Oracle® Database Database Net Services Reference





Oracle Database Database Net Services Reference, 19c

E96314-19

Copyright © 2002, 2025, Oracle and/or its affiliates.

Primary Author: Spurthi SV

Contributing Authors: Douglas Williams, Prakash Jashnani, Binika Kumar

Contributors: Abhishek Dadhich, Alan Williams, Anita Patel, Bhaskar Mathur, Ching Tai, Christopher Jones, David Lin, Feroz Khan, Hector Pujol, Jean Zeng, Kant Patel, Kevin Neel, Krishna Itikarlapalli, Mark Dilman, Misaki Miyashita, Murali Purayathu, Norman Woo, Peter Knaggs, Robert Achacoso, Santanu Datta, Saravanakumar Ramasubramanian, Sarma Namuduri, Scot McKinley, Sharad Chandran R, Sherry LaMonica, Srinivas Pamu, Steve Ding, Sudeep Reguna, Sweta Mogra, Thanigai Nallathambi, Yi Ouyang

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software, software documentation, data (as defined in the Federal Acquisition Regulation), or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software," "commercial computer software documentation," or "limited rights data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed, or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

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

Intel and Intel Inside 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, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

	rΔi	Fa	^^
\mathbf{P}		เล	ce

Audience		XİV
Documenta	ation Accessibility	xiv
Diversity ar	nd Inclusion	xiv
Related Do	cuments	xiv
Convention	S .	ΧV
Change	s in This Release for Oracle Database Net Service	s Reference
New Featur	res	XV
Deprecated	d Features	xviii
Listener	Control Utility	
1.1 Lister	ner Control Utility Overview	1-1
1.2 SET	and SHOW Commands of the Listener Control utility	1-2
	buted Operations	1-3
	le Net Listener Security	1-3
1.5 Lister	ner Control Utility Commands	1-3
1.5.1	EXIT	1-5
1.5.2	HELP	1-6
1.5.3	QUIT	1-7
1.5.4	RELOAD	1-7
1.5.5	SAVE_CONFIG	1-8
1.5.6	SERVICES	1-9
1.5.7	SET	1-10
1.5.8	SET CURRENT_LISTENER	1-11
1.5.9	SET DISPLAYMODE	1-12
1.5.10	SET INBOUND_CONNECT_TIMEOUT	1-13
	SET LOG_DIRECTORY	1-14
1.5.12	SET LOG_FILE	1-14
1.5.13	——————————————————————————————————————	1-15
1.5.14		1-16
1.5.15	SET TRC_DIRECTORY	1-17



	1.5.16	SET TRC_FILE	1-18
	1.5.17	SET TRC_LEVEL	1-18
	1.5.18	SHOW	1-19
	1.5.19	SPAWN	1-20
	1.5.20	START	1-21
	1.5.21	STATUS	1-23
	1.5.22	STOP	1-25
	1.5.23	TRACE	1-25
	1.5.24	VERSION	1-26
2	Oracle (Connection Manager Control Utility	
	2.1 Com	mand Modes and Syntax	2-1
	2.2 Orac	cle Connection Manager Control Utility Overview	2-3
	2.3 Orac	cle Connection Manager Control Utility Commands	2-3
	2.3.1	ADMINISTER	2-4
	2.3.2	CLOSE CONNECTIONS	2-5
	2.3.3	EXIT	2-7
	2.3.4	HELP	2-7
	2.3.5	QUIT	2-8
	2.3.6	RELOAD	2-8
	2.3.7	RESUME GATEWAYS	2-9
	2.3.8	SAVE_PASSWD	2-10
	2.3.9	SET	2-10
	2.3.10		2-11
	2.3.11	_	2-11
	2.3.12		2-12
	2.3.13	_	2-13
	2.3.14		2-13
	2.3.15	_	2-14
	2.3.16	_	2-15
	2.3.17		2-15
	2.3.18		2-16
	2.3.19 2.3.20	_	2-17 2-17
	2.3.21	-	2-17
	2.3.21	_	2-19
	2.3.22		2-19
	2.3.23		2-19
	2.3.24		2-23
	2.3.26		2-23
	2.3.27		2-24
	2.0.21	5.15.1 0/11EW/110	2 27



