



An Oracle White Paper
June 2013

Oracle Single Client Access Name (SCAN)

Introduction	1
Network Requirements for Using SCAN	2
Option 1 – Use the Corporate DNS	2
Option 2 – Use the Oracle Grid Naming Service (GNS).....	4
Workaround if No DNS Server is Available at Installation Time	4
SCAN Configuration with Oracle Grid Infrastructure 11g Release 2 ..	5
SCAN Configuration with Oracle Grid Infrastructure 12c Release 1...	6
Enabling Multiple-Subnet Support for SCAN.....	7
Oracle Database Configuration Using SCAN.....	10
Client Load Balancing using SCAN	10
Multiple-Subnet Support and LISTENER_NETWORKS.....	11
Version and Backward Compatibility	12
Miscellaneous SCAN-related Configurations	13
Using SCAN with Multiple Ports on the Same Subnet.....	13
Using SCAN in a MAA Environment not using GDS	14
Using SCAN in a MAA Environment using GDS	14
Using SCAN with Oracle Connection Manager.....	15
Summary and Conclusion.....	15

Introduction

Single Client Access Name (SCAN) is a feature used in Oracle Real Application Clusters environments that provides a single name for clients to access any Oracle Database running in a cluster. You can think of SCAN as a cluster alias for databases in the cluster. The benefit is that the client's connect information does not need to change if you add or remove nodes or databases in the cluster.

SCAN was first introduced with Oracle Real Application Clusters (RAC) 11g Release 2 and provides additional functionality in Oracle RAC 12c. Having a single name to access the cluster to connect to a database in this cluster allows clients to use EZConnect and the simple JDBC thin URL to access any database running in the cluster, independently of the number of databases or servers running in the cluster and regardless on which server(s) in the cluster the requested database is actually active.

```
EZconnect      sqlplus system/manager@sales1-scan:1521/oltp
JDBC connect  jdbc:oracle:thin:@sales1-scan:1521/oltp
```

Example 1: Sample EZConnect and Thin JDBC Connect Strings

Network Requirements for Using SCAN

The default SCAN configuration is defined during the installation of Oracle Grid Infrastructure that is distributed with Oracle Database 11g Release 2 or higher. Oracle Grid Infrastructure is a single Oracle Home that contains Oracle Clusterware and Oracle Automatic Storage Management.

You must install Oracle Grid Infrastructure first in order to use Oracle RAC 11g Release 2 or higher. During the interview phase of the Oracle Grid Infrastructure installation, you will be prompted to provide a SCAN name. There are 2 options for defining the SCAN:

1. Define a SCAN using the corporate DNS (Domain Name Service)
2. Define a SCAN using the Oracle Grid Naming Service (GNS)

Option 1 – Use the Corporate DNS

If you choose Option 1, you must ask your network administrator to create at least one single name that resolves to three IP addresses using a round-robin algorithm. Three IP addresses are recommended considering load balancing and high availability requirements regardless of the number of servers in the cluster.

The IP addresses must be on the same subnet as your default public network in the cluster. The name must be 15 characters or less in length, not including the domain, and it must be resolvable without the domain suffix (for example: “sales1-scan” must be resolvable as opposed to “scan1-can.example.com”). The IPs must not be assigned to a network interface, since Oracle Clusterware will take care of it.

sales1-scan.example.com	IN A	133.22.67.194
	IN A	133.22.67.193
	IN A	133.22.67.192

Example 2: Sample DNS entry for SCAN

You can check the SCAN configuration in DNS using “nslookup”. If your DNS is set up to provide round-robin access to the IPs resolved by the SCAN entry, then run the “nslookup” command at least twice to see the round-robin algorithm work. The result should be that each time, the “nslookup” would return a set of three IPs in a different order.

