary hashing technique that is known in the art.¹³⁸

E. SNQ 1: Independent Claims 17 and 18 Are Obvious Based on Oracle-NamesAdminGuide in view of OracleSG and McGarvey.

Claims 17 and 18 recite a system having a data repository storing data and a location server storing the data’s location information wherein a client may request the location of data

¹³⁷ *Id.* at 4:40-56.

¹³⁸ *Id.* at 6:26-36; Greene Decl. ¶¶ 98-103.

using an identifier associated with the entity's location information. Claims 17 and 18 also recite using a redirect message to redirect a client to another location server if the initial location server lacks the location information responsive to the client's query.

OracleNamesAdminGuide discloses a system having data repositories and location Names Servers storing location information wherein a client may request the location of data using a network-service-name-alias (i.e., identifier) associated with the data's host name, port, and database instance used to build a connection descriptor (i.e., location information).¹³⁹ OracleNamesAdminGuide further teaches or renders obvious using a redirect message to redirect a client to another location server if the initial location server lacks the location information responsive to the client's query.

To the extent OracleNamesAdminGuide doesn't disclose all elements of claims 17 and 18, OracleSG renders obvious associating an identifier to location information and McGarvey obvious using a redirect message to redirect a client to another location server.

1. Reasons to Modify OracleNamesAdminGuide in View of OracleSG and McGarvey.

- a) OracleSG's teachings render obvious associating an identifier string with location information.*

As the above overviews show, OracleNamesAdminGuide and OracleSG are in the same field of art and discuss distributed database systems using Oracle7 technology, and its associated tools and utilities.¹⁴⁰ They contain overlapping disclosures focused on the similar purposes of improving query efficiency and resolution of user access.

¹³⁹ See OracleNamesAdminGuide at 28, 32, 36.

¹⁴⁰ See, e.g., *id.* at 5, 22, 52-54; OracleSG at 20, 27, 695; Greene Decl. ¶¶ 101-104.

Before the '640 Patent, it was well known and ubiquitous for computer network systems to associate location information with identifiers. With respect to Oracle7 systems like those in OracleNamesAdminGuide, OracleSG teaches that SQL*Net associates the data's network address to an "easy name" synonym:

A synonym is merely a way of assigning an alias with which users can reference a table or view. ...Synonyms simplify this access because you can create a simple synonym that contains the details of a complex reference string.¹⁴¹

Oracle Network Manager ... allows you to generate the network configuration files for SQL*Net version 2. This version makes it much easier for users to specify a database to connect to; however, the burden of providing the translation between the easy names and the exact server and instance addresses falls on the DBA [database administrator]. I would not recommend configuring SQL*Net version 2 without this utility.¹⁴²

As the below claim charts detail regarding claim elements 17.b, 17.d-17.e, 18.b, 18.c, and 18.e-18.g, a POSITA would have found it obvious for OracleNamesAdminGuide to associate location information (i.e., connection descriptor) with an identifier string (i.e., network-service-name-alias). As OracleSG explains, OracleNamesAdminGuide's SQL*Net uses executable code to make it easier for users to connect to various databases and find their data; however, "the burden of providing the translation between the easy names and the exact server and instance address falls on the DBA."¹⁴³ Making use of identifier strings to improve the users' experience and to ease their ability to find their desired data is a primary objective of an Oracle database administrator ("DBA"). As Mr. Greene identifies, the benefits of associating data with an identifier

¹⁴¹ See OracleSG at 189-90.

¹⁴² *Id.* at 86-87; Greene Decl. ¶ 105.

¹⁴³ See OracleSG at 87.

string are directly perceived by the end user, and a DBA would have been motivated to simplify the users' experience in this manner by providing an association between unique customer identifiers and the location information of the desired data.¹⁴⁴

Indeed, before the '640 Patent, POSITAs were combining these teachings in Oracle7 systems in the 1990s, and OracleSG includes techniques explaining to DBAs how to improve their systems.¹⁴⁵ And OracleSG directs DBAs to combine OracleSG's teachings with various reference manuals and resources to take advantage of all of Oracle7's capabilities. For example, to use SQL*Net effectively, OracleSG provides that a DBA should record error messages and consult "specific product manuals (for example, SQL*Net)" for solutions.¹⁴⁶ As Mr. Greene concludes, maintaining a working knowledge of the state of the art would be imperative for managing a distributed system, especially an Oracle7 system, because a POSITA would have been expected to combine the system with related technologies to improve efficiency and optimization.¹⁴⁷

For those reasons, a POSITA would have been motivated to associate an identifier string with OracleNamesAdminGuide's data and location information for use in optimizing query efficiency, based on OracleSG's teachings. As OracleSG explains, a POSITA was often asked to help improve query efficiency during the relevant timeframe:

You may be regarded by your project team as the database guru because you have the title of DBA [database administrator]. If this is the case, try to obtain as much knowledge as possible about the DBMS [database management system] and query

¹⁴⁴ See Greene Decl. ¶¶ 105-107.

¹⁴⁵ See OracleSG at 20.

¹⁴⁶ See *id.* at 605; Greene Decl. ¶¶ 105-107.

¹⁴⁷ See OracleSG at 44, 52; Greene Decl. ¶¶ 105-107.

optimization through training courses and books. It may mean more work for you, but it could save many hours of project team labor if you can quickly provide your compatriots with the answers that they need.¹⁴⁸

As Mr. Greene concludes, based on the commonplace need for efficiency in running queries, a POSITA would have been motivated to associate the identifier string with data and location information because this allows OracleNamesAdminGuide's Names Server to use executable code to quickly return the location information associated with the identifier string to the requesting client with minimized cost.¹⁴⁹

b) McGarvey's teachings render obvious redirect messaging.

As the above prior art overviews show, McGarvey is in the same field as OracleNamesAdminGuide and OracleSG. For example, as discussed above in Section III.D.1, OracleNamesAdminGuide describes a distributed database system, focusing on Names Servers that respond to a client's data requests using unique customer identifiers associated with the data's network address.¹⁵⁰ Similarly, McGarvey describes a distributed network configuration that uses well-known name resolution techniques.¹⁵¹

Before the '640 Patent, it was well known and ubiquitous for computer network systems to use redirect messaging when a names server lacks the requested location information. For example, McGarvey teaches a "particular name server may contain information on only a subset of the domain namespace, but the name server typically belongs to a hierarchy of name servers, and

¹⁴⁸ OracleSG at 51; Greene Decl. ¶ 106.

¹⁴⁹ See OracleSG at 51; Greene Decl. ¶ 107.

¹⁵⁰ See OracleNamesAdminGuide at 5, 22-24, 52-54.

¹⁵¹ See McGarvey at Abstract, 1:40-49, 2:61-67.

if it does, ... it returns information referring the client to another server in the hierarchy, so that the client can redirect the query.”¹⁵²

As the below claim charts detail regarding claim elements 17.g and 18.h toward redirect messages, a POSITA would have found it obvious for OracleNamesAdminGuide’s first Names Server—if it lacks the requested location information—to send a redirect message to the client containing redirect information for the client to calculate the location of a different Names Server having the location information. A POSITA would have been motivated to do so because this would further the use of OracleNamesAdminGuide’s SQL*Net and Names Servers to respond to certain data requests and lessen the workload of the root Names Server because it no longer needs to perform forwarding functions to other servers containing the requested information.¹⁵³ Further, if a Names Server can’t resolve a client’s information query, a POSITA would have been motivated to have the server transmit a redirect message so the client can calculate the location of another Names Server housing the location information, as Mr. Greene explains.¹⁵⁴ This is because using such redirect messages was an established method in hierarchical naming services to address unresolved location requests.¹⁵⁵

Moreover, a POSITA would have been motivated to do so because providing a redirect message to calculate the location of a different Names Server would improve the ability of OracleNamesAdminGuide’s system to locate the client’s requested information. By taking each of these steps, a POSITA would have designed OracleNamesAdminGuide’s system to detect and

¹⁵² *Id.* at 1:13-48; Greene Decl. ¶¶ 108-109.

¹⁵³ *E.g.*, OracleNamesAdminGuide at 33, 61, 137; Greene Decl. ¶¶ 108-109.

¹⁵⁴ *See* Greene Decl. ¶¶ 108-109.

¹⁵⁵ *See* McGarvey at 2:61-67, 5:3-32.

resolve problems without placing the burden on the user. This is the DBA's inherent role, and the clear benefits for the user would have motivated a POSITA to combine OracleNamesAdminGuide and McGarvey in this way, as Mr. Greene explains.¹⁵⁶

Further, it was known in the art that a hierarchical structure in a larger database system would reduce processing demands on each Names Server by dividing the directory into separate portions. This would have resulted in the Names Server redirecting a request to the respective Names Server containing the pertinent location information.¹⁵⁷ Both McGarvey and OracleNamesAdminGuide discuss name resolution approaches in networks that may rely on TCP/IP and domain name models.¹⁵⁸ A key role in database management is to “to detect problems before they impact users.”¹⁵⁹ Far from a new development in the art, redirect messages were commonly used for resolution by those skilled in the art, and OracleSG even referred to redirects as a useful tool for Oracle systems.¹⁶⁰

2. OracleNamesAdminGuide in View of OracleSG and McGarvey Renders Obvious Independent Claim 18.

During the original examination, the Examiner indicated that the prior art of record didn't teach or renders obvious the “redirect message” limitation of claims 17-18. OracleNamesAdminGuide teaches this limitation because it discloses that, if a first Names Server doesn't have the requested location information, it returns a redirect message to the client. Based on that redirect message, OracleNamesAdminGuide teaches that the client sends its request to a second Names

¹⁵⁶ See Greene Decl. ¶¶ 108-109.

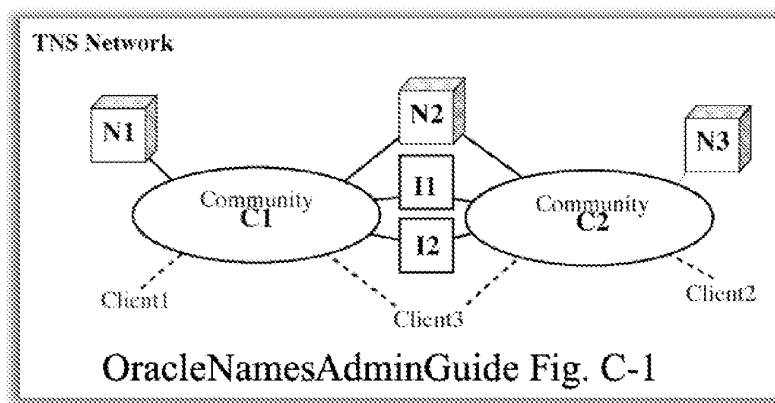
¹⁵⁷ See *id.* ¶¶ 108-109.

¹⁵⁸ See McGarvey at Abstract, 2:61-67; OracleNamesAdminGuide at 56-57.

¹⁵⁹ OracleSG at 44.

¹⁶⁰ See *id.* at 112; Greene Decl. ¶ 109.

Server that returns the requested location information.¹⁶¹ For instance, as shown below in Figure C-1, OracleNamesAdminGuide discloses a system having multiple Names Servers (N1 to N3). OracleNamesAdminGuide teaches the option of N1 forwarding a client's request to N2 if N1 lacks the location information to respond to the request. But OracleNamesAdminGuide also teaches that it is often desirable to disable N1's forwarding function to lessen the workload of N1, especially when N1 is a performance bottleneck in cross-region request processing.¹⁶² When the forwarding function is disabled, OracleNamesAdminGuide explains that N1 "simply tells the requestor the address of the Names Server that can answer the request."¹⁶³



As Mr. Greene explains, a POSITA would have understood that this message from N1 to the client is a redirect message used by the client to calculate the location of a different Names Server (e.g., N2) that contains the location information.¹⁶⁴

¹⁶¹ *E.g.*, OracleNamesAdminGuide at 137.

¹⁶² *E.g.*, *id.*

¹⁶³ *Id.*

¹⁶⁴ *See* Greene Decl. ¶¶ 110-111.

Similarly, McGarvey teaches that it was known in the art for a client to send a location request to a first names server. If the first names server doesn't have the requested location information, it returns a redirect message to the client. Based on that redirect message, the client sends its request to a second names server that returns the requested location information.¹⁶⁵ Based on McGarvey's teachings, a POSITA would have found it obvious that, if OracleNamesAdminGuide's first Names Server lacks the requested location, the first Names Server would transmit a redirect message containing redirect information to the client to calculate a location of a different Names Server that has the location information.¹⁶⁶ As Mr. Greene explains, a POSITA would have been motivated to do so because this would further the use of SQL*Net and Names Servers to respond to certain data requests, and further OracleNamesAdminGuide's goal of lessening the workload of the root server by no longer needing to perform forwarding functions to other servers containing the requested information.¹⁶⁷

The below claim chart shows that OracleNamesAdminGuide in view of OracleSG and McGarvey discloses or renders obvious each element of independent claim 18.¹⁶⁸

¹⁶⁵ See McGarvey at 1:13-48.

¹⁶⁶ See *id.*; Greene Decl. ¶ 112.

¹⁶⁷ See OracleNamesAdminGuide at 33, 61, 137; Greene Decl. ¶ 112.

¹⁶⁸ To establish obviousness, there is no requirement that the prior art be enabling. See *ABT Sys., LLC v. Emerson Elec. Co.*, 797 F.3d 1350, 1360 (Fed. Cir. 2015); *Symbol Tech., Inc. v. Opticon, Inc.*, 935 F.2d 1569, 1578 (Fed. Cir. 1991). But even if the prior art references did have to be enabling, both OracleNamesAdminGuide and OracleSG describe Oracle7 systems which were fully functional and operational products that existed before the '640 Patent. Manuals, and specifically software manuals, have consistently been found to be enabling, even when there are differences between the manual and the related product. See *Stored Value Sols., Inc. v. Card Activation Techs., Inc.*, 499 F. App'x 5, 15 (Fed. Cir. 2012); *Microstrategy Inc. v. Bus. Objects Americas*, 410 F. Supp. 2d 348, 362-63 (D. Del. 2006), *aff'd*, 238 F. App'x 605 (Fed. Cir. 2007).