	2.3.28 SHOW PARAMETERS	2-25
	2.3.29 SHOW RULES	2-26
	2.3.30 SHOW SERVICES	2-27
	2.3.31 SHOW STATUS	2-27
	2.3.32 SHOW VERSION	2-28
	2.3.33 SHUTDOWN	2-29
	2.3.34 STARTUP	2-29
	2.3.35 SUSPEND GATEWAY	2-30
3	Syntax Rules for Configuration Files	
	3.1 Overview of Configuration File Syntax	3-1
	3.2 Syntax Rules for Configuration Files	3-2
	3.3 Network Character Set for Keywords	3-2
	3.4 Character Set for Listener and Net Service Names	3-3
4	Protocol Address Configuration	
	4.1 Protocol Addresses	4-1
	4.1.1 ADDRESS	4-1
	4.1.2 ADDRESS_LIST	4-2
	4.2 Protocol Parameters	4-2
	4.3 Recommended Port Numbers	4-4
	4.4 Port Number Limitations	4-4
5	Parameters for the sqlnet.ora File	
	5.1 Overview of Profile Configuration File	5-1
	5.2 sqlnet.ora Profile Parameters	5-2
	5.2.1 ACCEPT_MD5_CERTS	5-7
	5.2.2 ACCEPT_SHA1_CERTS	5-7
	5.2.3 ADD_SSLV3_TO_DEFAULT	5-8
	5.2.4 BEQUEATH_DETACH	5-8
	5.2.5 DEFAULT_SDU_SIZE	5-9
	5.2.6 DISABLE_INTERRUPT	5-9
	5.2.7 DISABLE_OOB	5-10
	5.2.8 DISABLE_OOB_AUTO	5-10
	5.2.9 EXADIRECT_FLOW_CONTROL	5-10
	5.2.10 EXADIRECT_RECVPOLL	5-11
	5.2.11 IPC.KEYPATH	5-11
	5.2.12 NAMES.DEFAULT_DOMAIN	5-11
	5.2.13 NAMES.DIRECTORY_PATH	5-12



5.2.14	NAMES.LDAP_AUTHENTICATE_BIND	5-13
5.2.15	NAMES.LDAP_CONN_TIMEOUT	5-13
5.2.16	NAMES.LDAP_PERSISTENT_SESSION	5-13
5.2.17	NAMES.NIS.META_MAP	5-14
5.2.18	OCI_COMPARTMENT	5-14
5.2.19	OCI_DATABASE	5-16
5.2.20	OCI_IAM_URL	5-17
5.2.21	OCI_TENANCY	5-19
5.2.22	PASSWORD_AUTH	5-20
5.2.23	RECV_BUF_SIZE	5-24
5.2.24	SDP.PF_INET_SDP	5-24
5.2.25	SEC_USER_AUDIT_ACTION_BANNER	5-25
5.2.26	SEC_USER_UNAUTHORIZED_ACCESS_BANNER	5-25
5.2.27	SEND_BUF_SIZE	5-25
5.2.28	SQLNET.ALLOW_WEAK_CRYPTO	5-26
5.2.29	SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS	5-27
5.2.30	SQLNET.ALLOWED_LOGON_VERSION_CLIENT	5-29
5.2.31	SQLNET.ALLOWED_LOGON_VERSION_SERVER	5-30
5.2.32	SQLNET.AUTHENTICATION_SERVICES	5-35
5.2.33	SQLNET.CLIENT_REGISTRATION	5-37
5.2.34	SQLNET.CLOUD_USER	5-37
5.2.35	SQLNET.COMPRESSION	5-39
5.2.36	SQLNET.COMPRESSION_ACCELERATION	5-39
5.2.37	SQLNET.COMPRESSION_LEVELS	5-40
5.2.38	SQLNET.COMPRESSION_THRESHOLD	5-40
5.2.39	SQLNET.CRYPTO_CHECKSUM_CLIENT	5-40
5.2.40	SQLNET.CRYPTO_CHECKSUM_SERVER	5-41
5.2.41	SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT	5-42
5.2.42	SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER	5-42
5.2.43	SQLNET.DBFW_PUBLIC_KEY	5-43
5.2.44	SQLNET.DOWN_HOSTS_TIMEOUT	5-44
5.2.45	SQLNET.ENCRYPTION_CLIENT	5-44
5.2.46	SQLNET.ENCRYPTION_SERVER	5-45
5.2.47	SQLNET.ENCRYPTION_TYPES_CLIENT	5-45
5.2.48	SQLNET.ENCRYPTION_TYPES_SERVER	5-46
5.2.49	SQLNET.EXPIRE_TIME	5-47
5.2.50	SQLNET.IGNORE_ANO_ENCRYPTION_FOR_TCPS	5-48
5.2.51	SQLNET.INBOUND_CONNECT_TIMEOUT	5-48
5.2.52	SQLNET.FALLBACK_AUTHENTICATION	5-49
5.2.53	SQLNET.KERBEROS5_CC_NAME	5-50
5.2.54	SQLNET.KERBEROS5_CLOCKSKEW	5-51
5.2.55	SQLNET.KERBEROS5_CONF	5-51