First nslookup	Second nslookup
<pre>[oracle@mynode] nslookup sales1-scan Server: 131.32.249.41 Address: 131.32.249.41#53 Non-authoritative answer: Name: sales1-scan.example.com Address: 133.22.67.192 Name: sales1-scan.example.com Address: 133.22.67.193 Name: sales1-scan.example.com Address: 133.22.67.194</pre>	<pre>[oracle@mynode] nslookup sales1-scan Server: 131.32.249.41 Address: 131.32.249.41#53 Non-authoritative answer: Name: sales1-scan.example.com Address: 133.22.67.193 Name: sales1-scan.example.com Address: 133.22.67.194 Name: sales1-scan.example.com Address: 133.22.67.192</pre>

Example 3: Look up the SCAN configuration in DNS using “nslookup”

Note: If your DNS server does not return a set of three IPs as shown in figure 3 or does not round-robin, ask your network administrator to enable such a setup. Round-robin on DNS level allows for a connection request load balancing across SCAN listeners floating in the cluster. It is not required for SCAN to function as a whole and the absence of such a setup will not prevent the failover of a connection request to another SCAN listener, in case the first SCAN listener in the list is down.

The Oracle Client typically handles failover of connections requests across SCAN listeners in the cluster. Oracle Clients of version Oracle Database 11g Release 2 or higher will not require any special configuration to provide this type of failover. Older clients require considering additional configuration¹. It is therefore **recommended** that the **minimum version of the client** used to connect to a database using SCAN is of version **Oracle Database 11g Release 2 or higher**.

Using client-side DNS caching may generate a false impression that DNS round-robin is not occurring from the DNS server. (DNS not return a set of three IPs as shown in figure 3). Client-side DNS caches are typically used to minimize DNS requests to an external DNS server as well as to minimize DNS resolution time. This is a simple recursive DNS server with local items.

If the client-side DNS cannot be set up to provide round-robin locally or cannot be disabled, Oracle Clients using a JDBC:thin connect will typically attempt a connection to the SCAN-IP and SCAN-listener which is returned first in the list. This basically disables the connection request load balancing across SCAN listeners in the cluster from those clients, but does not affect SCAN functionality as a whole. Oracle Call Interface (OCI) based database access drivers will apply an internal round-robin algorithm and do not need to be considered in this case.

¹ See *Oracle Client and Oracle Database Version Compatibility for SCAN* in this paper for more information.

Option 2 – Use the Oracle Grid Naming Service (GNS)

If you choose option 2, you only need to enter the SCAN name during the interview. At some stage in the cluster configuration, three IP addresses will be acquired from either a DHCP service or using “Stateless Address AutoConfiguration” (SLAAC) when using IPv6 based IP addresses with Oracle RAC 12c (using GNS, however, assumes that you use some form of dynamic IP assignment on your public network) to create the SCAN. SCAN name resolution will then be provided by the GNS².

Workaround if No DNS Server is Available at Installation Time

Oracle Universal Installer (OUI) enforces providing a default SCAN resolution during the Oracle Grid Infrastructure installation, since the SCAN concept is an essential part during the creation of Oracle RAC 11g Release 2 or higher databases in the cluster. All Oracle Database 11g Release 2 or higher tools used to create a database (e.g. the Database Configuration Assistant (DBCA), or the Network Configuration Assistant (NetCA)) would assume its presence. Hence, OUI will not let you continue with the installation until you have provided a suitable SCAN resolution.

However, in order to overcome the installation requirement without setting up a DNS-based SCAN resolution, you can use a hosts-file based workaround. In this case, you would use a typical hosts-file entry to resolve the SCAN to only 1 IP address and one IP address only. It is not possible to simulate the round-robin resolution that the DNS server does using a local host file. The host file look-up the OS performs will only return the first IP address that matches the name. Neither will you be able to do so in one entry (one line in the hosts-file). Thus, you will create only 1 SCAN for the cluster. (Note that you will have to change the hosts-file on all nodes in the cluster for this purpose.)

This workaround might also be used when performing an upgrade from former (pre-Oracle Database 11g Release 2 or higher) releases. However, it is strongly recommended to enable the SCAN configuration as described under “Option 1” or “Option 2” in this paper shortly after the upgrade or the initial installation. In order to make the cluster aware of the modified SCAN configuration, delete the entry in the hosts-file and then issue: `srvctl modify scan -n <scan_name>` as the root user on one node in the cluster.

The scan_name provided can be the existing fully qualified name (or a new name), but should be resolved through DNS, having 3 IPs associated with it, as discussed. The remaining re-configuration is then performed automatically.