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.preamble] A system for retrieving data location information for data stored in a distributed network, the system comprising:	To the extent the preamble is limiting, OracleNamesAdminGuide discloses an Oracle database system for retrieving location information for data stored in a distributed network. For instance, OracleNamesAdminGuide describes an Oracle7 system that provides user access to a distributed database. OracleNamesAdminGuide teaches the system components include SQL*Net functionalities that allow users to request data that is located in the system using Oracle Names servers. <i>E.g.</i>, OracleNamesAdminGuide at 5, 32. Further, the below analysis regarding claim elements 18.b-18.e shows how OracleNamesAdminGuide retrieves data location information for data stored in a distributed network, and is incorporated herein. <i>See</i> OracleNamesAdminGuide at 5: “Oracle Names product ... makes network address and database link information in an enterprise-wide distributed database network available to all nodes throughout the network.” <i>See</i> OracleNamesAdminGuide at 82: “Network Manager allows information that is used across your network in several files to be entered only once, and allows changes in configuration data for one component to “ripple through” and cause changes in other related component information. Network Manager also enables you to use control commands in Oracle Names Control Utility (NAMESCTL).” <i>See</i> OracleNamesAdminGuide at 32: “With the Dynamic Discovery Option, a network using the Oracle Names Service consists of: • one or more Oracle Names Servers... • one or more databases... • one or more clients[.]” <i>See</i> OracleNamesAdminGuide at 32:

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.preamble] A system for retrieving data location information for data stored in a distributed network, the system comprising: <i>continued</i>	Figure 2 – 1 Components of a Network Using Oracle Names with the Dynamic Discovery Option Based on the above teachings, OracleNamesAdminGuide discloses a system for retrieving data location information for data stored in a distributed network.¹⁶⁹
[18.a] a data repository configured to store data,	OracleNamesAdminGuide discloses this claim element. For instance, as shown below in Figure 2-1, OracleNamesAdminGuide teaches its system has multiple databases for storing data. As Mr. Greene explains, a POSITA would have understood that each of OracleNamesAdminGuide's databases is a data repository configured to store data.¹⁷⁰ See OracleNamesAdminGuide at 32: "... [A] network using the Oracle Names Service consists of: • one or more Oracle Names Servers We recommend that you set up at least two Names Servers for fault tolerance.... • <i>one or more databases</i> • one or more listeners • one or more clients Figure 2–1 shows the components described above with the Dynamic Discovery Option enabled.

¹⁶⁹ See Greene Decl. at pp.53-54.

¹⁷⁰ See *id.* at pp.55-57.

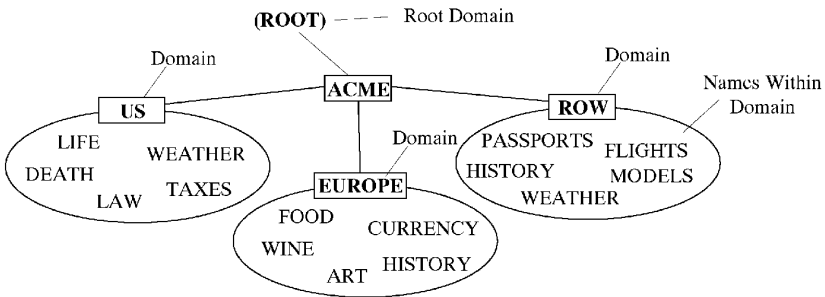
Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.a] a data repository configured to store data, <i>continued</i>	Figure 2 – 1 Components of a Network Using Oracle Names with the Dynamic Discovery Option See OracleNamesAdminGuide at 28: “Oracle Names 1.0 required administrators to enter the names and addresses of all databases in the network into Oracle Network Manager. Network Manager then created a network definition which it <i>stored in a database</i>.” See OracleNamesAdminGuide at 51: “For a given network, all of these objects are <i>stored together in the Network Manager network definition database</i>.” See OracleNamesAdminGuide at 30: “Once you have started DDO [i.e., dynamic discovery option], the Names Server remembers the addresses <i>of all databases</i>.” See OracleNamesAdminGuide at 36: “Each administrative region has: • Oracle Network Manager—This tool writes all definitions for all of Oracle’s network software, including SQL*Net, the MultiProtocol Interchange, and Oracle Names to the network definition on a database. It provides a structured way to configure the product initially, and to periodically update the network configuration. • one or more Oracle Names Servers • <i>one or more databases</i>, listeners and clients Figure 3 – 1 shows the relationship of the components described above with a single administrative region.” See OracleNamesAdminGuide at 36:

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.a] a data repository configured to store data, <i>continued</i>	Figure 3 – 1 Components of a Network Using Oracle Names Without the Dynamic Discovery Option See OracleNamesAdminGuide at 99: “A global database link is a link that connects <i>each database in a network to all other databases</i>. This link is created automatically by the Network Manager for use with Oracle Names. Each global database link created by Network Manager connects you to every other database server in the network.” See OracleNamesAdminGuide at 90: “... NAMES.ORA ... file contains control parameters for the Names Server and points the Names Server to <i>the database where the network definition is stored</i>.” Based on the above teachings, OracleNamesAdminGuide discloses a data repository configured to store data.¹⁷¹
[18.b] wherein the data is associated with an identifier string;	OracleNamesAdminGuide discloses this claim element or renders it obvious alone or in view of OracleSG. For instance, OracleNamesAdminGuide teaches that data stored in its databases is located at a network address associated with a network-service-name-alias, which is an identifier string. As explained above in Section III.D.1.d, a POSITA would have understood that such an alias is the claimed “identifier string” because it is a unique encoding that identifies an entity (e.g., database) and that is associates with a connection descriptor (i.e., location information) in a Names Server.¹⁷² OracleNamesAdminGuide explains that using such aliases allow customers to request data from the database using a simple name rather than a

¹⁷¹ See Greene Decl. at pp.55-57.

¹⁷² See *id.* at pp.57-59.

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.b] wherein the data is associated with an identifier string; <i>continued</i>	lengthy network location address: “Oracle Names makes network address and database link information available to all nodes throughout the network. <i>Each database server’s network address is stored with a name that is used to identify it. Client applications can then request a database connection with a simple name rather than a lengthy address.</i>” See OracleNamesAdminGuide at 20. See OracleNamesAdminGuide at 23: “An Oracle Names Server stores names and addresses for network services such as databases or MultiProtocol Interchanges, database link definitions, and object aliases. The Names Server stores the following types of objects, which are described below: • <i>database service names—The service name is the global database name that is mapped to the SQL*Net connect descriptor.</i> The combination of a listener address and connect data make up a connect descriptor that locates a database service. Gateways to non-Oracle databases and Oracle RDB databases are also stored. • <i>database links—Global database links are automatically created to each defined database from all database servers in the network. These links are made available to all users.</i> • <i>aliases—You can assign service name aliases as alternate names for any defined database service or database link. Service name aliases are resolved as if they were the original object.</i> • <i>V1 connect strings—A V1 connect string can be mapped to a service name in the Oracle Names database. The service name you enter acts as an alias for the SQL*Net V1 connect string,</i> making it easy to access a server in a network that still uses SQL*Net V1.” See OracleNamesAdminGuide at 101: “[A] network administrator could create aliases so that users can connect to a database or use a database link by using a simple name instead of having to specify the global database name or global database link name (that is, the fully-qualified name).” See OracleNamesAdminGuide at 128-129: “Alias name (sometimes referred to as “canonical name”).... Canonical name: bones.dem.medicine Alias translation[] ... bones.dem.medicine[.]” See OracleNamesAdminGuide at 52-53: “The client application issues a connection request to SQL*Net of the form: <pre>sqlplus scott/tiger@POULTRY</pre>

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.b] wherein the data is associated with an identifier string; <i>continued</i>	where POULTRY is a database service name defined in the Oracle Names Server. SQL*Net determines from the client configuration in the SQLNET.ORA file that the client's default domain is WORLD and the preferred Names Server is the one shown. SQL*Net sends the Names Server a request to resolve the name POULTRY.WORLD." See OracleNamesAdminGuide at 41-42: "All network names include one or more domains. A domain is a group of machines and network services. Domains are usually hierarchically related for organizational and administrative purposes. However, all network names could instead reside in a single domain—the WORLD domain. The top level domain is the predefined root domain. All other domains are hierarchically below the root.... Figure 3 – 6 shows a hierarchical naming model with five domains: the (ROOT) domain, ACME domain, US.ACME, EUROPE.ACME, and ROW.ACME (Rest of World) domains.  Figure 3 – 6 Hierarchical Naming Model Within each domain all names must be unique, but across domains they can be repeated. Notice that both WEATHER and HISTORY are repeated, but the full global names are unique (that is, HISTORY.ROW.ACME vs. HISTORY.EUROPE.ACME). Network domains are similar to file directories used by many operating systems (such as UNIX) in that they are hierarchical; however, unlike file systems, network domains may or may not correspond to any physical arrangement of databases and other objects in a network; they are name spaces." Based on the above teachings, OracleNamesAdminGuide discloses or renders obvious data associated with an identifier string.¹⁷³

¹⁷³ See Greene Decl. at pp.57-59.

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.b] wherein the data is associated with an identifier string; <i>continued</i>	<u>Moreover, OracleSG Renders Obvious Claim Element 18.b.</u> OracleSG explains how OracleNamesAdminGuide's SQL*Net functionalities are used to resolve requests for services to their respective locations. For instance, OracleSG teaches that SQL*Net's Oracle Network Manager allows for associating the data's network address to an "easy name" synonym: "Oracle Network Manager ... allows you to generate the network configuration files for SQL*Net version 2. This version makes it much easier for users to specify a database to connect to; however, the burden of providing the translation between the easy names and the exact server and instance addresses falls on the DBA. I would not recommend configuring SQL*Net version 2 without this utility." <i>See</i> OracleSG at 87. Further, OracleSG teaches that such synonyms were well-known, useful features in Oracle7 systems like OracleNamesAdminGuide's because those synonyms are unique identifiers of data that are associated with the data's address reference string: "A synonym is merely a way of assigning an alias with which users can reference a table or view. ... Synonyms simplify this access because you can create a simple synonym that contains the details of a complex reference string." <i>See</i> OracleSG at 189-90. <i>See also</i> OracleSG at 181: "Data storage objects contain no data of their own. Instead, they reference data that is contained in one or more tables. The following [is a] data storage object[]: Synonyms, which are merely alternative names for tables." <i>See</i> OracleSG at 352: "If you are supporting development, database object creation, ownership, and naming standards become important. When you have controlled applications, the ways users can add data are controlled and they use the computer only to perform necessary business functions.... [Y]ou need to consider object names. This is especially true of permanent table and synonym names that developers may hard-code into their applications and with which you may become stuck. I generally like to establish a table naming hierarchy and have developers create public synonyms for all the data resources that may be moved into production. Try to get all applications coded to look for the public synonym name versus a true schema. table combination. Then there is leeway to reorganize and rename the basic table data without interrupting the applications (just ensure you set your new public synonyms correctly)."

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.b] wherein the data is associated with an identifier string; <i>continued</i>	For the reasons set forth in Section III.E.1 on motivation to combine, a POSITA would have been motivated to have OracleNamesAdminGuide's system associate data with an identifier string that relates to the data's location address, based on OracleSG's teachings. For instance, this would have resulted in the effective use of SQL*Net functionality, and improve query efficiency and optimization of Oracle7 systems to ease users' ability to find their desired data. <i>See</i> OracleSG at 44, 52, 87, 605. ¹⁷⁴
[18.c] a client responsive to a data query to query a data location server for location information associated with the identifier string;	OracleNamesAdminGuide discloses this claim element or renders it obvious alone or in view of OracleSG. OracleNamesAdminGuide describes that an Oracle Names Server (i.e., location server) responds to client queries requesting a data set's connection descriptor (i.e., location information). And OracleNamesAdminGuide discloses that this connection descriptor is associated with the network-service-name-alias (i.e., identifier string) of the requested data set. As explained above in Section III.D.1.c, a POSITA would have understood that OracleNamesAdminGuide's connection descriptor is the claimed "location information" because each connection descriptor is information pertaining to one or more locations of data.¹⁷⁵ <u>Claimed "client"</u>: As discussed above in Section III.C, Patent Owner has proposed "client" means "a network-attached component that initiates update or lookup of identifier/location mappings from a location server with location request messages." See the discussion regarding claim element 18.b showing that OracleNamesAdminGuide discloses the requested data set is associated with a network-service-name-alias (i.e., identifier string). And as the below teachings illustrate, OracleNamesAdminGuide teaches network-attached clients initiating update/location mappings from Names Servers (i.e., location servers) using location requests. <i>See</i> OracleNamesAdminGuide at 51-52: "Role of the Client[:] The client is the software that requests the name to be resolved. The name itself typically comes from the user's application, or the actual user.... When the name is successfully resolved, the client uses it to connect to the intended destination. All interaction with the Names Server and use of the results is transparent to the user. Only an unsuccessful attempt is seen by the client or user."

¹⁷⁴ *See* Greene Decl. at pp.59-60.

¹⁷⁵ *See id.* at p.60.

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.c] a client responsive to a data query to query a data location server for location information associated with the identifier string; <i>continued</i>	See OracleNamesAdminGuide at 24: “Oracle Names has many <i>users</i> and one or more <i>administrators</i>. The true user of Oracle Names is always either <i>a client application in a client-server connection</i>, or an Oracle server in a distributed database. Oracle Names is not visible to the user or the application programmer.” See OracleNamesAdminGuide at 33: Figure 2-2 Oracle Names Server Components Using the Dynamic Discovery Option See OracleNamesAdminGuide at 33: “Architecturally, Oracle Names is a network service that <i>provides name resolution to Oracle clients</i> and servers.” See OracleNamesAdminGuide at 43-44: “<i>A client can request a network service within its default domain using the service’s simple, unqualified name</i>, that is, without specifying a domain name. If a user requests a name without a ‘.’ character in it, the default domain name will be automatically appended to the database service or database link name requested.” See OracleNamesAdminGuide at 91: “The client needs to be able to direct the global name request to a Names Server in the network. Multiple Names Servers should be listed wherever possible for greater reliability when the preferred Names Server is unavailable.” See OracleNamesAdminGuide at 91-92: “Each client or set of similar clients needs to be assigned a client profile. The complete definition of a client profile includes an intersection of the position of the client in the TNS network, and its relation to the Names Servers in its administrative region. The following two sets of information make up the client profile:

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.c] a client responsive to a data query to query a data location server for location information associated with the identifier string; <i>continued</i>	• community—Each client is defined in terms of what community it belongs to, and which Interchange(s) it prefers to try, in order, when connecting to other communities. • Oracle Names—Each client is also defined by the default domain it uses, and the list of Names Servers, in order, that should be contacted to resolve global names.” <u>Claimed “location server”</u>: As explained above in Section III.D.1.a, a POSITA would have understood that OracleNamesAdminGuide’s Names Server is a “location server” because it is a network-attached server that maintains data location mappings to service client requests: “Oracle Names makes network address and database link information available to all nodes throughout the network. <i>Each database server’s network address is stored with a name that is used to identify it. Client applications can then request a database connection with a simple name rather than a lengthy address.</i>” See OracleNamesAdminGuide at 20. See also OracleNamesAdminGuide at 22: “<i>Oracle Names eases the burden of administering the details of the network.</i> If the address of a network object changes, the change is made in one place, and all clients get the benefit of the change immediately. Any number of clients have access to a network object’s address from a single source. <i>The user and application developer are exposed only to the name of a network object. No knowledge of the connection path is required.</i> This is a much more user-friendly way to use networks and network services. As with other user-oriented network software such as electronic mail, people only need to think in terms of names to get their work done. <i>When making a connection, the client specifies the name of a destination and Oracle Names provides the address.</i>” Further, OracleNamesAdminGuide teaches that its Names Server receives a client’s query for location information associated with a network-service-name-alias (i.e., identifier string). As explained above in Section III.D.1.c, a POSITA would have understood that OracleNamesAdminGuide’s connection descriptor is “location information” because each descriptor pertains to a location of data and identifies that location using the host name, port, and database instance.¹⁷⁶

¹⁷⁶ As Mr. Greene explains, a POSITA would have understood that a “host name” may include a direct IP address or a host name that uses DNS to obtain an IP address. Greene Decl. at p.62-63.