SQLNET.KERBEROS5_CONF_LOCATION	5-52
SQLNET.KERBEROS5_KEYTAB	5-52
SQLNET.KERBEROS5_REALMS	5-53
SQLNET.KERBEROS5_REPLAY_CACHE	5-53
SQLNET.OUTBOUND_CONNECT_TIMEOUT	5-53
SQLNET.RADIUS_ALTERNATE	5-54
SQLNET.RADIUS_ALTERNATE_PORT	5-55
SQLNET.RADIUS_ALTERNATE_RETRIES	5-55
SQLNET.RADIUS_ALTERNATE_TIMEOUT	5-56
SQLNET.RADIUS_AUTHENTICATION	5-56
SQLNET.RADIUS_AUTHENTICATION_INTERFACE	5-56
SQLNET.RADIUS_AUTHENTICATION_PORT	5-57
SQLNET.RADIUS_AUTHENTICATION_RETRIES	5-57
SQLNET.RADIUS_AUTHENTICATION_TIMEOUT	5-58
SQLNET.RADIUS_CHALLENGE_KEYWORD	5-58
SQLNET.RADIUS_CHALLENGE_RESPONSE	5-59
SQLNET.RADIUS_CLASSPATH	5-59
SQLNET.RADIUS_SECRET	5-59
SQLNET.RADIUS_SEND_ACCOUNTING	5-60
SQLNET.RECV_TIMEOUT	5-60
SQLNET.SEND_TIMEOUT	5-61
SQLNET.URI	5-62
SQLNET.USE_HTTPS_PROXY	5-63
SQLNET.WALLET_OVERRIDE	5-63
SSL_CERT_REVOCATION	5-64
SSL_CRL_FILE	5-65
SSL_CRL_PATH	5-66
SSL_CIPHER_SUITES	5-66
SSL_CLIENT_AUTHENTICATION	5-69
SSL_DISABLE_WEAK_EC_CURVES	5-70
SSL_EXTENDED_KEY_USAGE	5-70
SSL_SERVER_DN_MATCH	5-71
SSL_VERSION	5-72
TCP.CONNECT_TIMEOUT	5-73
TCP.EXCLUDED_NODES	5-73
TCP.INVITED_NODES	5-74
TCP.NODELAY	5-74
TCP.QUEUESIZE	5-74
TCP.VALIDNODE_CHECKING	5-75
TNSPING.TRACE_DIRECTORY	5-76
TNSPING.TRACE_LEVEL	5-76
TOKEN_AUTH	5-76
	SQLNET.KERBEROS5_KEYTAB SQLNET.KERBEROS5_REALMS SQLNET.KERBEROS5_REPLAY_CACHE SQLNET.OUTBOUND_CONNECT_TIMEOUT SQLNET.RADIUS_ALTERNATE SQLNET.RADIUS_ALTERNATE_PORT SQLNET.RADIUS_ALTERNATE_TIMEOUT SQLNET.RADIUS_ALTERNATE_TIMEOUT SQLNET.RADIUS_AUTHENTICATION SQLNET.RADIUS_AUTHENTICATION_INTERFACE SQLNET.RADIUS_AUTHENTICATION_PORT SQLNET.RADIUS_AUTHENTICATION_PORT SQLNET.RADIUS_AUTHENTICATION_TIMEOUT SQLNET.RADIUS_AUTHENTICATION_TIMEOUT SQLNET.RADIUS_CHALLENGE_KEYWORD SQLNET.RADIUS_CHALLENGE_RESPONSE SQLNET.RADIUS_SECRET SQLNET.RADIUS_SECRET SQLNET.RADIUS_SECRET SQLNET.RADIUS_SECRET SQLNET.RADIUS_SEND_ACCOUNTING SQLNET.RECY_TIMEOUT SQLNET.USE_HTTPS_PROXY SQLNET.USE_HTTPS_PROXY SQLNET.WALLET_OVERRIDE SSL_CERT_REVOCATION SSL_CRL_FILE SSL_CRL_PATH SSL_CIPHER_SUITES SSL_CLIENT_AUTHENTICATION SSL_DISABLE_WEAK_EC_CURVES SSL_EXTENDED_KEY_USAGE SSL_EXTENDED_KEY_USAGE SSL_SERVER_DN_MATCH SSL_SERVER_DN_MATCH SSL_SERVERSION TCP.CONNECT_TIMEOUT TCP.EXCLUDED_NODES TCP.NODELAY TCP.QUEUESIZE TCP.VALIDNODE_CHECKING TNSPING.TRACE_LEVEL