² For details on how to install a cluster using the Grid Naming Service, see the Oracle® Grid Infrastructure Installation Guide 11g Release 2 (11.2) http://download.oracle.com/docs/cd/E11882_01/install.112/e10812/prelinux.htm#BABFDGHJ

SCAN Configuration with Oracle Grid Infrastructure 11g Release 2

During cluster configuration, several resources are created in the cluster for SCAN. For each of the 3 IP addresses that the SCAN resolves to, a SCAN VIP resource is created and a SCAN Listener is created. The SCAN Listener is dependent on the SCAN VIP and the 3 SCAN VIPs (along with their associated listeners) will be dispersed across the cluster.

This means, each pair of resources (SCAN VIP and Listener) will be started on a different server in the cluster, assuming the cluster consists of three or more nodes.

In case, a 2-node-cluster is used (for which 3 IPs are still recommended for simplification reasons), one server in the cluster will host two sets of SCAN resources under normal operations. If the node on which a SCAN VIP is running fails, the SCAN VIP and its associated listener will fail over to another node in the cluster. If by means of such a failure the number of available servers in the cluster becomes less than three, one server would again host two sets of SCAN resources. If a node becomes available in the cluster again, the formerly mentioned dispersion will take effect and relocate one set accordingly.

```
[grid@mynode] srvctl config scan_listener
SCAN Listener LISTENER_SCAN1 exists. Port: TCP:1521
SCAN Listener LISTENER_SCAN2 exists. Port: TCP:1521
SCAN Listener LISTENER_SCAN3 exists. Port: TCP:1521

[grid@mynode] srvctl config scan
SCAN name: sales1-scan, Network: 1/133.22.67.0/255.255.252.0/
SCAN VIP name: scan1, IP: /sales1-scan.example.com/133.22.67.192
SCAN VIP name: scan2, IP: /sales1-scan.example.com/133.22.67.193
SCAN VIP name: scan3, IP: /sales1-scan.example.com/133.22.67.194
```

Example 4: Sample SCAN configuration in Oracle Grid Infrastructure 11g Release 2

SCAN Configuration with Oracle Grid Infrastructure 12c Release 1

Most of the SCAN design principles as outlined for Oracle Grid Infrastructure 11g Release 2 remain with Oracle Grid Infrastructure 12c. However, based on customer requirements and feedback, the SCAN concept has been enhanced with Oracle Grid Infrastructure 12c as follows:

1. SCAN and Oracle Clusterware managed VIPs now support IPv6 based IP addresses
2. SCAN is by default restricted to only accept service registration from nodes in the cluster
3. SCAN supports multiple subnets in the cluster (one SCAN per subnet)

Only the default SCAN (on the default network, typically network number 1) can be installed and configured during the OUI-based installation of the Oracle Grid Infrastructure. Multiple subnet-support in the cluster needs to be enabled as a post-installation task.

These enhancements require changes in the configuration of the SCAN and SCAN_LISTENER:

```
[grid@mynode]$ srvctl config scan
SCAN name: sales1-scan.example.com, Network: 1
Subnet IPv4: 133.22.67.0/255.255.252.0/eth0
Subnet IPv6:
SCAN 0 IPv4 VIP: 133.22.67.194
SCAN name: sales1-scan.example.com, Network: 1
Subnet IPv4: 133.22.67.0/255.255.252.0/eth0
Subnet IPv6:
SCAN 1 IPv4 VIP: 133.22.67.193
SCAN name: sales1-scan.example.com, Network: 1
Subnet IPv4: 133.22.67.0/255.255.252.0/eth0
Subnet IPv6:
SCAN 2 IPv4 VIP: 133.22.67.192

[grid@mynode]$ srvctl config scan_listener
SCAN Listener LISTENER_SCAN1 exists. Port: TCP:1521
Registration invited nodes:
Registration invited subnets:
SCAN Listener LISTENER_SCAN2 exists. Port: TCP:1521
Registration invited nodes:
Registration invited subnets:
SCAN Listener LISTENER_SCAN3 exists. Port: TCP:1521
Registration invited nodes:
Registration invited subnets:
```

Example 5: SRVCTL output examples

As you can see from the example output shown in Example 5, supporting IPv6 based IPs for SCAN (and thereby the Node VIPs) is an essential concept in Oracle Grid Infrastructure 12c SCAN. In example 5, IPv6 based IP addresses are not used and only one SCAN has been deployed in the cluster on network number one, which is assigned to the Network Interface Card (NIC) eth0.

The example output in Example 5 also shows traces of the new security features in SCAN. By default, SCAN in Oracle Grid Infrastructure 12c will only accept a registration of database instances (as part of the Dynamic Database Instance registration with listeners) that are part of the cluster that a SCAN listener is assigned to.

Note: In order to only allow database instance registration of databases hosted in the cluster, the default registration of database instances with their assigned SCAN(s) uses the private interconnect. If – for some reason – databases that cannot access or do not use the private interconnect that is defined for the cluster need to be registered with a respective SCAN listener, the SCAN listener can be modified to allow additional registrations on a per (named) node or subnet basis as illustrated in Example 6 below. Note that invited nodes do not need to be present in the cluster when adding them to the list in order to pre-approve nodes that might join in future.

```
[grid@mynode]$ srvctl modify scan_listener -invitednodes test -update
[grid@mynode]$ srvctl config scan_listener
SCAN Listener LISTENER_SCAN1 exists. Port: TCP:1521
Registration invited nodes: test
Registration invited subnets:
SCAN Listener LISTENER_SCAN2 exists. Port: TCP:1521
Registration invited nodes: test
Registration invited subnets:
SCAN Listener LISTENER_SCAN3 exists. Port: TCP:1521
Registration invited nodes: test
Registration invited subnets:
```

Example 6: Modify SCAN Listener to allow registration from additional node(s)

Enabling Multiple-Subnet Support for SCAN

In order to use a SCAN on more than one subnet in the cluster, you must first enable the use of multiple subnets in the cluster, which in general is a post-installation task, including the following steps:

1. Create additional subnet(s) on the (public) network(s)
2. Assign node VIPs to newly created subnet(s)
3. Create a node Listener for newly created subnet(s)
4. Create (a) SCAN(s) using the option one or two

Step 1 assumes that you either have multiple network interface cards (NICs) that you plan on using for the public network communication in your Oracle RAC cluster, or you have one (set) of interface(s) that you want to use with multiple IP addresses from various network. Both can be established in various ways, while in either case, the Oracle Clusterware layer should be able to see multiple interfaces as determined by the `oifcfg` command as shown in Example 7.

```

[grid@mynode]$ oifcfg iflist
eth0 133.22.67.0
eth1 192.168.122.0
eth1 169.254.0.0
eth2 10.1.1.0

[grid@mynode]$ oifcfg setif -global eth2/10.1.1.0:public
[grid@mynode]$ oifcfg getif
eth0 133.22.67.0 global public
eth1 192.168.122.0 global cluster_interconnect,asm
eth2 10.1.1.0 global public

[root@mynode]# srvctl add network -netnum 2
-subnet 10.1.1.0/255.255.252.0/eth2
[root@mynode]# srvctl config network -netnum 2
Network 2 exists
Subnet IPv4: 10.1.1.0/255.255.252.0/eth0, static
Subnet IPv6:

```

Example 7: Create additional subnets on the public network

Steps 2 to 4 from above list use Oracle Clusterware standard commands as shown in Example 8.

```

#ADD node VIPs
[root@mynode]# srvctl add vip -node SalesSrv145 -netnum 2 -address SalesSrv
145v2/255.255.252.0
[root@mynode]# srvctl add vip -node SalesSrv146 -netnum 2 -address SalesSrv
146v2/255.255.252.0
[root@mynode]# srvctl add vip -node SalesSrv147 -netnum 2 -address SalesSrv
147v2/255.255.252.0
[root@mynode]# srvctl add vip -node SalesSrv148 -netnum 2 -address SalesSrv
148v2/255.255.252.0

#ADD node listener on network number 2
[grid@mynode]$ srvctl add listener -listener ListNet2 -netnum 2 -endpoints "TCP:1528"

#ADD SCAN on network number 2
[root@mynode]# srvctl add scan -scanname scantest -netnum 2

#START node VIPs
[root@mynode]# srvctl start vip -vip SalesSrv145v2
[root@mynode]# srvctl start vip -vip SalesSrv146v2
[root@mynode]# srvctl start vip -vip SalesSrv147v2
[root@mynode]# srvctl start vip -vip SalesSrv148v2

#START ListNet2 node listener on network number 2
[grid@mynode]$ srvctl start listener -listener ListNet2
[grid@mynode]$ srvctl status listener -listener ListNet2
Listener LISTNET2 is enabled
Listener LISTNET2 is running on node(s): SalesSrv145, SalesSrv146, SalesSrv147 ...

#START SCAN on network number 2
[root@mynode]# srvctl start scan -netnum 2

#Check configuration and status for SCAN
[root@mynode]# srvctl config scan -netnum 2
SCAN name: scantest, Network: 2
Subnet IPv4: 10.1.1.0/255.255.252.0/eth2
Subnet IPv6:
SCAN 0 IPv4 VIP: 10.1.1.250

[root@mynode]# srvctl status scan -netnum 2
SCAN VIP scan1_net2 is enabled
SCAN VIP scan1 net2 is running on node SalesSrv146

```

Example 8: Assign Node VIPs and Create SCAN on the newly created subnet

Notes for Examples 7 and 8:

1. Operations introduced with “[root@mynode]” need to be executed as a privileged user
 - Operations introduced with “[root@mynode]” may be executed as a privileged user
2. It is assumed that the names in “-address” as part of the “srvctl add vip” command resolve to a valid IP address either via DNS or /etc/hosts. Example:
 - SalesSrv 145v2 needs resolve to a valid and unused IP address in the 10.1.1.0 subnet
3. SCAN “scantest.example.com” was not created using Option 1 or Option 2
 - Instead, “Workaround if No DNS Server is Available at Installation Time” was used

Note also that adding the SCAN to network number 2, does not automatically create a SCAN listener on this network. In order to make full use of SCAN on network number 2, you need to create and start a SCAN listener on this network.

```
#ADD SCAN_LISTENER to network number 2
[grid@mynode]$ srvctl add scan_listener -netnum 2 -listener testscan
-endpoints TCP:1528

[grid@mynode]$ srvctl config scan_listener -netnum 2
SCAN Listener TESTSCAN_SCAN1_NET2 exists. Port: TCP:1528
Registration invited nodes:
Registration invited subnets:

[grid@mynode]$ srvctl start scan_listener -netnum 2

[grid@mynode]$ srvctl status scan_listener -netnum 2
SCAN Listener TESTSCAN_SCAN1_NET2 is enabled
SCAN listener TESTSCAN_SCAN1_NET2 is running on node SalesSrv146
```

Example 9: Adding a SCAN listener to the newly added SCAN configuration on netnum 2

Now that you have created another SCAN setup using a different subnet for public communication and created a respective node VIPs, node listener and SCAN listener, you will need to inform the database(s) using these SCANS how to register with them. The principles of databases instances registering with listeners and in particular with SCAN listeners has not changed with Oracle Database 12c compared to Oracle Database 11g Release 2 and will therefore be discussed in the generic part of this paper under the section “Multiple-Subnet Support and LISTENER_NETWORKS”.

Oracle Database Configuration Using SCAN

Ever since Oracle Database 11g Release 2, SCAN is an essential part of the Oracle RAC database configuration and therefore the `REMOTE_LISTENER` parameter is set to the SCAN per default, assuming that the database is created using standard Oracle tools (e.g. the formerly mentioned DBCA). This allows the instances to register with the SCAN Listeners as remote listeners to provide information on what services are being provided by the instance, the current load and a recommendation on how many incoming connections should be directed to the instance.

In this context, the `LOCAL_LISTENER` parameter must be considered. The `LOCAL_LISTENER` parameter should be set to the node-VIP. If you need fully qualified domain names, ensure that `LOCAL_LISTENER` is set to the fully qualified domain name (e.g. `node-VIP.example.com`). By default, a node listener is created on each node in the cluster during cluster configuration. If you want to use multiple subnets with your cluster, additional node-VIPs should be configured after the initial installation as discussed in section “Enabling Multiple-Subnet Support for SCAN” of this paper.

Starting with Oracle Grid Infrastructure 11g Release 2 the node listener run out of the Oracle Grid Infrastructure home and listens on the node-VIP using the specified port (default port is 1521).

Unlike in former database versions, it is not recommended to set your `REMOTE_LISTENER` parameter to a server side TNSNAMES alias that resolves the host to the SCAN (e.g: `HOST=sales1-scan`) in the address list entry, but use the simplified “SCAN:port” syntax as shown in Example 10.

NAME	TYPE	VALUE
<code>local_listener</code>	string	<code>(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP) (HOST=133.22.67.111) (PORT=1521))))</code>
<code>remote_listener</code>	string	<code>sales1-scan.example.com:1521</code>

Example 10: LOCAL and REMOTE_LISTENER default configuration

Note: if you use the easy connect naming method, you may need to modify your `SQLNET.ORA` to ensure that `EZCONNECT` is in the list when specifying the order of the naming methods used for the client name resolution lookups (`NAMES DIRECTORY_PATH=(tnsnames, ldap, ezconnect)`).

Client Load Balancing using SCAN

For clients connecting using Oracle SQL*Net 11g Release 2, three IP addresses will be received by the client by resolving the SCAN name through DNS as discussed. The client will then go through the list it receives from the DNS and try connecting through one of the IPs received. If the client receives an error, it will try the other addresses before returning an error to the user or application. This is similar to how client connection failover works in previous releases when an address list is provided in the client connection string.

When a SCAN Listener receives a connection request, the SCAN Listener will check for the least loaded instance providing the requested service. It will then re-direct the connection request to the local listener on the node where the least loaded instance is running. Subsequently, the client will be given the address of the local listener. The local listener will finally create the connection to the database instance.

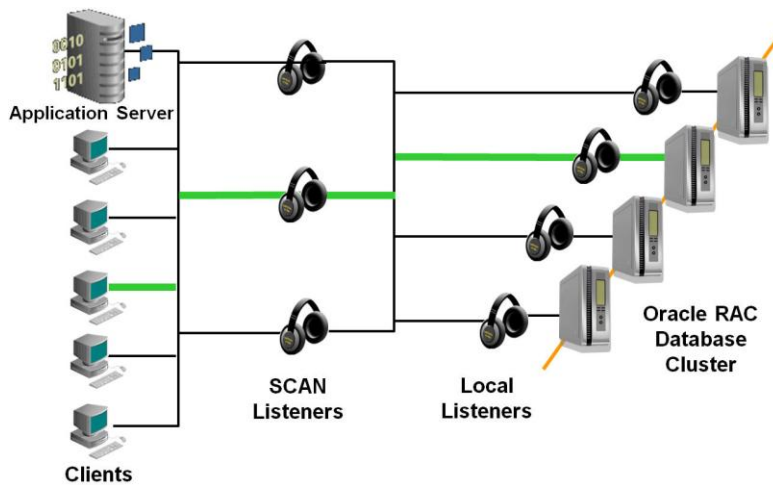


Figure 1: Connection Load Balancing using SCAN - Illustrated

Multiple-Subnet Support and LISTENER_NETWORKS

The Oracle 11g Release 2 documentation³ for the LISTENER_NETWORKS parameter says: “A network may contain multiple local and remote listeners. By default, all listeners are cross-registered with each other. By specifying a set of listeners in the LISTENER_NETWORKS initialization parameter, you can designate a subset of local listeners with a subset of remote listeners. Listeners specified by the LISTENER_NETWORKS parameter should not be specified by the LOCAL_LISTENER and REMOTE_LISTENER parameters.”

When using SCANs on multiple subnets, the LISTENER_NETWORKS parameter must be set for all databases that are meant to be reached via different subnets with Oracle Grid Infrastructure 11g Release 2. **With Oracle Grid Infrastructure 12c, the Oracle Database Agent manages the LISTENER_NETWORKS parameter as required by default.**

For more information on the LISTENER_NETWORKS parameter, see the Oracle documentation.

Version and Backward Compatibility

The successful use of SCAN to connect to an Oracle RAC database in the cluster depends on the ability of the client to understand and use the SCAN as well as on the correct configuration of the REMOTE_LISTENER parameter setting in the database as discussed. If the version of the Oracle Client connecting to the database as well as the Oracle Database version used are both Oracle Database 11g Release 2 or higher and the default configuration is used as described in this paper, no changes to the system are typically required.

The same holds true, if the Oracle Client version and the version of the Oracle Database that this client is connecting to are both pre-11g Release 2 version (e.g. Oracle Database 11g Release 1 or Oracle Database 10g Release 2, or older). In this case, the pre-11g Release 2 client would use a TNS connect descriptor that resolves to the node-VIPs of the cluster, while the Oracle pre-11g Release 2 database would still use a REMOTE_LISTENER entry pointing to the node-VIPs.

The disadvantage of this configuration is that SCAN would not be used and hence the clients are still exposed to changes every time the cluster changes in the backend. Similarly, if an Oracle Database 11g Release 2 or higher is used, but the clients remain on a former version. The solution is to change the Oracle client and / or Oracle Database REMOTE_LISTENER settings accordingly.

The following cases need to be considered:

Oracle Client Version	Oracle Database Version	Comment
Oracle Database 11g Release 2 or higher	Oracle Database 11g Release 2 or higher	No change required.
Oracle Database 11g Release 2 or higher	Pre- Oracle Database 11g Release 2	Add the SCAN VIPs as hosts to the REMOTE_LISTENER parameter.
Pre- Oracle Database 11g Release 2	Oracle Database 11g Release 2 or higher	Change the client TNSNAMES.ora to include the SCAN VIPs (* see below). IF the database was upgraded using the DBUA from a pre-11g Rel. 2 database, the DBUA will configure the REMOTE_LISTENER parameter to point to the node-VIPs as well as the SCAN.
Pre- Oracle Database 11g Release 2	Pre- Oracle Database 11g Release 2	If you want to make use of SCAN (recommended): add the SCAN VIPs as hosts to the REMOTE_LISTENER parameter. AND Change the client TNSNAMES.ora to include the SCAN VIPs (* see below). Otherwise, no change required.

Table 1: Oracle Client and Oracle Database Version Compatibility for SCAN

Note: If using a pre-11g Release 2 client (Oracle Database 11g Release or Oracle Database 10g Rel. 2, or older) you will not fully benefit from the advantages of SCAN.

Reason: The Oracle Client will not be able to handle a set of three IPs returned by the DNS for SCAN. Hence, it will try to connect to only the first address returned in the list and will more or less ignore the others. If the SCAN Listener listening on this specific IP is not available or the IP itself is not available, the connection will fail. In order to ensure load balancing and connection failover with pre-11g Release 2 clients, you will need to change the TNSNAMES.ora of the client so that it would use 3 address lines, where each address line resolves to one of the SCAN VIPs.

```
sales.example.com =(DESCRIPTION=
  (ADDRESS_LIST= (LOAD_BALANCE=on) (FAILOVER=ON)
    (ADDRESS=(PROTOCOL=tcp) (HOST=133.22.67.192) (PORT=1521))
    (ADDRESS=(PROTOCOL=tcp) (HOST=133.22.67.193) (PORT=1521))
    (ADDRESS=(PROTOCOL=tcp) (HOST=133.22.67.194) (PORT=1521)))
  (CONNECT_DATA=(SERVICE_NAME= salesservice.example.com)))
```

Example 11: TNSNAMES.ora for Oracle Database pre- 11g Release 2 Clients

Miscellaneous SCAN-related Configurations

SCAN can be used in multiple configurations. This paper describes the recommended setup, which does not exclude other configurations, as long as you are aware of their implications. The following section discusses some less-common, configurations and will be updated over time as needed.

Using SCAN with Multiple Ports on the Same Subnet

For the purpose of allowing applications to maintain a distinguishable connect string on host / port level using one SCAN as the host or to enable port-based firewall configurations, SCAN can be set up to support more than one port for the same subnet.

It needs to be noticed that different ports are only used for an initial connect. Once a connection using SCAN on any SCAN port has been established, port assignment is not maintained and routing of connections within the Oracle RAC cluster is performed using database services registered with one or more local listeners.

This means that all SCAN ports are treated the same, so that a connection that comes in on any SCAN port can be directed to any registered local listener on any port within the same subnet.

Note also that the multi-port assignment differs per version. Oracle Grid Infrastructure 11.2.0.3 would not let you set up multiple ports on one SCAN anymore, while Oracle Grid Infrastructure 11.2.0.2 would have let you use this configuration. Support for this configuration is resumed with Oracle Grid Infrastructure 11.2.0.3.6 and higher as well as Oracle Grid Infrastructure 12c.

Using SCAN in a MAA Environment not using GDS

If you have implemented a Maximum Availability Architecture (MAA) environment without Global Data Services (GDS), a new capability available with Oracle Database 12c that extends intelligent load balancing and client failover concepts to globally distributed environments, in which you use Oracle RAC for both your primary and standby database (in both, your primary and standby site), which are synchronized using Oracle Data Guard, using SCAN provides a simplified TNSNAMES configuration that a client can use to connect to the database independently of whether the primary or standby database is the currently active (primary) database.

In order to use this simplified configuration, Oracle Database 11g Release 2 introduces two new SQL*Net parameters that can be used on for connection strings of individual clients. The first parameter is **CONNECT_TIMEOUT**. It specifies the timeout duration (in seconds) for a client to establish an Oracle Net connection to an Oracle database. This parameter overrides `SQLNET.OUTBOUND_CONNECT_TIMEOUT` in the `SQLNET.ORA`. The second parameter is **RETRY_COUNT** and it specifies the number of times an `ADDRESS_LIST` is traversed before the connection attempt is terminated.

Using these two parameters, both, the SCAN on the primary site and the standby site, can be used in the client connection strings. Even, if the randomly selected address points to the site that is not currently active, the timeout will allow the connection request to failover before the client has waited unreasonably long (the default timeout depending on the operating system can be as long as 10 minutes).

```
sales.example.com =(DESCRIPTION= (CONNECT_TIMEOUT=10) (RETRY_COUNT=3)
  (ADDRESS_LIST= (LOAD_BALANCE=on) (FAILOVER=ON)
    (ADDRESS=(PROTOCOL=tcp) (HOST=sales1-scan) (PORT=1521))
    (ADDRESS=(PROTOCOL=tcp) (HOST=sales2-scan) (PORT=1521)))
  (CONNECT_DATA=(SERVICE_NAME= saleservice.example.com)))
```

Example 12: TNSNAMES.ORA entry for MAA environment

Using SCAN in a MAA Environment using GDS

Global Data Services (GDS) is a new capability available with Oracle Database 12c that extends intelligent load balancing and client failover concepts to globally distributed environments in which there are two or more failover targets that could be used to maintain availability. For MAA configurations using GDS, using SCAN listeners is avoided. For more information regarding the configuration of GDS and SCAN, please, consult the respective Oracle documentation.

Using SCAN with Oracle Connection Manager

If you use Oracle Connection Manager (CMAN) with your Oracle RAC Database, the `REMOTE_LISTENER` parameter for the Oracle RAC instances should include the CMAN server so that the CMAN server will receive load balancing related information and can therefore load balance connections across the available instances.

The easiest way to achieve this would be to add the CMAN-server as an entry to the `REMOTE_LISTENER` of the databases that clients want to connect to via CMAN as shown in figure 10. Note also that you will have to remove the SCAN from the TNSNAMES connect descriptor of the clients and further configurations will be required for the CMAN server. See the CMAN documentation for more details.

```
SQL> show parameters listener
```

NAME	TYPE	VALUE
listener_networks	string	
local_listener	string	(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=148.87.58.109)(PORT=1521))))
remote_listener	string	stscan3.oracle.com:1521,(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOST=CMANserver)(PORT=1521))))

Example 13: Server side TNSNAMES.ora example entry when using CMAN

Summary and Conclusion

Single Client Access Name (SCAN) is a feature used in Oracle Real Application Clusters environments that provides a single name for clients to access any Oracle Database running in a cluster. SCAN was first introduced with Oracle Real Application Clusters (RAC) 11g Release 2 as part of the Oracle Grid Infrastructure suite of products and was further enhanced with Oracle RAC 12c.

In any version, having a single name to access the cluster to connect to a database in this cluster allows clients to use EZConnect and the simple JDBC thin URL to access any database running in the cluster, independently of the number of databases or servers running in the cluster and regardless on which server(s) in the cluster the requested database is actually active.

Support for multiple subnets in the cluster as well as IPv6 support in addition to a more secure way of registering database instances with SCAN listeners floating in the cluster make the latest version of SCAN more flexible and effective in order to support DBaaS or private database cloud deployments.



Oracle Single Client Access Name (SCAN)
June 2013

Author: Markus Michalewicz
Contributing Authors: Barb Lundhild

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0113

Hardware and Software, Engineered to Work Together