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.c] a client responsive to a data query to query a data location server for location information associated with the identifier string; <i>continued</i>	See OracleNamesAdminGuide at 52-53: “The Names Server has a single purpose: to resolve, or assist in resolving, a client-initiated name request. It interprets the request, then looks up the name in its cache, or it calls a Names Server in another region. The response or error conditions are passed back to the client. Figure 3 – 11 shows a client requesting access to a Names Server named POULTRY and a Names Server providing the answer.... The service name is the same as the global object name. For example, a global database name might be POULTRY.WORLD, which is also the service name. <pre> graph TD subgraph Client_Application [Client Application] direction TB C1((1)) C3((3)) C4((4)) end subgraph Oracle_Names_Server [Oracle Names Server] direction TB N2((2)) N[N] end subgraph TNS_Network [TNS Network] direction TB S[Server POULTRY.WORLD] end C1 --> N2 N2 --> C3 C3 --> S S --> C4 </pre> The diagram illustrates the client-server connection process. It features four main components: a Client Application (represented by a computer icon), an Oracle Names Server (represented by a computer icon and a box labeled 'N'), a TNS Network (represented by a large oval), and a Server POULTRY.WORLD (represented by a server rack icon). The sequence of events is numbered 1 through 4: 1. The Client Application sends a request to the Oracle Names Server. 2. The Oracle Names Server responds to the Client Application. 3. The Client Application sends a request to the Server POULTRY.WORLD. 4. The Server POULTRY.WORLD responds to the Client Application. The TNS Network is shown as the central communication hub connecting the Client Application and the Server POULTRY.WORLD. Figure 3 – 11 Client-Server Connection The sequence of events is as follows: 1. The client application issues a connection request to SQL*Net of the form: <code>sqlplus scott/tiger@POULTRY</code> where POULTRY is a database service name defined in the Oracle Names Server. - SQL*Net determines from the client configuration in the SQLNET.ORA file that the client's default domain is WORLD and the preferred Names Server is the one shown. SQL*Net sends the Names Server a request to resolve the name POULTRY.WORLD. 2. The Names Server receives the request, looks it up in the cache, then sends the response back to the client. 3. The client receives the answer and substitutes the address in place of the initial name POULTRY. 4. The client contacts POULTRY and establishes a standard SQL*Net client-server connection.”

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.c] a client responsive to a data query to query a data location server for location information associated with the identifier string; <i>continued</i>	See OracleNamesAdminGuide at 23-24: “<i>The Names Server stores the following types of objects</i>, which are described below: • database service names—<i>The service name is the global database name that is mapped to the SQL*Net connect descriptor.</i> The combination of a listener address and connect data make up a connect descriptor that locates a database service. Gateways to non-Oracle databases and Oracle RDB databases are also stored. • database links—Global database links are automatically created to each defined database from all database servers in the network. These links are made available to all users. • aliases—<i>You can assign service name aliases as alternate names for any defined database service or database link.</i> Service name aliases are resolved as if they were the original object. • V1 connect strings—<i>A V1 connect string can be mapped to a service name in the Oracle Names database.</i> The service name you enter acts as an alias for the SQL*Net V1 connect string, making it easy to access a server in a network that still uses SQL*Net V1.... After services have been defined in Oracle Names, <i>users only need to refer to them by name to connect to a database or network service.</i> The necessary exchange of details occurs between the requester and the Oracle Names Server.” See OracleNamesAdminGuide at 119: “Summary of NAMESCTL Commands ... QUERY [will q]uery for the existence or contents of a network object name.” See OracleNamesAdminGuide at 128-129: “QUERY Purpose: Test or retrieve the contents of a network object stored in the Names Server.... Syntax: From the operating system: NAMESCTL QUERY object_name [object_type] [modifiers] From NAMESCTL program: QUERY name [type] [modifiers] Valid types: A.SMD Network addresses, as with database service definitions. CNAME.SMD Alias name (sometimes referred to as ‘canonical name’). DL.RDBMS.OMD Database link. NS.SMD Names Server addresses. System data used to communicate between Names Servers.

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.c] a client responsive to a data query to query a data location server for location information associated with the identifier string; <i>continued</i>	VIADD.NPO.OMD SQL*Net Version 1 connect string.... Arguments: Mandatory network object name and network object type. Usage: QUERY can be used to test that a defined piece of data can be found, and that the contents are correct.... Notes: If the QUERY command is used with just a name as a parameter, the Names Server responds with the number of pieces of data with that name, and the time required to complete the operation. If the QUERY command is used with the name and type supplied as arguments; the specific name is looked up and returned to the user.... Example: NAMESCTL> QUERY BONES.DEM.MEDICINE A.SMD Total response time: 0.04 seconds Response status: normal, successful completion Authoritative answer: yes Number of answers: 1 Canonical name: bones.dem.medicine TTL: 1 day Alias translations: from: bones.dem.medicine to: bones.dem.medicine Answers: data type is 'a.smd' <i>Syntax is ADDR:... (DESCRIPTION=(ADDRESS=(COMMUNITY=tcp)(PROTOCOL=TCP)(Host=cowboy) (Port=1522))(CONNECT_DATA=(SID=rodeo)))].f'</i> See OracleNamesAdminGuide at 109: "When a server initiates the use of a database link, the Names Server is called, and the request is executed, as with locally defined database links (links defined in the local database data dictionary). Consider, for example, the SQL statement shown: <pre>SELECT A.ITEM, B.COST FROM PARTS A, PRICE@FINANCE.US.ACME B WHERE A.PARTNO = B.PARTNO;</pre> FINANCE.US.ACME is looked up in the Names Server. Then the link is established between the current database and the FINANCE database.

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.c] a client responsive to a data query to query a data location server for location information associated with the identifier string; <i>continued</i>	(FINANCE is the name of the database service, as well as the name of the global database link.)[.]” See OracleNamesAdminGuide at 110: “When a database link is used to select information from a remote database table (for example, SELECT * FROM EMP@HR.US.ORACLE.COM), Oracle searches for a connect descriptor in the following locations in the following order: 1. the user’s private database links 2. the database public database links 3. the Oracle Names Server global database links Oracle ends the search when a USING clause is found in either a private or public database link. If a USING clause is not found in a private or public link, then the global database link is used. If a connect descriptor is not found, the following message is returned: ORA-02019: “connection description for remote database not found” If a CONNECT TO clause (username/password) is found in any of the searched database links, the first one found is used. If none are found, then the current username and password will be used.” See OracleNamesAdminGuide at 93: “When th[e names.default_domain] parameter is set to the default domain name (for example, US.ACME), the domain name will be automatically appended to the service name. For example, a client with a default_domain of US.ACME can request a database service named FINANCE.US.ACME as: sqlplus scott/tiger@FINANCE All names outside the default domain must be fully specified as global names; for example, FINANCE.US.ACME.... The Network Manager automatically adds the NAME.DEFAULT_ZONE parameter in addition to the NAMES.DEFAULT_DOMAIN for backward compatibility.” OracleNamesAdminGuide at 108: “The client requires a SQLNET.ORA file with one or more Names Server addresses defined in the NAMES.PREFERRED_SERVERS parameter. These Names Servers are contacted in order until a response is returned. Additionally, each client default domain is set with the NAMES.DEFAULT_DOMAIN parameter to simplify how domains are specified during connection requests and when using NAMESCTL.... To request access to a network object, the client uses the familiar client/server connection syntax: sqlplus scott/tiger@FINANCE[.]”

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.c] a client responsive to a data query to query a data location server for location information associated with the identifier string; <i>continued</i>	<i>See OracleNamesAdminGuide at 44: “Every region must have a unique name. The region, although it contains domains, is defined to ”reside” in one of the domains that it contains. The region name is formed by appending one of the domain names onto the end of the region name. For example, a region containing the US.ACME and UK.ACME domains could be called either region_name.US.ACME or region_name.UK.ACME.”</i> <i>See OracleNamesAdminGuide at 56-57: “Many companies already have global naming standards for their PC LANs, large hosts, or mail systems. If such policies exist, you can use them as the basis for the Oracle Names naming model. Using the same structure leverages the investment in the education users already have with global names.</i> <i>For example, many companies with TCP/IP networks use the Domain Name Service (DNS) model for communication on the worldwide Internet. In this model, all naming models are hierarchical from a set of base domains such as:</i> <i>• COM - commercial organizations</i> <i>• EDU - educational institutions</i> <i>• MIL - military</i> <i>• ORG – organizations</i> <i>Individual companies are then assigned domains within which to build their naming model (for example, ACME.COM)....</i> <i>For organizations which are on the Internet, and therefore already have one or more DNS domains, the sensible choice is to align their Oracle naming domains with their DNS domains.”</i> <i>See also OracleNamesAdminGuide at 29 (explaining Names Servers’ default TCP/IP port is 1575), 29 (describing a new service’s registration in system results in automatic replication of a service’s location in Names Servers), 30 (“the Names Server remembers the addresses of all databases”), 31 (teaching that Names Server stores network-service-name-aliases as they are registered), 33 (explaining Names Server’s cache “is an in-memory database of all data required to resolve names”), 83 (describing each Names Server has a unique service name), 99 (describing Names Servers loads network definitions of global database links), 107 (describing example query using network-service-name-alias and type to return information about the network object).</i>

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.c] a client responsive to a data query to query a data location server for location information associated with the identifier string; <i>continued</i>	Based on the above teachings, OracleNamesAdminGuide discloses or renders obvious a client responsive to a data query to query a data location server for location information associated with the identifier string.¹⁷⁷ <u>Moreover, OracleSG Renders Obvious Claim Element 18.c.</u> OracleSG explains how OracleNamesAdminGuide's SQL*Net functionalities are used to query a location server for location information associated with a network-service-name-alias (i.e., identifier string). As a foundation, OracleSG explains that it was understood Oracle systems provide "relational database management system (RDBMS) ... designed to provide the facilities to store and retrieve information in a variety of formats." <i>See</i> OracleSG at 114. Further, OracleSG teaches SQL*Net ensures translation of entity addresses to easy-to-remember identifiers: "Oracle Network Manager ... allows you to generate the network configuration files for SQL*Net version 2. This version makes it much easier for users to specify a database to connect to; however, the burden of providing the translation between the easy names and the exact server and instance addresses falls on the DBA. I would not recommend configuring SQL*Net version 2 without this utility." <i>See</i> OracleSG at 87. <i>See also</i> OracleSG at 189: "When you reference tables located in a different database using SQL *Net, the format can be rather complicated. Synonyms simplify this access because you can create a simple synonym that contains the details of a complex reference string." <i>See also</i> OracleSG at 152: "The following is the SQL query that enables you to determine the location of your data files:

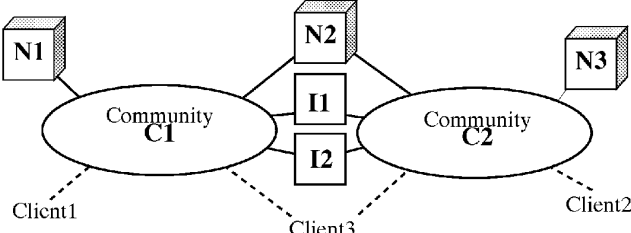
¹⁷⁷ *See* Greene Decl. at pp.60-68.

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.c] a client responsive to a data query to query a data location server for location information associated with the identifier string; <i>continued</i>	SQL> select * from sys.dba_data_files; FILE_NAME FILE_ID TABLESPACE_NAME BYTES BLOCKS STATUS C:\ORAWIN\OR8S1\edbs.ora 3 ROLLBACK_DATA 3145728 1536 AVAILABLE C:\ORAWIN\OR8S1\edbttemp.ora 4 TEMPORARY_DATA 2897152 1024 AVAILABLE C:\ORAWIN\OR8S1\edbuser.ora 2 USER_DATA 3145728 1536 AVAILABLE C:\ORAWIN\OR8S1\edbsys.ora 1 SYSTEM 13425768 5120 AVAILABLE SQL> [.]” For the reasons set forth in Section III.E.1 on motivation to combine, a POSITA would have been motivated to have OracleNamesAdminGuide’s system query a Names Server for location information associated with an identifier string, based on OracleSG’s teachings. For instance, this would have resulted in the effective use of SQL*Net functionality, and improve query efficiency and optimization of Oracle7 systems to ease users’ ability to find their desired data. <i>See</i> OracleSG at 44, 52, 87, 605.¹⁷⁸
[18.d] a data location server network comprising a plurality of data location servers,	OracleNamesAdminGuide discloses this claim element. As shown above regarding claim element 18.c, OracleNamesAdminGuide describes a system having a Names Server (i.e., location server). Those teachings are incorporated herein. Further, as shown below in Figures 2-1 and 3-1, OracleNamesAdminGuide teaches that its system includes a network of multiple Names Servers. <i>See</i> OracleNamesAdminGuide at 32 (“We recommend that you set up at least two Names Servers for fault tolerance.”):

¹⁷⁸ *See* Greene Decl. at pp.68-69.

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.d] a data location server network comprising a plurality of data location servers, <i>continued</i>	Figure 2 – 1 Components of a Network Using Oracle Names with the Dynamic Discovery Option OracleNamesAdminGuide at 33: Figure 3 – 1 Components of a Network Using Oracle Names Without the Dynamic Discovery Option <i>See also</i> OracleNamesAdminGuide at 60: “Oracle recommends that each administrative region should have at least two Names Servers. If one Names Server is unavailable, clients will automatically forward requests to the other. The degree of availability this allows depends on the regularity of service each node provides. For example, imagine that the amount of time a Name Servers node is down for unscheduled reasons is 1%. (It is usually much less.) The likelihood that both would be down at the same time is then roughly .01%, or 1% of 1%. There is also an issue of network loading, or load balancing. Although any given Names Server is only engaged for a short period of time when a client initiates a connection, the goal is to keep name resolutions lightning fast. With a very large volume of clients, or applications requiring

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.d] a data location server network comprising a plurality of data location servers, <i>continued</i>	many connections, load balancing across multiple Names Servers can increase system performance.” See OracleNamesAdminGuide at 67: “The Names Servers on the network using DDO must know about each other.” See OracleNamesAdminGuide at 29: “Since well-known Names Servers continually reconcile the contents of their cache with those of other active well-known Names Servers, registration with a single well-known Names Server is sufficient for an object to be known by all the well-known Names Servers on the network. This concept is known as <i>service replication</i>.” See OracleNamesAdminGuide at 44: “Each administrative region contains one or more Oracle Names Servers. Names Servers are used to translate global service names into their network addresses. Although having just one Names Server per administrative region is technically allowed, at least two Names Servers per region are recommended to provide highly reliable access to the details of the network.” OracleNamesAdminGuide at 51: “You use the Network Manager to configure: • the network data - The definition of the communities and MultiProtocol Interchanges in the network, and all Names Servers in the root administrative region. (This is sometimes referred to as “backbone data.”) • the administrative regions and their domains - The division of configuration into one Network Manager installation per region. The simplest and most common case is one administrative region with one domain (a flat naming model), although one administrative region with many domains, or multiple administrative regions, is also possible. • the Oracle Names Servers - <i>Within each region, the specifics of each of the Oracle Names Servers are also defined. Names Server specifics include the network node and addresses it runs on,</i> and various startup parameters affecting how it operates. • network objects in the local administrative region - The definition of database services, database links, and any aliases is also done through the Network Manager.” OracleNamesAdminGuide at 179-180:

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.d] a data location server network comprising a plurality of data location servers, <i>continued</i>	TNS Network  Figure C – 1 Sample Client Profiles “Names Servers [are] N1,N2, N3[.]” Based on the above teachings, OracleNamesAdminGuide discloses a data location server network comprising a plurality of data location servers.¹⁷⁹
[18.e] at least one of the plurality of data location servers containing location information associated with the identifier string,	OracleNamesAdminGuide discloses this claim element or renders it obvious alone or in view of OracleSG. As shown above regarding claim element 18.c, OracleNamesAdminGuide teaches its Names Server (i.e., location server) acts upon a connection descriptor (i.e., location information) associated with a network-service-name-alias (i.e., identifier string). Those teachings are incorporated herein. Further, OracleNamesAdminGuide explains that its Names Server stores connection descriptor associated with an entity (e.g., a database) and that the connection descriptor is associated with a network-service-name-alias: “An Oracle Names Server stores names and addresses for network services such as databases or MultiProtocol Interchanges, database link definitions, and object aliases. The Names Server stores the following types of objects, which are described below: • database service names—<i>The service name is the global database name that is mapped to the SQL *Net connect descriptor. The combination of a listener address and connect data make up a connect descriptor that locates a database service.</i> Gateways to non-Oracle databases and Oracle RDB databases are also stored. • database links—Global database links are automatically created to each defined database from all database servers in the network. These links are made available to all users. • <i>aliases</i>—You can assign service name aliases as alternate names for any defined database service or database link. <i>Service name aliases are resolved as if they were the original object.</i>

¹⁷⁹ See Greene Decl. at pp.69-71.

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.e] at least one of the plurality of data location servers containing location information associated with the identifier string, <i>continued</i>	• V1 connect strings—A V1 <i>connect string can be mapped to a service name in the Oracle Names database</i>. The service name you enter acts as an alias for the SQL*Net V1 connect string, making it easy to access a server in a network that still uses SQL*Net V1.... After services have been defined in Oracle Names, <i>users only need to refer to them by name to connect to a database or network service</i>. The necessary exchange of details occurs between the requester and the Oracle Names Server.” See OracleNamesAdminGuide at 23. Further, as the below example shows, OracleNamesAdminGuide teaches that its Names Server uses the POULTRY.WORLD alias to identify the location information associated with the alias and sends this location information to the client: “The Names Server has a single purpose: to resolve, or assist in resolving, a client-initiated name request. It interprets the request, then looks up the name in its cache, or it calls a Names Server in another region. The response or error conditions are passed back to the client. Figure 3 – 11 shows a client requesting access to a Names Server named POULTRY and a Names Server providing the answer.... The service name is the same as the global object name. For example, a global database name might be POULTRY.WORLD, which is also the service name. <pre> sequenceDiagram participant Client as Client Application participant NS as Oracle Names Server participant Server as Server POULTRY.WORLD participant TNS as TNS Network Client->>NS: 1 NS->>Client: 2 Client->>TNS: 3 TNS->>Server: 4 </pre> The diagram illustrates the client-server connection process. It features three main components: a 'Client Application' (represented by a computer icon), an 'Oracle Names Server' (represented by a server rack icon), and a 'Server POULTRY.WORLD' (represented by a server rack and a database cylinder icon). These components are interconnected via a central 'TNS Network' (represented by a large oval). The sequence of events is numbered 1 through 4: 1. An arrow points from the Client Application to the Oracle Names Server. 2. An arrow points from the Oracle Names Server back to the Client Application. 3. An arrow points from the Client Application to the TNS Network. 4. An arrow points from the TNS Network to the Server POULTRY.WORLD. Figure 3 – 11 Client-Server Connection The sequence of events is as follows: 1. The client application issues a connection request to SQL*Net of the form: <code>sqlplus scott/tiger@POULTRY</code> where POULTRY is a database service name defined in the Oracle Names Server.

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.e] at least one of the plurality of data location servers containing location information associated with the identifier string, <i>continued</i>	SQL*Net determines from the client configuration in the SQLNET.ORA file that the client's default domain is WORLD and the preferred Names Server is the one shown. SQL*Net sends the Names Server a request to resolve the name POULTRY.WORLD. 2. The Names Server receives the request, looks it up in the cache, then sends the response back to the client. 3. The client receives the answer and substitutes the address in place of the initial name POULTRY. 4. The client contacts POULTRY and establishes a standard SQL*Net client-server connection." See OracleNamesAdminGuide at 52-53. See OracleNamesAdminGuide at 128-129: "QUERY Purpose: Test or retrieve the contents of a network object stored in the Names Server.... Syntax: From the operating system: NAMECTL QUERY object_name [object_type] [modifiers] From NAMECTL program: QUERY name [type] [modifiers] Valid types: A.SMD Network addresses, as with database service definitions. CNAME.SMD Alias name (sometimes referred to as "canonical name"). DL.RDBMS.OMD Database link. NS.SMD Names Server addresses. System data used to communicate between Names Servers. VIADD.NPO.OMD SQL*Net Version 1 connect string.... Arguments: Mandatory network object name and network object type. Usage QUERY can be used to test that a defined piece of data can be found, and that the contents are correct.... Notes: If the QUERY command is used with just a name as a parameter, the Names Server responds with the number of pieces of data with that name, and the time required to complete the operation. If the QUERY command is used with the name and type supplied as arguments; the specific name is looked up and returned to the user....

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.e] at least one of the plurality of data location servers containing location information associated with the identifier string, <i>continued</i>	Example: NAMESCTL> QUERY BONES.DEM.MEDICINE A.SMD Total response time: 0.04 seconds Response status: normal, successful completion Authoritative answer: yes Number of answers: 1 Canonical name: bones.dem.medicine TTL: 1 day Alias translations: from: bones.dem.medicine to: bones.dem.medicine Answers: data type is 'a.smd' Syntax is ADDR:...(DESCRIPTION=(ADDRESS= (COMMUNITY=tcp)(PROTOCOL=TCP)(Host=cowboy) (Port=1522))(CONNECT_DATA=(SID=rodeo)))[.] See OracleNamesAdminGuide at 109: “When a server initiates the use of a database link, the Names Server is called, and the request is executed, as with locally defined database links (links defined in the local database data dictionary). Consider, for example, the SQL statement shown: <pre>SELECT A.ITEM, B.COST FROM PARTS A, PRICE@FINANCE.US.ACME B WHERE A.PARTNO = B.PARTNO;</pre> FINANCE.US.ACME is looked up in the Names Server. Then the link is established between the current database and the FINANCE database. (FINANCE is the name of the database service, as well as the name of the global database link.)[.]” See OracleNamesAdminGuide at 110: “When a database link is used to select information from a remote database table (for example, SELECT * FROM EMP@HR.US.ORACLE.COM), Oracle searches for a connect descriptor in the following locations in the following order: 1. the user’s private database links 2. the database public database links 3. the Oracle Names Server global database links Oracle ends the search when a USING clause is found in either a private or public database link. If a USING clause is not found in a private or

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.e] at least one of the plurality of data location servers containing location information associated with the identifier string, <i>continued</i>	public link, then the global database link is used. If a connect descriptor is not found, the following message is returned: ORA-02019: "connection description for remote database not found" If a CONNECT TO clause (username/password) is found in any of the searched database links, the first one found is used. If none are found, then the current username and password will be used." <i>See also</i> OracleNamesAdminGuide at 93 (describing associating location information with FINANCE.US.ACME alias), 108 (describing associating location information with FINANCE alias), 44 (describing unique network-service-name-alias for each domain), 56-57 (using DNS model to create unique network-service-name-alias associated with location information). Based on the above teachings, OracleNamesAdminGuide discloses or renders obvious one Names Server contains location information, including host name, port, and database instance, associated with the identifier string (i.e., network-service-name-alias).¹⁸⁰ <u>Moreover, OracleSG Renders Obvious Claim Element 18.e.</u> OracleSG explains how OracleNamesAdminGuide's SQL*Net configuration allows for associating entity addresses to easy-to-remember identifiers: "Oracle Network Manager ... allows you to generate the network configuration files for SQL*Net version 2. This version makes it much easier for users to specify a database to connect to; however, the burden of providing the translation between the easy names and the exact server and instance addresses falls on the DBA. I would not recommend configuring SQL*Net version 2 without this utility." <i>See</i> OracleSG at 86-87. <i>See also</i> OracleSG at 189: "When you reference tables located in a different database using SQL *Net, the format can be rather complicated. Synonyms simplify this access because you can create a simple synonym that contains the details of a complex reference string." <i>See also</i> OracleSG at 152: "The following is the SQL query that enables you to determine the location of your data files:

¹⁸⁰ *See* Greene Decl. at pp.71-75.

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey																														
[18.e] at least one of the plurality of data location servers containing location information associated with the identifier string, <i>continued</i>	<pre>SQL> select * from sys.dba_data_files;</pre> <table><thead><tr><th>FILE_NAME</th><th>FILE_ID</th><th>TABLESPACE_NAME</th><th>BYTES</th><th>BLOCKS</th><th>STATUS</th></tr></thead><tbody><tr><td>C:\ORAWIN\ORSL\ddrns.ora</td><td>3</td><td>ROLLBACK_DATA</td><td>3145728</td><td>1536</td><td>AVAILABLE</td></tr><tr><td>C:\ORAWIN\ORSL\ddtemp.ora</td><td>4</td><td>TEMPORARY_DATA</td><td>2807152</td><td>1024</td><td>AVAILABLE</td></tr><tr><td>C:\ORAWIN\ORSL\dduser.ora</td><td>2</td><td>USER_DATA</td><td>3145728</td><td>1536</td><td>AVAILABLE</td></tr><tr><td>C:\ORAWIN\ORSL\ddsys.ora</td><td>1</td><td>SYSTEM</td><td>10485760</td><td>8128</td><td>AVAILABLE</td></tr></tbody></table> <pre>SQL></pre> [.]” For the reasons set forth in Section III.E.1 on motivation to combine, a POSITA would have been motivated to have OracleNamesAdminGuide’s Names Server contain location information associated with the identifier string, based on OracleSG’s teachings. For instance, this would have resulted in the effective use of SQL*Net functionality, and improve query efficiency and optimization of Oracle7 systems to ease users’ ability to find their desired data. <i>See</i> OracleSG at 44, 52, 87, 605.¹⁸¹	FILE_NAME	FILE_ID	TABLESPACE_NAME	BYTES	BLOCKS	STATUS	C:\ORAWIN\ORSL\ddrns.ora	3	ROLLBACK_DATA	3145728	1536	AVAILABLE	C:\ORAWIN\ORSL\ddtemp.ora	4	TEMPORARY_DATA	2807152	1024	AVAILABLE	C:\ORAWIN\ORSL\dduser.ora	2	USER_DATA	3145728	1536	AVAILABLE	C:\ORAWIN\ORSL\ddsys.ora	1	SYSTEM	10485760	8128	AVAILABLE
FILE_NAME	FILE_ID	TABLESPACE_NAME	BYTES	BLOCKS	STATUS																										
C:\ORAWIN\ORSL\ddrns.ora	3	ROLLBACK_DATA	3145728	1536	AVAILABLE																										
C:\ORAWIN\ORSL\ddtemp.ora	4	TEMPORARY_DATA	2807152	1024	AVAILABLE																										
C:\ORAWIN\ORSL\dduser.ora	2	USER_DATA	3145728	1536	AVAILABLE																										
C:\ORAWIN\ORSL\ddsys.ora	1	SYSTEM	10485760	8128	AVAILABLE																										
[18.f] wherein each of the plurality of data location servers comprises computer executable code configured to execute the following steps in response to receiving a data location request from the client:	OracleNamesAdminGuide discloses this claim element or renders it obvious alone or in view of OracleSG. As shown below in Figure 2-2, OracleNamesAdminGuide explains that its Names Server includes a processor and memory. As Mr. Greene explains, a POSITA would have understood that, in order for this processor to function, the memory contains computer executable code to execute steps.¹⁸² Further, OracleNamesAdminGuide teaches that its Names Server (i.e., location server) loads and executes software to preform steps in response to a data location request: “At startup, the Names Server reads its configuration files to set up its operating parameters, then loads all data for the administrative region.” <i>See</i> OracleNamesAdminGuide at 160. OracleNamesAdminGuide at 33 (explaining processor “receives incoming address requests and <i>looks through the cache for the answer</i>”):																														

¹⁸¹ *See* Greene Decl. at p.76.

¹⁸² *See id.* at pp.76-78.

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.f] wherein each of the plurality of data location servers comprises computer executable code configured to execute the following steps in response to receiving a data location request from the client: <i>continued</i>	<pre> graph LR Client[Client] -- Incoming Name Requests --> RP((Request Processor)) RP <--> Cache[Cache] RP --> F[Forwarder] F -- Service Registration --> NL1((Network listener)) F -- Service Registration --> NL2((Network listener)) NL1 --- SX[Server X] NL2 --- SY[Server Y] NL1 -- Answered Name Requests --> Client NL2 -- Answered Name Requests --> Client </pre> Figure 2 – 2 Oracle Names Server Components Using the Dynamic Discovery Option See also OracleNamesAdminGuide at 62-63: “An Oracle Names Server is portable software that runs on a variety of computer platforms.... In general, Oracle Names runs on: • A variety of UNIX and VAX/VMS minicomputers and workstations. • Intel-based computers (PCs) running operating systems such as OS/2, NetWare, Windows NT, and UNIX. The computer does not have to be dedicated as an Oracle Names Server. In fact, Oracle Names typically requires relatively little CPU and memory to function. Your choice of computer to use as a Names Server should be based on: • Software availability—While Oracle Names runs on many operating systems, it does not run on all. As a shared resource, Oracle Names only needs to be on one type of operating system to serve all clients and servers. It does not have to be available on all of the operating systems you use....” See OracleNamesAdminGuide at 104: “The STARTUP command loads the Names Server into memory and tells it to begin executing. At startup, the Names Server loads its configuration, loads its data, then becomes available to answer requests.... The SHUTDOWN command stops the execution of a Names Server. The program stops executing, and releases any machine resources being used.” See OracleNamesAdminGuide at 116: “You can run NAMESCTL in three modes: • Interpreter mode - NAMESCTL is loaded and all operations are executed within it. When loaded, the program displays the following prompt: NAMESCTL>

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.f] wherein each of the plurality of data location servers comprises computer executable code configured to execute the following steps in response to receiving a data location request from the client: <i>continued</i>	• Command line mode - You can also execute most commands from the operating system command line by running the NAMESCTL program with a complete command as a parameter to the program. In this case, NAMESCTL will load and execute the command, then return the operating system prompt. Sample commands are: NAMESCTL START NAMESCTL STATUS CHEDDAR.ACME[.] Based on the above teachings, OracleNamesAdminGuide discloses or renders obvious each Names Server comprises computer executable code configured to execute the steps in response to receiving a data location request from the client.¹⁸³ <u>Moreover, OracleSG Renders Obvious Claim Element 18.f</u> OracleSG explains how OracleNamesAdminGuide's SQL*Net configuration executes code on clients and servers' memory and that the servers execute steps in response to data location requests. <i>See OracleSG at 313: "SQL*Net listeners. If you are planning a client-server installation, you need to start up SQL*Net listeners for each of the versions and protocols of SQL*Net that you are using."</i> <i>See OracleSG at 168: "Another advantage of stored procedures is that Oracle stores both the source code and a parsed version of the software. This saves Oracle the time of having to parse the statements each time as it would for SQL queries received from external software applications."</i> <i>See OracleSG at 125: "The first memory area of interest is the one that stores software code. All that functionality from the Oracle database software needs a home. This is the location for it. On some operating systems, this software area can be shared between instances. Note that this is the software that runs the database itself, not the applications that you have written and are running."</i> <i>See OracleSG at 133: "Computers execute instructions stored in memory. If you want your general ledger application to work, you need to have memory space available for it. This same concept applies to storing code for the various Oracle background processes."</i>

¹⁸³ See Greene Decl. at pp.76-78.

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.f] wherein each of the plurality of data location servers comprises computer executable code configured to execute the following steps in response to receiving a data location request from the client: <i>continued</i>	<i>See OracleSG at 142: “The memory space used to store the Oracle application software code is owned by the Oracle user and should be protected by the operating system from modification by others.”</i> <i>See OracleSG at 682: “... Oracle stores the procedures (not triggers) in both source code and parsed (ready-to-execute) forms, you do not have to spend time parsing queries and transactions that are stored in the database already.”</i> <i>See OracleSG at 87: “Oracle Network Manager ... allows you to generate the network configuration files for SQL*Net version 2. This version makes it much easier for users to specify a database to connect to; however, the burden of providing the translation between the easy names and the exact server and instance addresses falls on the DBA. I would not recommend configuring SQL*Net version 2 without this utility.”</i> <i>See OracleSG at 152: “The following is the SQL query that enables you to determine the location of your data files:</i> <pre> SQL> select * from sys.dba_data_files; FILE_NAME ----- FILE_ID TABLESPACE_NAME BYTES BLOCKS STATUS ----- C:\ORAWIN\ORSG\wdbase.ora 3 ROLLBACK_DATA 3145728 1536 AVAILABLE C:\ORAWIN\ORSG\wdbtemp.ora 4 TEMPORARY_DATA 2687152 1324 AVAILABLE C:\ORAWIN\ORSG\wdbuser.ora 2 USER_DATA 3145728 1536 AVAILABLE C:\ORAWIN\ORSG\wdbsys.ora 1 SYSTEM 10485760 5120 AVAILABLE SQL> </pre> [.]” For the reasons set forth in Section III.E.1 on motivation to combine, a POSITA would have been motivated to have OracleNamesAdminGuide’s Names Server include computer executable code configured to execute steps in response to receiving a data location request from the client, based on OracleSG’s teachings. For instance, this would have resulted in the effective use of SQL*Net functionality, and improve query

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.g] if the data location server contains the location string¹⁸⁵ associated with the identification string¹⁸⁶ provided in the data location request, the data location server transmits location information for use by the client to calculate a location of the data associated with the identification string;	efficiency and optimization of Oracle7 systems to ease users' ability to find their desired data. <i>See</i> OracleSG at 44, 52, 87, 605.¹⁸⁴ OracleNamesAdminGuide discloses this claim element or renders it obvious alone or in view of OracleSG. As shown regarding claim element 18.c, OracleNamesAdminGuide teaches its Names Server (i.e., location server) receives a query for a connection descriptor (i.e., location information) associated with a network-service-name-alias (i.e., identifier string). As shown below in the example for Figure 3-1 and the QUERY command, OracleNamesAdminGuide explains that, in response to the POULTRY query, the Names Server transmits location information, including host name, port, and database instance, to the client if the server contains location information associated with the POULTRY.WORLD alias. As shown below in Figure 3-1, as step 4, the client uses that information to calculate a location of the data associated with the POULTRY.WORLD alias. OracleNamesAdminGuide at 52-53: "Figure 3 – 11 shows a client requesting access to a Names Server named POULTRY and a Names Server providing the answer.... The service name is the same as the global object name. For example, a global database name might be POULTRY.WORLD, which is also the service name. <pre> sequenceDiagram participant Client as Client Application participant NS as Oracle Names Server participant Server as Server POULTRY.WORLD participant TNS as TNS Network Client->>NS: 1 NS->>Client: 2 Client->>TNS: 3 TNS->>Server: 4 </pre> The diagram illustrates a client-server connection sequence. A Client Application (represented by a computer icon) sends a request (1) to the Oracle Names Server (represented by a server rack icon). The Oracle Names Server responds (2) to the Client Application. The Client Application then sends a request (3) to the TNS Network (represented by a large oval). The TNS Network then sends a request (4) to the Server POULTRY.WORLD (represented by a server rack icon). The Server POULTRY.WORLD is also connected to a database cylinder icon. Figure 3 – 11 Client-Server Connection The sequence of events is as follows:

¹⁸⁴ *See* Greene Decl. at pp.78-80.

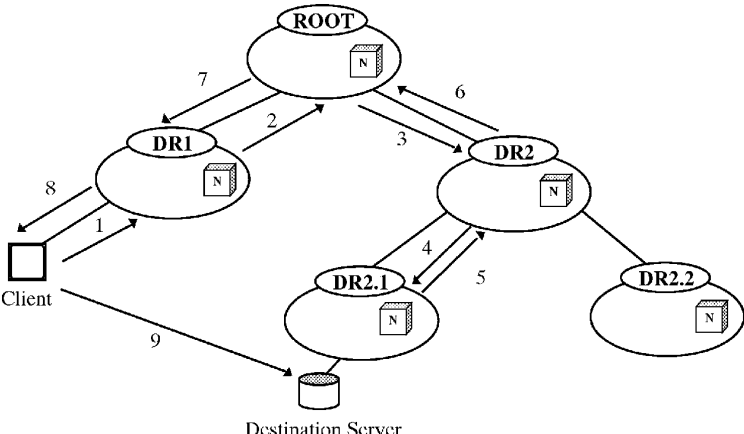
¹⁸⁵ While unclear, it appears claim element 18.g's "location string" refers to the "location information" recited in claim element 18.e.

¹⁸⁶ While unclear, it appears claim element 18.g's "identification string" refers to the "identifier string" recited in claim elements 18.b, 18.c, and 18.e.

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.g] if the data location server contains the location string associated with the identification string provided in the data location request, the data location server transmits location information for use by the client to calculate a location of the data associated with the identification string; <i>continued</i>	1. The client application issues a connection request to SQL*Net of the form: <code>sqlplus scott/tiger@POULTRY</code> where POULTRY is a database service name defined in the Oracle Names Server. SQL*Net determines from the client configuration in the SQLNET.ORA file that the client's default domain is WORLD and the preferred Names Server is the one shown. SQL*Net sends the Names Server a request to resolve the name POULTRY.WORLD. 2. The Names Server receives the request, looks it up in the cache, then sends the response back to the client. 3. The client receives the answer and substitutes the address in place of the initial name POULTRY. 4. The client contacts POULTRY and establishes a standard SQL*Net client-server connection." See OracleNamesAdminGuide at 128-129: "QUERY Purpose: Test or retrieve the contents of a network object stored in the Names Server.... Usage QUERY can be used to test that a defined piece of data Notes: can be found, and that the contents are correct.... If the QUERY command is used with just a name as a parameter, the Names Server responds with the number of pieces of data with that name, and the time required to complete the operation. <i>If the QUERY command is used with the name and type supplied as arguments; the specific name is looked up and returned to the user....</i> Example: NAMESCTL> QUERY BONES.DEM.MEDICINE A.SMD Total response time: 0.04 seconds Response status: normal, successful completion Authoritative answer: yes Number of answers: 1 Canonical name: bones.dem.medicine TTL: 1 day Alias translations: from: bones.dem.medicine to: bones.dem.medicine Answers:

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.g] if the data location server contains the location string associated with the identification string provided in the data location request, the data location server transmits location information for use by the client to calculate a location of the data associated with the identification string; <i>continued</i>	data type is 'a.smd' Syntax is ADDR:... (DESCRIPTION=(ADDRESS=(COMMUNITY=tcp)(PROTOCOL=TCP)(Host=cowboy) (Port=1522))(CONNECT_DATA=(SID=rodeo)))[.] Similar to the Figure 3-11 example, OracleNamesAdminGuide teaches a Figure 3-12 example in which the client sends a request to the Names Servers. In response to the request, the Names Server transmits location information to the client if the server contains location information associated with the JERRY.WORLD alias, and the client uses that information to calculate a location of the data associated with the serve name at step 4: "Name resolution in a distributed database in the single administrative region configuration is similar to names resolution in a decentralized administrative region. The global database link name must be the same as the global database name...." The diagram illustrates a distributed database connection. At the top, a box contains the SQL statement: <code>SELECT OBJECT FROM RUN@JERRY WHERE OBJECT LIKE '%STICK%'</code>. Below this, a client (represented by a computer icon) sends a request (labeled 1) to an Oracle Names Server (represented by a computer icon). The Oracle Names Server is connected to a TNS Network (represented by a large oval). The TNS Network is connected to a database server (represented by a computer icon) labeled 'Server JERRY.WORLD'. The database server is also connected to a 'DB Link JERRY.WORLD' (represented by a computer icon). The sequence of events is numbered 1 through 4: 1. Client sends request to Names Server; 2. Names Server responds to client; 3. Client sends request to database server; 4. Database server responds to client. Figure 3 – 12 Distributed Database Connection The sequence of events is as follows: 1. Server TOM receives the SQL statement shown and checks its data dictionary for a private or public database link called JERRY. When it does not find one, SQL*Net determines from the client configuration in the SQLNET.ORA file that the client's default domain is WORLD and the preferred Names Server is the one shown. SQL*Net then sends the Names Server a request to resolve the database link name JERRY.WORLD. 2. The Names Server receives the request, looks it up in the cache, then sends the response back to the server TOM. 3. Server TOM receives the answer and substitutes the response in place of the initial name.

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.g] if the data location server contains the location string associated with the identification string provided in the data location request, the data location server transmits location information for use by the client to calculate a location of the data associated with the identification string; <i>continued</i>	4. Server TOM contacts server JERRY and establishes a standard SQL*Net distributed database connection.” See OracleNamesAdminGuide 51-52. <i>See also</i> OracleNamesAdminGuide at 22: “The user and application developer are exposed only to the name of a network object. No knowledge of the connection path is required. This is a much more user-friendly way to use networks and network services. As with other user-oriented network software such as electronic mail, people only need to think in terms of names to get their work done. <i>When making a connection, the client specifies the name of a destination and Oracle Names provides the address.</i> Such administrative advantages are critical to managing large distributed systems. In a production environment, it’s important that users and applications be shielded from changes made to the network infrastructure. In the long run, this results in less work for the administrator and more reliability for the network.” <i>See</i> OracleNamesAdminGuide at 51-52: “Role of the Client[:]The client is the software that requests the name to be resolved. The name itself typically comes from the user’s application, or the actual user.... <i>When the name is successfully resolved, the client uses it to connect to the intended destination.</i> All interaction with the Names Server and use of the results is transparent to the user. Only an unsuccessful attempt is seen by the client or user.” <i>See</i> OracleNamesAdminGuide at 23: “An Oracle Names Server stores names and addresses for network services such as databases or MultiProtocol Interchanges, database link definitions, and object aliases. The Names Server stores the following types of objects, which are described below: • database service names—<i>The service name is the global database name that is mapped to the SQL*Net connect descriptor.</i> The combination of a listener address and connect data make up a connect descriptor that locates a database service. Gateways to non-Oracle databases and Oracle RDB databases are also stored.... • V1 connect strings—A V1 <i>connect string can be mapped to a service name in the Oracle Names database. The service name you enter acts as an alias for the SQL*Net V1 connect string,</i> making it easy to access a server in a network that still uses SQL*Net V1.... After services have been defined in Oracle Names, <i>users only need to refer to them by name to connect to a database or network service.</i> The

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.g] if the data location server contains the location string associated with the identification string provided in the data location request, the data location server transmits location information for use by the client to calculate a location of the data associated with the identification string; <i>continued</i>	necessary exchange of details occurs between the requester and the Oracle Names Server.” See OracleNamesAdminGuide at 191-192 (describing DR1 Names Server storing location information from another region in memory to service requests from client): “Figure C – 4 shows this process and the sequence of events assuming that no data has been previously cached. 1. The client sends request to preferred Names Server in its local region (DR1). 2. The preferred Names Server in DR1 issues request for the Names Servers authoritative for the destination server (in region DR2.1).  Destination Server Figure C – 4 Network Messages in a Multi-Level Foreign Name Resolution 3. The root Names Server does not have the answer (because root only knows about its direct child regions), but forwards the request to a Names Server in region DR2. 4. The receiving server in DR2 forwards the request to a Names Server in region DR2.1. 5. The name is retrieved from its authoritative Names Server in DR2.1. 6. The names Server in region DR2 caches the answer and returns it to the root. 7. The root caches the answer and returns it to the preferred Names Server in region DR1. 8. The preferred Names Server caches the answer and returns it to the client. 9. The client establishes a connection to the destination server. For the duration of the TTL, subsequent requests for the same name to the same Names Server do not require step 9.

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.g] if the data location server contains the location string associated with the identification string provided in the data location request, the data location server transmits location information for use by the client to calculate a location of the data associated with the identification string; <i>continued</i>	Also for the duration of the TTL, any subsequent requests in DR1 for names in the DR2.1 region are resolved by communicating directly with its servers.” Based on the above teachings, OracleNamesAdminGuide discloses or renders obvious that, if the Names Server contains the location information associated with the network-service-name-alias (i.e., identifier string) provided in the data location request, the Names Server transmits location information for use by the client to calculate a location of the data associated with the network-service-name-alias.¹⁸⁷ <u>Moreover, OracleSG Renders Obvious Claim Element 18.g.</u> OracleSG explains how OracleNamesAdminGuide’s SQL*Net configuration permits a server to service data location requests using easy-to-remember identifiers. <i>See</i> OracleSG at 313: “SQL*Net listeners. If you are planning a client-server installation, you need to start up SQL*Net listeners for each of the versions and protocols of SQL*Net that you are using.” <i>See</i> OracleSG at 87: “Oracle Network Manager ... allows you to generate the network configuration files for SQL*Net version 2. This version makes it much easier for users to specify a database to connect to; however, the burden of providing the translation between the easy names and the exact server and instance addresses falls on the DBA. I would not recommend configuring SQL*Net version 2 without this utility.” <i>See</i> OracleSG at 189: “When you reference tables located in a different database using SQL *Net, the format can be rather complicated. Synonyms simplify this access because you can create a simple synonym that contains the details of a complex reference string.” <i>See</i> OracleSG at 152: “The following is the SQL query that enables you to determine the location of your data files:

¹⁸⁷ *See* Greene Decl. at pp.80-84.

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey																														
[18.g] if the data location server contains the location string associated with the identification string provided in the data location request, the data location server transmits location information for use by the client to calculate a location of the data associated with the identification string; <i>continued</i>	<pre>SQL> select * from sys.dba_data_files;</pre><table><thead><tr><th>FILE_NAME</th><th>FILE_ID</th><th>TABLESPACE_NAME</th><th>BYTES</th><th>BLOCKS</th><th>STATUS</th></tr></thead><tbody><tr><td>C:\ORAWIN\ORSSI\ddrns.ora</td><td>3</td><td>ROLLBACK_DATA</td><td>3145728</td><td>1536</td><td>AVAILABLE</td></tr><tr><td>C:\ORAWIN\ORSSI\ddtemp.ora</td><td>4</td><td>TEMPORARY_DATA</td><td>2807152</td><td>1004</td><td>AVAILABLE</td></tr><tr><td>C:\ORAWIN\ORSSI\dduser.ora</td><td>2</td><td>USER_DATA</td><td>3145728</td><td>1536</td><td>AVAILABLE</td></tr><tr><td>C:\ORAWIN\ORSSI\ddsys.ora</td><td>1</td><td>SYSTEM</td><td>10485760</td><td>5120</td><td>AVAILABLE</td></tr></tbody></table><pre>SQL></pre> [.]” For the reasons set forth in Section III.E.1 on motivation to combine, a POSITA would have been motivated to have OracleNamesAdminGuide’s Names Server, if it contains the location information associated with the network-service-name-alias (i.e., identifier string) provided in the data location request, the Names Server transmits location information for use by the client to calculate a location of the data associated with the alias, based on OracleSG’s teachings. For instance, this would have resulted in the effective use of SQL*Net functionality, and improve query efficiency and optimization of Oracle7 systems to ease users’ ability to find their desired data. <i>See</i> OracleSG at 44, 52, 87, 605.¹⁸⁸	FILE_NAME	FILE_ID	TABLESPACE_NAME	BYTES	BLOCKS	STATUS	C:\ORAWIN\ORSSI\ddrns.ora	3	ROLLBACK_DATA	3145728	1536	AVAILABLE	C:\ORAWIN\ORSSI\ddtemp.ora	4	TEMPORARY_DATA	2807152	1004	AVAILABLE	C:\ORAWIN\ORSSI\dduser.ora	2	USER_DATA	3145728	1536	AVAILABLE	C:\ORAWIN\ORSSI\ddsys.ora	1	SYSTEM	10485760	5120	AVAILABLE
FILE_NAME	FILE_ID	TABLESPACE_NAME	BYTES	BLOCKS	STATUS																										
C:\ORAWIN\ORSSI\ddrns.ora	3	ROLLBACK_DATA	3145728	1536	AVAILABLE																										
C:\ORAWIN\ORSSI\ddtemp.ora	4	TEMPORARY_DATA	2807152	1004	AVAILABLE																										
C:\ORAWIN\ORSSI\dduser.ora	2	USER_DATA	3145728	1536	AVAILABLE																										
C:\ORAWIN\ORSSI\ddsys.ora	1	SYSTEM	10485760	5120	AVAILABLE																										
[18.h] if the data location server does not contain the location string¹⁸⁹ associated with the identification string,¹⁹⁰ the location server transmits a redirect message to the client, wherein the redirect message contains redirect	OracleNamesAdminGuide discloses this claim element or renders it obvious alone or in view of OracleSG and/or McGarvey. As shown regarding claim element 18.c (and incorporated herein), OracleNamesAdminGuide teaches its Names Server (i.e., location server) receives a query for a connection descriptor (i.e., location information) associated with a network-service-name-alias (i.e., identifier string). Further, in the hierarchical configuration shown below in Figure C-4, when responding to the query, OracleNamesAdminGuide explains that, if the queried Names Server lacks the requested information, the server sends a message to the requesting client containing the address of the Names Server that can service the request. OracleNamesAdminGuide																														

¹⁸⁸ *See* Greene Decl. at pp.85-86.

¹⁸⁹ While unclear, it appears claim element 18.h’s “location string” refers to the “location information” recited in claim element 18.e.

¹⁹⁰ While unclear, it appears claim element 18.h’s “identification string” refers to the “identifier string” recited in claim elements 18.b, 18.c, and 18.e.

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
information for use by the client to calculate a location of a different data location server in the plurality of data location servers, wherein the different data location server contains the location string.	teaches that this occurs when the server's name-request forwarding function is disabled, which is advantageous to lessen the workload of the server of the root region because it no longer needs to perform forwarding functions to other servers containing the requested information. <i>E.g.</i>, OracleNamesAdminGuide 137. As Mr. Greene explains, a POSITA would have understood that this message to the client is a redirect message used by the client to calculate the location of a location server different than the server at issue in claim element 18.g.¹⁹¹ See OracleNamesAdminGuide at 191: Figure C - 4 Network Messages in a Multi-Level Foreign Name Resolution See OracleNamesAdminGuide at 119-120: “Summary of NAMESCTL Commands ... SET FORWARDING_AVAILABLE [will t]urn on or off name request forwarding for a Names Server.” OracleNamesAdminGuide at 137: “SET FORWARDING_AVAILABLE Purpose: <i>Turns on or off name request forwarding for a Names Server....</i> Syntax: From the operating system: NAMESCTL SET FORWARDING_AVAILABLE OFF From NAMESCTL program: SET FORWARDING_AVAILABLE OFF... Usage: This setting is intended only for Names Servers that have Notes: no local clients and are exclusively handling requests from foreign Names Servers. This usually would only

¹⁹¹ See Greene Decl. at pp.86-87.

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.h] if the data location server does not contain the location string associated with the identification string, the location server transmits a redirect message to the client, wherein the redirect message contains redirect information for use by the client to calculate a location of a different data location server in the plurality of data location servers, wherein the different data location server contains the location string. <i>continued</i>	apply to Names Servers in the root region when the root is configured without clients or services. <i>If such a server is a performance bottleneck in cross-region request processing then disabling forwarding in that Names Server will cut its workload in half. Rather than forward the request and return the answer the Names Server simply tells the requestor the address of the Names Server that can answer the request.</i> Note that there is no overall reduction in work; the work is simply displaced from the non-forwarding Names Server to the requesting Names Server. WARNING: If FORWARDING_AVAILABLE is set to off, any clients who rely directly on that Names Server will be unable to resolve foreign names. Clients are not capable of redirecting their requests as Names Servers would. Their requests will fail at that point, even if other Names Servers are listed in the NAMES.PREFERRED_SERVERS configuration parameter. Example: NAMESCTL> SET FORWARDING_AVAILABLE OFF Request processing is now disabled.” See OracleNamesAdminGuide at 33: “The Oracle Names Server is a network service comprised of multiple components. Each Oracle Names Server without using DDO contains: • the cache—the Names cache is an in-memory database of all data required to resolve names. The cache contains the following types of data which is refreshed at different intervals: – system and topology data—includes the network data and <i>addresses of Names Servers in other administrative regions</i>. (Network data includes all communities and MultiProtocol Interchanges in the entire network, and Names Server addresses in the root administrative region.)... – non-authoritative data – data defined outside the local administrative region, for example, <i>foreign database and Names Server addresses</i>. Non-authoritative data only exists in multi-region configurations....” See OracleNamesAdminGuide at 67: “The Names Servers on the network using DDO must know about each other.”

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.h] if the data location server does not contain the location string associated with the identification string, the location server transmits a redirect message to the client, wherein the redirect message contains redirect information for use by the client to calculate a location of a different data location server in the plurality of data location servers, wherein the different data location server contains the location string. <i>continued</i>	Based on the above teachings, OracleNamesAdminGuide discloses or renders obvious that, if the Names Server doesn't contain the location information associated with the network-service-name-alias (i.e., identifier string) provided in the data location request, the Names Server transmits a redirect message to the client containing redirect information for use by the client to calculate a location of the Names Server having the location information.¹⁹² <u>Moreover, McGarvey Renders Obvious Claim Element 18.h.</u> McGarvey illustrates that it was well known in network configuration technologies that a names server receiving a location query, and if the server can't resolve that query, the server will send a redirect message pointing to another names server that can resolve the query. For instance, McGarvey teaches a system using TCP/IP address resolution to access data on its network. <i>See</i> McGarvey at 2:61-67, Fig. 1. McGarvey explains that it was known a "particular name server may contain information on only a subset of the domain namespace, but the name server typically belongs to a hierarchy of name servers, and if it does, either it communicates with other members of the hierarchy to attempt to resolve host names, <i>or it returns information referring the client to another server in the hierarchy, so that the client can redirect the query.</i>" McGarvey at 1:43-48. <i>See also</i> McGarvey at 5:3-32: "...[A]n application program on host 101 attempts to establish communication with host 102, and first issues a <i>name server query</i> The process then waits for a response to be received from any of the name servers queried or for their timeout intervals to elapse Suppose ... that a response is received before the timeout interval elapses.... If the server replies that the host is unknown, then the process waits for replies from name servers on other domains, where the host may reside in step 340. <i>If the query fails for the name servers on all of the domains, because each name server reports that the host is unknown or times out, a check is made for a local host names file in step 345. If the file is present, it is searched for the address of the remote host in step 350.</i>" Similarly, OracleNamesAdminGuide contemplates using a DNS model to organize its data throughout its distribute database based on TCP/IP-type location information: "...[M]any companies with TCP/IP networks use the Domain Name Service (DNS) model for communication on the

¹⁹² *See* Greene Decl. at pp.87-89.

Claim 18	OracleNamesAdminGuide in view of OracleSG and McGarvey
[18.h] if the data location server does not contain the location string associated with the identification string, the location server transmits a redirect message to the client, wherein the redirect message contains redirect information for use by the client to calculate a location of a different data location server in the plurality of data location servers, wherein the different data location server contains the location string. <i>continued</i>	worldwide Internet. In this model, all naming models are hierarchical from a set of base domains To adapt any example in this book to the DNS Internet model, add a single high-level domain For organizations which are on the Internet, and therefore already have one or more DNS domains, the sensible choice is to align their Oracle naming domains with their DNS domains.” OracleNamesAdminGuide at 56-57. For the reasons set forth in Section III.E.1 on motivation to combine, a POSITA would have been motivated to have OracleNamesAdminGuide’s Names Server, if it doesn’t contain the location, the server transmits a redirect message containing redirect information to the client to calculate a location of a different Names Server that has the location information, based on McGarvey’s teachings. For instance, in a larger system, it would have eased administration and balanced loads to configure a hierarchy of domains. This would have improved the ability to locate the client’s requested information by providing a redirect message to the client for calculating the location of a different Names Server. And this would have furthered the use of SQL*Net and Names Servers to respond to certain data requests, and lessened the workload of the root server by no longer needing to perform forwarding functions to other servers containing the requested information. <i>See</i> OracleNamesAdminGuide at 33, 61, 137.¹⁹³

3. OracleNamesAdminGuide in View of OracleSG and McGarvey Renders Obvious Independent Claim 17.

Claims 17 and 18 are substantially similar, except claim 17 doesn’t include claim 18’s limitation that, “if the data location server contains the location string associated with the identification string provided in the data location request, the data location server transmits location information for use by the client to calculate a location of the data associated with the identification string.” For the reasons shown above regarding claim 18, the below claim chart shows that

¹⁹³ *See* Greene Decl. at pp.89-90.

OracleNamesAdminGuide in view of OracleSG and McGarvey discloses or renders obvious each element of independent claim 17.

Claim 17	OracleNamesAdminGuide in view of OracleSG and McGarvey
[17.preamble] A system for retrieving data location information for data stored in a distributed network, the system comprising:	See discussion of claim 18's preamble. To the extent it is limiting, OracleNamesAdminGuide discloses this preamble, which is the same as claim 18's preamble.¹⁹⁴ As shown regarding claim 18's preamble, OracleNamesAdminGuide discloses a system for retrieving data location information for data stored in a distributed network.
[17.a] a plurality of data repositories configured to store data,	See discussion for claim element 18.a. As shown above for claim element 18.a, OracleNamesAdminGuide discloses a data repository configured to store data. Further, OracleNamesAdminGuide teaches that its data repository is one of a plurality of repositories configured to store data.¹⁹⁵ See OracleNamesAdminGuide at 32: Figure 2 - 1 Components of a Network Using Oracle Names with the Dynamic Discovery Option See OracleNamesAdminGuide at 28: “Oracle Names 1.0 required administrators to enter the names and <i>addresses of all databases in the network</i> into Oracle Network Manager.”

¹⁹⁴ See Greene Decl. at p.90.

¹⁹⁵ See *id.* at pp.90-91.

Claim 17	OracleNamesAdminGuide in view of OracleSG and McGarvey
[17.a] a plurality of data repositories configured to store data, <i>continued</i>	See OracleNamesAdminGuide at 35: “The following section discusses how Oracle Names 2.0 architecture is designed to function on most <i>SQL*Net networks composed of clients and databases</i>.” See OracleNamesAdminGuide at 99: “A global database link is a link that connects <i>each database in a network to all other databases</i>. This link is created automatically by the Network Manager for use with Oracle Names. Each global database link created by Network Manager connects you to every other database server in the network.” Based on those same teachings, OracleNamesAdminGuide discloses claim element 17.a.
[17.b] wherein the data is associated with a respective identifier string in each data repository;	See discussion for claim elements 18.b-18.c. As shown above for claim elements 18.b-18.c, OracleNamesAdminGuide discloses, or renders obvious alone or in view of OracleSG, the data is associated with an identifier string.¹⁹⁶ Based on those same teachings, OracleNamesAdminGuide discloses claim element 17.b or renders it obvious alone or in view of OracleSG.
[17.c] a data location se[rv]er network having a plurality of data location servers,	See discussion for claim element 18.d. As shown regarding claim element 18.d, OracleNamesAdminGuide discloses a data location server network comprising a plurality of Names Servers (i.e., data location servers).¹⁹⁷ Based on those same teachings, OracleNamesAdminGuide discloses claim element 17.c.
[17.d] each of the plurality of data location servers containing location strings associated with respective identifier strings and	See discussion for claim element 18.e. As shown regarding claim element 18.e, OracleNamesAdminGuide discloses each Names Server (i.e., location server) contains connection descriptors (i.e., location strings) associative with respective network-service-name-aliases (i.e., identifier strings), or renders this obvious alone or in view of OracleSG.¹⁹⁸

¹⁹⁶ See Greene Decl. at p.91.

¹⁹⁷ See *id.* at pp.91-92.

¹⁹⁸ See *id.* at p.92.

Claim 17	OracleNamesAdminGuide in view of OracleSG and McGarvey
[17.d] each of the plurality of data location servers containing location strings associated with respective identifier strings and <i>continued</i>	Based on those same teachings, OracleNamesAdminGuide discloses claim element 17.d or renders it obvious alone or in view of OracleSG.
[17.e] each of the plurality of data location servers having computer executable code configured to execute the following steps:	See discussion for claim elements 18.g, which is similar to claim element 17.e. As shown regarding claim element 18.g, OracleNamesAdminGuide discloses each location servers has computer executable code configured to execute steps in response to data location requests, or renders this obvious alone or in view of OracleSG.¹⁹⁹ Based on those same teachings, OracleNamesAdminGuide discloses claim element 17.e. or renders it obvious alone or in view of OracleSG.
[17.f] in response to receiving a data location request from a client to retrieve a location string associated with an identification string provided in the data location request, transmitting a redirect message to the client if the identification string is not associated with a location string at the data location server, wherein the redirect message contains information for use by the client to calculate a location of a different data location server in the plurality of data location servers, wherein	See discussion for claim elements 18.h, which is substantially similar to claim element 17.f. As shown regarding claim element 18.h, OracleNamesAdminGuide discloses responding to a data location request by, if the data location server lacks the location string associated with the identification string, the location server transmits a redirect message to the client, wherein the redirect message contains redirect information for use by the client to calculate a location of a different data location server in the plurality of data location servers, wherein the different data location server contains the location string.²⁰⁰ Further, as shown regarding claim element 18.h, OracleNamesAdminGuide renders obvious claim element 18.h in view of McGarvey's teachings. Based on those same teachings, OracleNamesAdminGuide discloses claim element 17.f or renders it obvious alone or in view of McGarvey.

¹⁹⁹ See *id.*

²⁰⁰ See Greene Decl. at pp.92-93.

Claim 17	OracleNamesAdminGuide in view of OracleSG and McGarvey
the different data location server contains the location string.	

F. SNQ 2: Dependent Claim 24 Are Obvious Based on OracleNamesAdminGuide in view of OracleSG and McGarvey and Further in View of Rajani.

Claim 24 depends from claim 18 and recites that the location information includes a hash table distributed over location servers. Rajani teaches using a hashing algorithm to organize information in a table. And as shown above in SNQ 1, the OracleNamesAdminGuide-OracleSG-McGarvey combination renders obvious claim 18. Based on OracleSG's and Rajani's teachings and the general knowledge available in the art, a POSITA would have found it obvious for OracleNamesAdminGuide's location information to include a hash table distributed over the Names Servers. Thus, the prior art establishes that claim 24 is obvious.

1. OracleSG's Teachings Render Obvious Using a Hash Table.

As the below claim chart details regarding claim 24, a POSITA would have found it obvious for OracleNamesAdminGuide's system to have the capability of using a hash table to organize location information, based on OracleSG's teachings. Before the '640 Patent, it was well known and ubiquitous for a computer network system to use a hash table to locate information stored in the system. For example, OracleSG teaches using a hash function to improve the efficiency of a query in an Oracle7 system.²⁰¹

Like OracleSG, the '640 Patent and its prosecution history establish that it was generally known in the art for a computer network system to use a hash table to locate information stored in the system. For example, the '640 Patent specifically acknowledges that it was generally

²⁰¹ See OracleSG at 152, 633.

known in the art that hashing algorithms existed to locate information in a table (e.g., the hash-pjw algorithm).²⁰² As another example, during the original examination of the '640 Patent, the Examiner took official notice that “use of a [] hashing operation in a computer networking environment was well known in the art at the time the invention was made.”²⁰³ And as another example, Applicant stated during the same examination that “[t]he disclosure illustrates this using the standard hashpjw function from a reference computer science text book.... [T]he system distributes hash-function computation, or creates a ‘distributed hash table’.”²⁰⁴

As the below claim chart details, when making forwarding decisions to other Names Servers, a POSITA would have found it obvious to configure an initial Names Server to determine another Names Server based on OracleSG’s hashing function applied to the network-service-name-alias (i.e., identifier string). To do so, a POSITA would have been motivated to have the initial Names Server locate the other Names Server containing the location information for the desired data by applying a hashing function to the data’s network-service-name-alias (i.e., identifier string). Having this configuration would have furthered OracleNamesAdminGuide’s goal to optimize search performance in its location request queries.²⁰⁵

²⁰² See '640 Patent at 15:1-21.

²⁰³ '640 History at 447-48 (02/24/2004 Office Action).

²⁰⁴ *Id.* at 522 (08/24/2004 Remarks).

²⁰⁵ See OracleNamesAdminGuide at 60; Greene Decl. ¶¶ 119-21.

2. Reasons to Modify OracleNamesAdminGuide in View of OracleSG and McGarvey and Further in View of Rajani

As the above prior art overviews show, Rajani is in the same field of art as OracleNamesAdminGuide.²⁰⁶ As further described above in Sections III.D.1 and III.D.4, OracleNamesAdminGuide and Rajani share the common goal of parsing data sets quickly. Rajani teaches how to “quickly ascertain” the location of a file in a table of file names (i.e., a names server) in a hierarchically configured storage system.²⁰⁷ Specifically, Rajani describes using a hash function to convert each file’s hierarchical pathname into a unique identifier, which is used to identify the file’s location in the network.²⁰⁸ And Rajani emphasizes that indexing location entries distributed over a hierarchical network would be especially useful as systems grow larger and more complicated, which results in location queries becoming more labor intensive on the system.²⁰⁹

Because OracleNamesAdminGuide and Rajani share the common goal of parsing data sets quickly, a POSITA would have been motivated to use Rajani’s hashing table techniques in OracleNamesAdminGuide’s Names Servers to quickly and efficiently determine the location of the requested data.²¹⁰ It was understood in the art that DBAs would actively search for resources to solve problems that may arise in the specific database environment that they are working in. As OracleSG states: “[T]ry to obtain as much knowledge as possible about the DBMS and query optimization through training courses and books. It may mean more work for you, but it could

²⁰⁶ See, e.g., OracleNamesAdminGuide at 5, 22, 52-54; Rajani at Abstract, 8:53-60, Claim 1.

²⁰⁷ Rajani at 1:32-34.

²⁰⁸ See Rajani at 8:53-60, Claim 1.

²⁰⁹ Rajani at 1:37-45.

²¹⁰ See Greene Decl. ¶¶ 122-23.

save many hours of project team labor if you can quickly provide your compatriots with the answers that they need.”²¹¹ OracleNamesAdminGuide describes one specific database environment—hierarchical configurations—that provides certain RDBMS-related advantages. But such hierarchical configurations can complicate the naming practices and location information needed to identify data. For example, in non-hierarchical configurations, Oracle7’s “flat-naming” convention allows the omission of domains.²¹² Conversely, OracleNamesAdminGuide’s hierarchical configurations include multiple domains requiring that the data’s location information identify the specific domain of the data.²¹³

Mr. Greene explains that, as a system’s configuration becomes more complex requiring more information to uniquely identify the data’s location, the storage and processing requirements for the OracleNamesAdminGuide’s Names Servers increase along with it.²¹⁴ A DBA would have been motivated to minimize these burdens and operate the Names Servers at optimal efficiency. This would have furthered OracleNamesAdminGuide’s goal of having its Names Servers identify the requested data’s location “lightning fast.”²¹⁵

Mr. Greene explains that, based on the desire to maximize efficiency in running queries, a POSITA would have been motivated to combine OracleNamesAdminGuide’s and OracleSG’s techniques of locating data with Rajani’s use of hashing location tables to quickly return location information associated with the requested data.²¹⁶

²¹¹ OracleSG at 51.

²¹² See OracleSG at 41.

²¹³ See OracleNamesAdminGuide at 41-42; Greene Decl. ¶ 123.

²¹⁴ See Greene Decl. ¶ 124.

²¹⁵ OracleNamesAdminGuide at 60; see Greene Decl. ¶¶ 124.

²¹⁶ See Greene Decl. ¶¶ 125-26.

Thus, to the extent that the combination of OracleNamesAdminGuide and OracleSG doesn't render obvious any aspect of claim 24, a POSITA would have been motivated and found it obvious to have OracleNamesAdminGuide's location information include a portion of a hash table distributed over multiple Names Servers based on Rajani's teachings. This would have furthered OracleNamesAdminGuide's goal of having its Names Servers identify the requested data's location "lightning fast."²¹⁷ The benefits of this modification to the Names Servers would become more pronounced as the complexity of the system grows, especially in hierarchical configurations that result in the data's location information including the data's specific domain.²¹⁸

Even further, and well before the '640 Patent, it was well known and ubiquitous for a computer network system to use a hash table to locate information stored in the system.²¹⁹ Like Rajani, the '640 Patent and its family prosecution history establish that it was generally known in the art to use hashing operations in a computer networking environment. For example, the '640 Patent specifically acknowledges that it was generally known in the art that hashing algorithms existed to locate information in a table (e.g., the hashpjw algorithm).²²⁰

Accordingly, as the below claim chart details, based on Rajani's teachings, a POSITA would have found it obvious for OracleNamesAdminGuide's location information to include a portion of a hash table distributed over multiple Names Servers. As Mr. Greene explains, a

²¹⁷ OracleNamesAdminGuide at 60.

²¹⁸ See *id.* at 41-42; Greene Decl. ¶ 127.

²¹⁹ See Rajani at 4:40-56; see also *id.* at 6:26-36 (teaching hashing to achieve similar results based on the particular needs of the arrangement and file structure).

²²⁰ See '640 Patent at 15:1-21; see also '640 History at 447-448 (02/24/2004 Office Action), 522 (08/24/2004 Remarks).

POSITA would have been motivated to do so because this would allow OracleNamesAdminGuide's Names Servers to use a hash table for reducing access time of the location information relative to other data structures and reduce memory allocation.²²¹

3. OracleNamesAdminGuide in View of OracleSG and McGarvey and Further in View of Rajani Renders Obvious Dependent Claim 24.

The below claim chart shows that OracleNamesAdminGuide in view of OracleSG, McGarvey, and Rajani renders obvious dependent claim 24.

Claim 24	OracleNamesAdminGuide in view of OracleSG, McGarvey, & Rajani
[24] The system of claim 18, wherein the location information comprises a portion of a hash table distributed over the plurality of data location servers.	As discussed above, OracleNamesAdminGuide in view of OracleSG and McGarvey renders obvious claim 18. Rajani's teaching further render obvious claim 24, which depends from claim 18. As Mr. Greene explains, POSITA would have understood that OracleNamesAdminGuide's system likely used hashing table distributed over Names Servers. Because such a table would have been internal to OracleNamesAdminGuide's functionalities and not configurable by users, OracleNamesAdminGuide had no need to expressly explain such hashing tables.²²² OracleSG teaches that hashing functions are one of the ways to improve the efficiency of a query. <i>See</i> OracleSG at 152. By Oracle7's release, hashing in search queries was a known technique or Oracle systems or any other database. OracleSG describes functionalities deemed most efficient in its Rule-Based Optimizer, which includes hashing functions: "The concept behind the rule-based optimizer is simple. It looks at the SQL statement to determine all the possible ways that it can execute the query. Then it selects the one on its list of queries that it thinks will be most effective. The key here is that it has a simple list of possible ways to approach the problem that is fixed for all types and sizes of tables. The following is the basic order in which the execution plans are selected by the rule-based optimizer: 1. Access a single row by using its internal row ID.... 3. Access a single row by combining a hash cluster key with a unique or primary key from a table not in the cluster.... 6. Access by hash cluster keys..." <i>See</i> OracleSG at 633.

²²¹ *See* Rajani at 1:34-56, 8:15-19; Greene Decl. ¶ 128.

²²² *See* Greene Decl. at p.98.

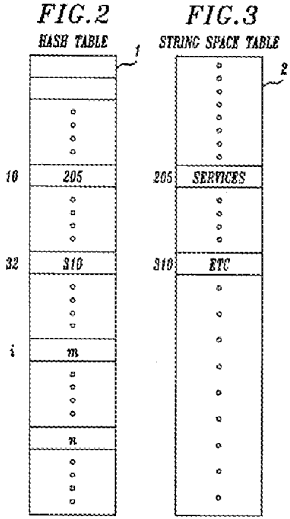
Claim 24	OracleNamesAdminGuide in view of OracleSG, McGarvey, & Rajani
[24] The system of claim 18, wherein the location information comprises a portion of a hash table distributed over the plurality of data location servers. <i>continued</i>	Further, as the '640 Patent acknowledges, it was generally known in the art that hashing algorithms existed, e.g., hashpjw, to locate information in a table: "The well-known function preferably applied is the hash pjw function presented by Aho, Sethi and Ullman in their text Compilers, Principles, Techniques and Tools: <pre> uint32_t hash (uint8_t *s, uint32_t slen, uint32_t size) { uint32_t g; uint32_t i; uint32_t h = 0; uint8_t c; for (i = 0; i < slen; i++) { c = s[i]; h = (h << 4) + c; g = (h & 0xf0000000); if (g) {^ h ^= g >> 24; h = g; } } return h % size; }." </pre> '640 Patent at 15:1-20. During the original examination, the Examiner took Official notice that "use of a [] hashing operation in a computer networking environment was well known in the art at the time the invention was made." '640 History at 447-448 (02/24/2004). And the applicants stated during examination that "[t]he disclosure illustrates this using the standard hashpjw function from a reference computer science text book.... [T]he system distributes hash-function computation, or creates a 'distributed hash table'." '640 History at 522 (08/24/2004 Remarks). Similarly, OracleNamesAdminGuide contemplates using a DNS model to organize its data throughout its distributed database based on TCP/IP-type

Claim 24	OracleNamesAdminGuide in view of OracleSG, McGarvey, & Rajani
[24] The system of claim 18, wherein the location information comprises a portion of a hash table distributed over the plurality of data location servers. <i>continued</i>	location information: "...[M]any companies with TCP/IP networks use the Domain Name Service (DNS) model for communication on the worldwide Internet. In this model, all naming models are hierarchical from a set of base domains To adapt any example in this book to the DNS Internet model, add a single high-level domain For organizations which are on the Internet, and therefore already have one or more DNS domains, the sensible choice is to align their Oracle naming domains with their DNS domains." OracleNamesAdminGuide at 56-57. Based on Oracle SG's teachings and the general knowledge available in the art, a POSITA would have found it obvious for OracleNamesAdminGuide's location information include a portion of a hash table distributed over the Names Servers.²²³ Having this configuration would further OracleNamesAdminGuide goal to optimize search performance in its location request queries. <i>See</i> OracleNamesAdminGuide at 60 ("Although any given Names Server is only engaged for a short period of time when a client initiates a connection, the goal is to keep name resolutions lightning fast.").²²⁴ <u>Moreover, Rajani Renders Obvious Claim 24.</u> To the extent the OracleNamesAdminGuide-OracleSG-McGarvey combination doesn't render obvious claim 24, a POSITA would have found it obvious based on Rajani's teachings. For instance, as described above, OracleNamesAdminGuide discloses the use of a second Names Server to improve performance results. Similarly, Rajani teaches a network having a hierarchically configured storage system distributed across multiple computers and how to "quickly ascertain" the data's location in a table of file names.²²⁵ And Rajani describes hashing the locations of data (e.g., files) to make the location entries on the table of file names more easily accessible: "A method of locating a file ... comprising the steps of ... determining the location at which said file is stored ..., storing each of said components in an available location of a first table of locations such that the address of said available location defines the unique identity value of the component stored thereat, and deriving for each of said components an associated hashing number and storing in a second table of locations at a

²²³ As discussed above in Section III.C, Patent Owner has proposed "hash table" means "a data structure that stores values in a table, where values are stored and retrieved by applying a hash function to an input and using the function result as an index into the table."

²²⁴ *See* Greene Decl. at pp.98-99.

²²⁵ Rajani at 1:32-45, Fig. 1.

Claim 24	OracleNamesAdminGuide in view of OracleSG, McGarvey, & Rajani
[24] The system of claim 18, wherein the location information comprises a portion of a hash table distributed over the plurality of data location servers. <i>continued</i>	location identified by said hashing number the address of the available location.” See Rajani at 8:37-60. See also Rajani at 4:1-15, Figs. 2-3: “Specifically, and referring now to FIG. 2 and 3, computer 110 employs in the process of locating or creating a record respectively associated with a file that is either being stored or unloaded from one memories 30-1 through 30-P a so-called hash Table (FIG. 2) and string space Table (FIG. 3). In the following discussion it will be assumed that a file is being stored in one of the memories 30-1 through 30-P, for example, memory 30-1, and that the path name of the file is /etc/services. It is also assumed that the originator of the file is computer 10-1. (It is to be understood of course that the ensuing discussion equally pertains to each of the computers 10 and equally pertains to a file having a different path name, e.g., /usr/data1, or a longer pathname, i.e., /usr/etc Mbin/services.).”  The diagrams show two vertical tables. FIG. 2, labeled 'HASH TABLE' with a pointer '1', has rows with memory addresses 10, 205, 32, 310, and 'i' on the left. The corresponding cells contain '205', '310', and 'm'. FIG. 3, labeled 'STRING SPACE TABLE' with a pointer '2', has rows with memory addresses 205 and 310 on the left. The corresponding cells contain 'SERVICES' and 'ETC'. See Rajani at 4:28-31: “Computer 10 generates each S index by first passing each component of the pathname through a conventional hashing algorithm to obtain a memory offset (i.e., h=hash ("string component"=i).” See Rajani at 4:40-44: “Computer 110 then retrieves from hash Table 1 the contents of the location defined by the offset of 32, which contents in the present illustrative example should be the S index for the component “etc”, i.e., the value of 310 as shown in FIG. 2.”

Claim 24	OracleNamesAdminGuide in view of OracleSG, McGarvey, & Rajani
[24] The system of claim 18, wherein the location information comprises a portion of a hash table distributed over the plurality of data location servers. <i>continued</i>	<i>See</i> Rajani at 4:51-56: “Computer 110 then similarly verifies that the value of 205 is the S index for the component “services”, as shown at location 10 of hash Table 1. Thus, the Sindex for a component in a path-name is the memory location in string space Table 2 containing that component.” <i>See</i> Rajani 5:40-48: “As mentioned above, the hash number and S index are used as offsets to access information stored in Tables 1 and 2, respectively, as shown in FIGS. 2 and 3. What this means is that an S index is stored in Table 1 using a group of memory bytes-illustratively four bytes. The particular group of bytes that are used for the storage of the S index in Table 1 is determined by the value of the associated hash number, i.e., the hash number is used as an offset to locate the associated S index in Table 1.” <i>See</i> Rajani 6:26-36: “The foregoing is merely illustrative of the principles of the invention. Those skilled in the art will be able to devise numerous arrangements, which, although not explicitly shown or described herein, nevertheless embody those principles that are within the spirit and scope of the invention. For example, special mapping techniques would allow the D-node and H-node Tables to be combined into one Table for each of the computers 10. In addition, the collision among hashing numbers could be dealt with using a so-called secondary hashing technique that is known in the art.” For the reasons set forth in Sections III.F.1-III.F.2 on motivation to combine, a POSITA would have found it obvious for OracleNamesAdminGuide to include a portion of location information in one Names Server based on a hash function used to organize the information across multiple Names Servers wherein each Names Server can determine the relevant Names Server based on the hash function applied to the network-service-name-alias (i.e., identifier string). A POSITA would have been motivated to do so because having this configuration would further OracleNamesAdminGuide’s goal to optimize search performance in its location request queries and to have its Names Servers identify the requested data’s location “lightning fast.” <i>See</i> OracleNamesAdminGuide at 60. The benefits of this modification to the Names Servers would become more pronounced as the complexity of the system grows, especially in hierarchical configurations that result in the data’s location information including the data’s specific domain. <i>See</i> OracleNamesAdminGuide at 41-42.

Claim 24	OracleNamesAdminGuide in view of OracleSG, McGarvey, & Rajani
[24] The system of claim 18, wherein the location information comprises a portion of a hash table distributed over the plurality of data location servers. <i>continued</i>	Thus, based on Rajani's teachings, a POSITA would have been motivated to have OracleNamesAdminGuide's information include a portion of a hash table distributed over the plurality of Names Servers. As explained in Sections III.F.1-III.F.2 on motivation to combine, using a hash table in OracleNamesAdminGuide would have reduced access time of the location information relative to other data structures and reduce memory allocation. <i>See</i> Rajani at 1:34-56, 8:15-19.²²⁶

G. Secondary Considerations

This Request demonstrates that the Challenged Claims of the '640 Patent are unpatentable as obvious in view of the prior art references. In the related district court matter, Patent Owner has argued that the purported existence of copying, long-felt-but unsolved-need, failure of others, commercial success and industry praise, and skepticism of skilled artisans weighs against a finding of obviousness.²²⁷ But these alleged secondary indicia of non-obviousness aren't sufficient to overcome this Request's showing of obviousness for at least two reasons.

First, the Federal Circuit has made it clear that a nexus must exist between an indicium of non-obviousness and the claimed invention.²²⁸ For example, *In re Kao* explains that, "[w]here the offered secondary consideration actually results from something other than what is both claimed and novel in the claim, there is no nexus to the merits of the claimed invention."²²⁹ As another example, *Fox Factory* specifies that, "[a]lthough we do not require the patentee to prove

²²⁶ *See* Greene Decl. at pp.100-02.

²²⁷ *See* Ex. 22 (Final Validity Contentions) at 4-5.

²²⁸ *In re Kao*, 639 F.3d 1057, 1068 (Fed. Cir. 2011); *Tokai Corp. v. Easton Enters., Inc.*, 632 F.3d 1358, 1369 (Fed. Cir. 2011); *Ormco Corp. v. Align Tech., Inc.*, 463 F.3d 1299, 1312 (2006).

²²⁹ 639 F.3d at 1068.

perfect correspondence to meet the coextensiveness requirement, what *we do require is that the patentee demonstrate that the product is essentially the claimed invention.*”²³⁰ In the related district court case, Patent Owner has presented evidence relating only generally to distributed systems. But Patent Owner has failed to present the required objective evidence of non-obviousness having *a nexus* to the claimed invention or that the product-at-issue is *essentially the claimed invention*.

Second, Patent Owner has never shown that any sold commercial product actually practices the claimed invention. Without such a showing, Patent Owner’s purported evidence toward copying, commercial success, industry praise, and skepticism isn’t relevant to the non-obviousness analysis. Indeed, the Federal Circuit has pronounced that “to determine whether a nexus exists, our cases require consideration of whether the marketed product embodies the claimed features.”²³¹

As explained below, Patent Owner hasn’t produced evidence that a commercial product embodies the claimed invention or that there is a nexus between the claimed invention and the purported secondary indicia of non-obviousness:

Copying: To establish copying, the MPEP states that a patentee must show that others in the field copied the claimed invention after failing to design a product based on the prior art.²³² But Patent Owner has previously made only nebulous assertions about the similarities between the claimed invention and other commercial products. More specifically, Patent Owner has only

²³⁰ *Fox Factory, Inc. v. SRAM LLC*, 944 F.3d 1366, 1374 (Fed. Cir. 2019) (emphasis added).

²³¹ *Bosch Automotive Service Solutions, LLC v. Matal*, 878 F.3d 1027, 1036-37 (Fed. Cir. 2017).

²³² See MPEP 716.06.

claimed that its non-adjudicated infringement allegations in the related district court matter are evidence of copying, but provided no further evidence.

Long-Felt Need and Failure of Others: To show this secondary consideration, the MPEP states that a patentee must present evidence of (a) a persistent need recognized by POSITAs, (b) others not satisfying the purported need before the claimed invention, or (c) the claimed invention actually satisfying the long-felt need.²³³ But Patent Owner has only previously referred to others (e.g., U.S. Patent No. 5,542,087 to Neimat) discussing the general need for hashing schemes and load balancing in distributed systems.

Commercial Success and Industry Praise: Patent Owner hasn't met its burden of proof in establishing a nexus between the claimed invention and any evidence of commercial success. And Patent Owner has failed to establish that any purported success isn't the result of promotion or advertising, shift in advertising, consumption by purchasers normally tied to sellers, or other business events extraneous to the merits of the claimed invention, as the MPEP requires.²³⁴ Further, Patent Owner has failed to present any evidence of persons praising the merits of the claimed invention rather than praising general concepts like distributed data storage. And Patent Owner hasn't presented any evidence that it has licensed the claim invention to any third party.

Skepticism of Skilled Artisans: The MPEP establishes that, to show skepticism, a patentee must present evidence of actual skepticism of the claimed invention.²³⁵ But Patent Owner has previously made only nebulous hearsay assertions about audience members' reception of a presentation at Yale University.

²³³ See MPEP 716.04.

²³⁴ See MPEP 716.03.

²³⁵ See MPEP 716.05.

As such, the purported existence of copying, long-felt need, failure of others, commercial success, industry praise, and skepticism doesn't overcome the conclusion of obviousness.²³⁶

IV. CONCLUSION

Requester respectfully submits that OracleNamesAdminGuide in view of OracleSG and McGarvey raises a substantial new question of patentability for independent claims 17 and 18, and that OracleNamesAdminGuide in view of OracleSG, McGarvey, and Rajani raises a substantial new question of patentability for dependent claim 24. As such, the Requester requests reexamination of claims 17, 18, and 24.

Dated: November 19, 2021

Respectfully submitted,

/Joseph F. Edell /

Joseph F. Edell (Reg. No. 67,625)

FISCH SIGLER LLP

5301 Wisconsin Avenue NW

Fourth Floor

Washington, DC 20015

Phone: 202.362.3524

Email: Joe.Edell.IPR@fischllp.com

²³⁶ See Greene Decl. ¶ 130.