	5.2.98 TOKEN_LOCATION	5-82
	5.2.99 USE_CMAN	5-86
	5.2.100 USE_DEDICATED_SERVER	5-87
	5.2.101 WALLET_LOCATION	5-88
	5.3 ADR Diagnostic Parameters in sqlnet.ora	5-90
	5.3.1 About ADR Diagnostic Parameters	5-90
	5.3.2 ADR_BASE	5-91
	5.3.3 DIAG_ADR_ENABLED	5-91
	5.3.4 TRACE_LEVEL_CLIENT	5-92
	5.3.5 TRACE_LEVEL_SERVER	5-92
	5.3.6 TRACE_TIMESTAMP_CLIENT	5-93
	5.3.7 TRACE_TIMESTAMP_SERVER	5-93
	5.4 Non-ADR Diagnostic Parameters in sqlnet.ora Files	5-94
	5.4.1 LOG_DIRECTORY_CLIENT	5-94
	5.4.2 LOG_DIRECTORY_SERVER	5-95
	5.4.3 LOG_FILE_CLIENT	5-95
	5.4.4 LOG_FILE_SERVER	5-96
	5.4.5 TRACE_DIRECTORY_CLIENT	5-96
	5.4.6 TRACE_DIRECTORY_SERVER	5-96
	5.4.7 TRACE_FILE_CLIENT	5-97
	5.4.8 TRACE_FILE_SERVER	5-97
	5.4.9 TRACE_FILEAGE_CLIENT	5-97
	5.4.10 TRACE_FILEAGE_SERVER	5-98
	5.4.11 TRACE_FILELEN_CLIENT	5-98
	5.4.12 TRACE_FILELEN_SERVER	5-99
	5.4.13 TRACE_FILENO_CLIENT	5-99
	5.4.14 TRACE_FILENO_SERVER	5-99
	5.4.15 TRACE_UNIQUE_CLIENT	5-100
6	Local Naming Parameters in the tnsnames.ora File	
	6.1 Overview of Local Naming Parameters	6-2
	6.2 General Syntax of tnsnames.ora	6-3
	6.3 Using Multiple Descriptions in tnsnames.ora Files	6-3
	6.4 Multiple Address Lists in tnsnames.ora Files	6-3
	6.5 Connect-Time Failover and Client Load Balancing with Oracle Connection Managers	6-4
	6.6 Connect Descriptor Descriptions	6-6
	6.6.1 DESCRIPTION_LIST	6-6
	6.6.2 DESCRIPTION	6-6
	6.7 Protocol Addresses	6-7
	6.7.1 ADDRESS	6-7
	6.7.2 HTTPS_PROXY	6-7



5-82

	6.7.3	HTTPS_PROXY_PORT	6-8
	6.7.4	ADDRESS_LIST	6-9
6.8	Optio	onal Parameters for Description	6-9
	6.8.1	ENABLE	6-10
	6.8.2	FAILOVER	6-10
	6.8.3	LOAD_BALANCE	6-11
	6.8.4	RECV_BUF_SIZE	6-12
	6.8.5	SDU	6-13
	6.8.6	SEND_BUF_SIZE	6-13
	6.8.7	SOURCE_ROUTE	6-14
	6.8.8	TYPE_OF_SERVICE	6-15
6.9	Conr	nection Data Section	6-16
	6.9.1	COLOCATION_TAG	6-16
	6.9.2	CONNECT_DATA	6-17
	6.9.3	FAILOVER_MODE	6-17
	6.9.4	GLOBAL_NAME	6-18
	6.9.5	HS	6-19
	6.9.6	INSTANCE_NAME	6-19
	6.9.7	KERBEROS5_PRINCIPAL	6-20
	6.9.8	RDB_DATABASE	6-21
	6.9.9	SHARDING_KEY	6-22
	6.9.10	SUPER_SHARDING_KEY	6-24
	6.9.11	SERVER	6-25
	6.9.12	SERVICE_NAME	6-26
6.1	0 Sec	curity Section	6-26
	6.10.1	AUTHENTICATION_SERVICE	6-28
	6.10.2	IGNORE_ANO_ENCRYPTION_FOR_TCPS	6-30
	6.10.3	KERBEROS5_CC_NAME	6-30
	6.10.4	OCI_COMPARTMENT	6-31
	6.10.5	OCI_DATABASE	6-33
	6.10.6	OCI_IAM_URL	6-34
	6.10.7	OCI_TENANCY	6-36
	6.10.8	PASSWORD_AUTH	6-37
	6.10.9	SECURITY	6-41
	6.10.10	SSL_DISABLE_WEAK_EC_CURVES	6-41
	6.10.11	SSL_SERVER_CERT_DN	6-42
	6.10.12	SSL_SERVER_DN_MATCH	6-43
	6.10.13	S SSL_VERSION	6-44
	6.10.14	TOKEN_AUTH	6-45
	6.10.15	TOKEN_LOCATION	6-50
	6.10.16	WALLET_LOCATION	6-54
6.1	1 Time	eout Parameters	6-55



6.11.1	CONNECT_TIMEOUT	6-56
6.11.2	RETRY_COUNT	6-57
6.11.3	RETRY_DELAY	6-58
6.11.4	TRANSPORT_CONNECT_TIMEOUT	6-58
6.11.5	RECV_TIMEOUT	6-59
6.12 Cor	mpression Parameters	6-60
6.12.1	COMPRESSION	6-60
6.12.2	COMPRESSION_LEVELS	6-61
Oracle I	Net Listener Parameters in the listener.ora File	
	view of Oracle Net Listener Configuration File	7-1
7.2 Proto	ocol Address Parameters	7-2
7.2.1	ADDRESS	7-3
7.2.2		7-3
	Firewall	7-4
	IP	7-4
7.2.5	•	7-4
	RECV_BUF_SIZE	7-5
7.2.7	SEND_BUF_SIZE	7-6
	nection Rate Limiter Parameters	7-7
7.3.1	CONNECTION_RATE_listener_name	7-7
7.3.2	RATE_LIMIT	7-7
	rol Parameters	7-9
7.4.1	ADMIN_RESTRICTIONS_listener_name	7-10
7.4.2	ALLOW_MULTIPLE_REDIRECTS_listener_name	7-11
7.4.3	ENABLE_EXADIRECT_listener_name	7-11
7.4.4	CRS_NOTIFICATION_listener_name	7-12
7.4.5	DEDICATED_THROUGH_BROKER_LISTENER	7-12
7.4.6	DEFAULT_SERVICE_listener_name	7-12
7.4.7	INBOUND_CONNECT_TIMEOUT_listener_name	7-13
7.4.8	LOCAL_REGISTRATION_ADDRESS_listener_name	7-14
7.4.9	MAX_ALL_CONNECTIONS_listener_name	7-14
7.4.10		7-15
7.4.11	REGISTRATION_EXCLUDED_NODES_listener_name	7-15
7.4.12	REGISTRATION_INVITED_NODES_listener_name	7-15
7.4.13	REMOTE_REGISTRATION_ADDRESS_listener_name	7-16
7.4.14	SAVE_CONFIG_ON_STOP_listener_name	7-17
7.4.15	SERVICE_RATE_listener_name	7-17
7.4.16	SSL_CIPHER_SUITES	7-18
7.4.17	SSL_CLIENT_AUTHENTICATION	7-20
7.4.18	SSL_DISABLE_WEAK_EC_CURVES	7-21



	7.4.19	SSL_VERSION	7-22
	7.4.20	SUBSCRIBE_FOR_NODE_DOWN_EVENT_listener_name	7-23
	7.4.21	USE_SID_AS_SERVICE_listener_name	7-23
	7.4.22	VALID_NODE_CHECKING_REGISTRATION_listener_name	7-24
	7.4.23	WALLET_LOCATION	7-25
	7.5 ADR	Diagnostic Parameters for Oracle Net Listener	7-26
	7.5.1	ADR_BASE_listener_name	7-27
	7.5.2	DIAG_ADR_ENABLED_listener_name	7-27
	7.5.3	LOG_FILE_NUM_listener_name	7-28
	7.5.4	LOG_FILE_SIZE_listener_name	7-28
	7.5.5	LOGGING_listener_name	7-29
	7.5.6	TRACE_LEVEL_listener_name	7-29
	7.5.7	TRACE_TIMESTAMP_listener_name	7-30
	7.6 Non-	ADR Diagnostic Parameters for Oracle Net Listener	7-30
	7.6.1	LOG_DIRECTORY_listener_name	7-31
	7.6.2	LOG_FILE_listener_name	7-31
	7.6.3	TRACE_DIRECTORY_listener_name	7-31
	7.6.4	TRACE_FILE_listener_name	7-32
	7.6.5	TRACE_FILEAGE_listener_name	7-32
	7.6.6	TRACE_FILELEN_listener_name	7-32
	7.6.7	TRACE_FILENO_listener_name	7-33
	7.7 Class	s of Secure Transports Parameters	7-33
	7.7.1	SECURE_REGISTER_listener_name	7-34
	7.7.2	Using COST Parameters in Combination	7-34
	7.7.3	DYNAMIC_REGISTRATION_listener_name	7-35
	7.7.4	SECURE_PROTOCOL_listener_name	7-35
	7.7.5	SECURE_CONTROL_listener_name	7-36
8	Oracle (Connection Manager Parameters	
	8.1 Over	view of Oracle Connection Manager Configuration File	8-1
		ele Connection Manager Parameters	8-3
	8.2.1	ADDRESS	8-5
	8.2.2	ADMIN_GROUP	8-6
	8.2.3	ASO_AUTHENTICATION_FILTER	8-6
	8.2.4	COMPRESSION	8-7
	8.2.5	COMPRESSION_LEVELS	8-7
	8.2.6	COMPRESSION_THRESHOLD	8-8
	8.2.7	CONNECTION_STATISTICS	8-8
	8.2.8	EVENT_GROUP	8-8
	8.2.9	EXPIRE_TIME	8-9
	8.2.10	IDLE_TIMEOUT	8-10



	8.2.11	INBOUND_CONNECT_TIMEOUT	8-10
	8.2.12	LOG_FILE_NUM	8-11
	8.2.13	LOG_FILE_SIZE	8-11
	8.2.14	MAX_ALL_CONNECTIONS	8-11
	8.2.15	MAX_CMCTL_SESSIONS	8-12
	8.2.16	MAX_CONNECTIONS	8-12
	8.2.17	MAX_GATEWAY_PROCESSES	8-13
	8.2.18	MAX_REG_CONNECTIONS	8-13
	8.2.19	MIN_GATEWAY_PROCESSES	8-13
	8.2.20	OUTBOUND_CONNECT_TIMEOUT	8-13
	8.2.21	PASSWORD_instance_name	8-14
	8.2.22	REGISTRATION_EXCLUDED_NODES	8-14
	8.2.23	REGISTRATION_INVITED_NODES	8-14
	8.2.24	RULE	8-15
	8.2.25	SDU	8-16
	8.2.26	SERVICE_RATE	8-17
	8.2.27	SESSION_TIMEOUT	8-17
	8.2.28	SSL_CIPHER_SUITES	8-18
	8.2.29	SSL_CLIENT_AUTHENTICATION	8-20
	8.2.30	SSL_VERSION	8-21
	8.2.31	TRACE_FILE	8-22
	8.2.32	USE_SID_AS_SERVICE	8-22
	8.2.33	VALID_NODE_CHECKING_REGISTRATION	8-23
	8.2.34	WALLET_LOCATION	8-23
8.3	Orac	e Connection Manager in Traffic Director Mode Parameters	8-25
	8.3.1	SERVICE_AFFINITY	8-26
	8.3.2	TDM	8-26
	8.3.3	TDM_BIND_THREAD	8-27
	8.3.4	TDM_DATATYPE_CHECK	8-27
	8.3.5	TDM_PRCP_MAX_CALL_WAIT_TIME	8-28
	8.3.6	TDM_PRCP_MAX_TXN_CALL_WAIT_TIME	8-28
	8.3.7	TDM_SHARED_THREADS_MAX	8-29
	8.3.8	TDM_SHARED_THREADS_MIN	8-29
	8.3.9	TDM_THREADING_MODE	8-29
8.4		Diagnostic Parameters for Oracle Connection Manager	8-30
	8.4.1	ADR_BASE	8-30
	8.4.2	DIAG_ADR_ENABLED	8-31
	8.4.3	LOG_LEVEL	8-31
	8.4.4	TRACE_LEVEL	8-32
	8.4.5	TRACE_TIMESTAMP	8-32
8.5		ADR Diagnostic Parameters for Oracle Connection Manager	8-33
	8.5.1	LOG_DIRECTORY	8-33



TRACE_FILELEN TRACE_FILENO Ory Usage Parameters in the Idap.ora File verview of Directory Server Usage File rectory Usage Parameters DEFAULT_ADMIN_CONTEXT DIRECTORY_SERVER_TYPE DIRECTORY_SERVERS Oendices Verview of Unsupported in this Release verview of Unsupported Features Oracle Net Connection Pooling Oracle Names Oracle Net Listener Password Insupported Parameters	8-34 8-34 8-34 9-1 9-1 9-2 9-2 9-2 4-1 A-1 A-2 A-2
ory Usage Parameters in the Idap.ora File verview of Directory Server Usage File rectory Usage Parameters DEFAULT_ADMIN_CONTEXT DIRECTORY_SERVER_TYPE DIRECTORY_SERVERS Dendices res Not Supported in this Release verview of Unsupported Features Oracle Net Connection Pooling Oracle Names Oracle Net Listener Password	9-1 9-2 9-2 9-2 9-2 A-1 A-1 A-2 A-2
verview of Directory Server Usage File rectory Usage Parameters DEFAULT_ADMIN_CONTEXT DIRECTORY_SERVER_TYPE DIRECTORY_SERVERS Dendices Tes Not Supported in this Release Verview of Unsupported Features Oracle Net Connection Pooling Oracle Names Oracle Net Listener Password	9-1 9-2 9-2 9-2 A-1 A-1 A-2 A-2
rectory Usage Parameters DEFAULT_ADMIN_CONTEXT DIRECTORY_SERVER_TYPE DIRECTORY_SERVERS Dendices res Not Supported in this Release verview of Unsupported Features Oracle Net Connection Pooling Oracle Names Oracle Net Listener Password	9-1 9-2 9-2 9-2 A-1 A-1 A-2 A-2
DEFAULT_ADMIN_CONTEXT DIRECTORY_SERVER_TYPE DIRECTORY_SERVERS Dendices Tes Not Supported in this Release Verview of Unsupported Features Dendices Oracle Net Connection Pooling Oracle Names Oracle Net Listener Password	9-2 9-2 9-2 9-2 A-1 A-1 A-2 A-2
DIRECTORY_SERVER_TYPE DIRECTORY_SERVERS Dendices Tes Not Supported in this Release Verview of Unsupported Features Oracle Net Connection Pooling Oracle Names Oracle Net Listener Password	9-2 9-2 A-1 A-1 A-2 A-2
DIRECTORY_SERVERS Dendices Tes Not Supported in this Release Verview of Unsupported Features 1. Oracle Net Connection Pooling 2. Oracle Names 3. Oracle Net Listener Password	9-2 A-1 A-1 A-2 A-2
res Not Supported in this Release verview of Unsupported Features 1 Oracle Net Connection Pooling 2 Oracle Names 3 Oracle Net Listener Password	A-1 A-1 A-2 A-2
res Not Supported in this Release verview of Unsupported Features 1 Oracle Net Connection Pooling 2 Oracle Names 3 Oracle Net Listener Password	A-1 A-2 A-2
verview of Unsupported Features Oracle Net Connection Pooling Oracle Names Oracle Net Listener Password	A-1 A-2 A-2
Oracle Net Connection Pooling Oracle Names Oracle Net Listener Password	A-1 A-2 A-2
Oracle Names Oracle Net Listener Password	A-2 A-2
3 Oracle Net Listener Password	A-2
nsupported Parameters	
	A-2
nsupported Control Utility Commands	A-2
nsupported or Deprecated Protocols	A-3
de Considerations for Oracle Net Services	
nonymous Access to Oracle Internet Directory	B-1
Schema for Oracle Net Services	
ructural Object Classes	C-1
tributes	C-2
ary	
t	ade Considerations for Oracle Net Services nonymous Access to Oracle Internet Directory Schema for Oracle Net Services tructural Object Classes ttributes ary



Preface

Review this publication to obtain a complete listing and description of control utility commands and configuration file parameters that you can use to manage Oracle Net Services components.

- Audience
- Documentation Accessibility
- Diversity and Inclusion
- Related Documents
- Conventions

Audience

Oracle Database Net Services Reference is intended for network administrators who are responsible for configuring and administering network components.

To use this document, you should be familiar with the networking concepts and configuration tasks described in *Oracle Database Net Services Administrator's Guide*.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Related Documents

For additional information, see the following Oracle resources:



- Oracle Database Net Services Administrator's Guide
- Online Help for Oracle Net Services tools and utilities
- Oracle Database 19c documentation set
- Oracle Database Global Data Services Concepts and Administration Guide

A glossary of Oracle Net Services terms is available in *Oracle Database Net Services Administrator's Guide*.

Many books in the documentation set use the sample schemas of the seed database, which is installed by default when you install Oracle. