SINGLE CLIENT ACCESS NAME (SCAN)

Single Client Access Name (SCAN) is s a new Oracle Real Application Clusters (RAC) 11g Release 2 feature that provides a single name for clients to access Oracle Databases running in a cluster. The **benefit** is that the client's connect information does not need to change if you add or remove nodes in the cluster. Having a single name to access the cluster allows clients to use the EZConnect client and the simple JDBC thin URL to access any database running in the cluster, independently of which server(s) in the cluster the database is active. SCAN provides load balancing and failover for client connections to the database. The SCAN works as a cluster alias for databases in the cluster.

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

Figure 1: Sample EZConnect and Thin JDBC Connect Strings

NETWORK REQUIREMENTS FOR USING SCAN

The SCAN is configured during the installation of Oracle Grid Infrastructure that is distributed with Oracle Database 11g Release2. 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. 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 the SCAN in your corporate DNS (Domain Name Service)
2. Use the Grid Naming Service (GNS)

USING OPTION 1 - DEFINE THE SCAN IN YOUR CORPORATE DNS

If you choose Option 1, you must ask your network administrator to create a single name that resolves to 3 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 public network in the cluster. The name must be 15 characters or less in length, not including the domain, and must be resolvable without the domain suffix (for example: "sales1-scan' must be resolvable as opposed to "scan1-scan.example.com"). The IPs must not be assigned to a network interface (on the cluster), 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
```

Figure 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 3 IPs in a different order.

Oracle Technical Paper

First nslookup	Second nslookup

`[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`
	`[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`

Figure 3: Look up the SCAN configuration in DNS using "nslookup"

Note: If your DNS server does not return a set of 3 IPs as shown in figure 3 or does not round-robin, ask your network administrator to enable such a setup. DNS using a round-robin algorithm on its own does not ensure failover of connections. However, the Oracle Client typically handles this. It is therefore **recommended** that the minimum version of the client used is the Oracle Database 11g Release 2 client.

USING OPTION 2 - THE GRID NAMING SERVICE (GNS)

If you choose option 2, you only need to enter the SCAN during the interview. During the cluster configuration, three IP addresses will be acquired from a DHCP service (using GNS assumes you have a DHCP service available on your public network) to create the SCAN and name resolution for the SCAN will be provided by the GNS[1].

IF YOU DO NOT HAVE A DNS SERVER AVAILABLE AT INSTALLATION TIME

Oracle Universal Installer (OUI) enforces providing a 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 databases in the cluster. All Oracle Database 11g Release 2 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) releases. However, it is strongly recommended to enable the SCAN configuration as described under "Option 1" or "Option 2" above 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.

---

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

SCAN CONFIGURATION IN THE CLUSTER

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 + 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 where a SCAN VIP is running fails, the SCAN VIP and its associated listener will failover 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.

```
[oracle@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

[oracle@mynode] srvctl config scan
SCAN name: sales1-scan, Network: 1/133.22.67.0/255.255.255.0/
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
```

Figure 4: Sample SCAN configuration

DATABASE CONFIGURATION USING SCAN

For Oracle Database 11g Release 2, SCAN is an essential part of the 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. With Oracle Grid Infrastructure 11g Release 2 the node listener run out of the Oracle Grid Infrastructure home and listen 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 (HOST=sales1-scan for example) in the address list entry, but use the simplified "SCAN:port" syntax as shown in figure 5.

```
NAME                      TYPE         VALUE
------------------------- -----------  ------------------------------
local_listener            string       (DESCRIPTION=(ADDRESS_LIST=(AD
                                        DRESS=(PROTOCOL=TCP)(HOST=133.
                                        22.67.111)(PORT=1521))))
remote_listener           string       sales1-scan.example.com:1521
```

Figure 5: LOCAL_LISTENER and REMOTE_LISTENER – Typical setting

Note: if you are using 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 (the Oracle 11g Release 2 default is NAMES.DIRECTORY_PATH=(tnsnames, ldap, ezconnect)).

HOW CONNECTION LOAD BALANCING WORKS 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.
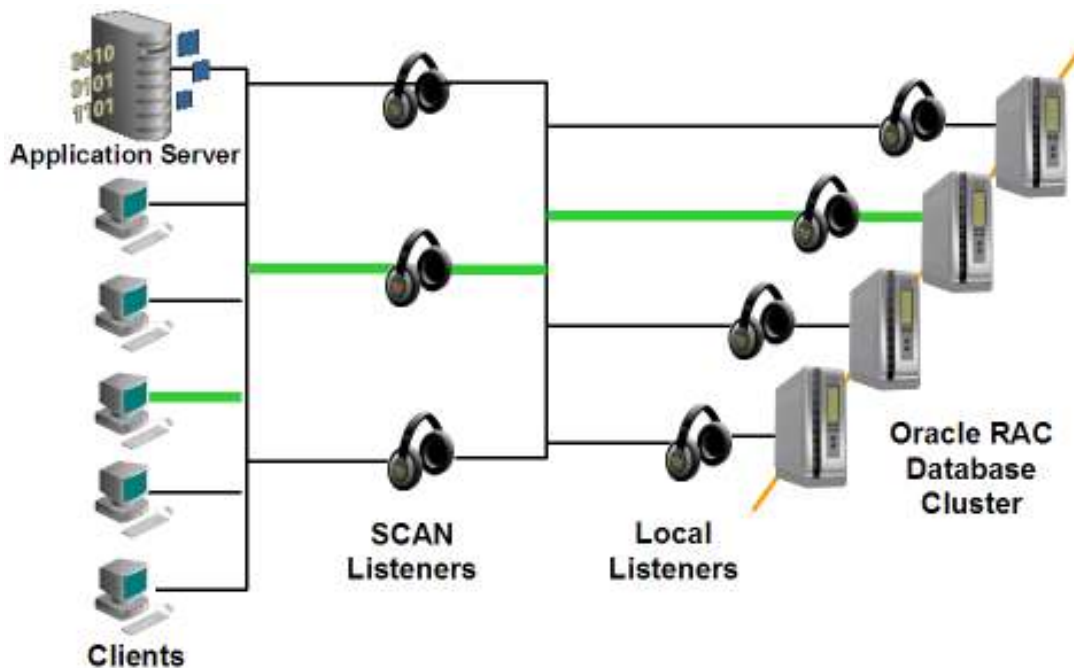

Figure 6: Connection Load Balancing using SCAN - Illustrated

Note: This example assumes an Oracle 11g Release 2 client using a default TNSNAMES. ora:

```
ORCLservice =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = sales1-scan.example.com)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = MyORCLservice)
    ))
```

Figure 7: Default Oracle 11g Release 2 TNSNAMES.ora using SCAN - Example

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. 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 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 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	Oracle Database 11g Release 2	No change required.
Oracle Database 11g Release 2	**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	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 Release 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)))
```

Figure 8: Sample TNSNAMES.ora for Oracle Database pre- 11g Release 2 Clients

USING SCAN IN A MAXIMUM AVAILABILITY ARCHITECTURE ENVIRONMENT

If you have implemented a Maximum Availability Architecture (MAA) environment, 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.OUTBOUT_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= salesservice.example.com)))
```

Figure 9 Sample TNSNAMES.ORA entry for MAA environment

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))))
```

Figure 10: Server side TNSNAMES.ora example entry when using CMAN

Authors: Barb Lundhild, Markus Michalewicz

Last Updated: August 10th, 2010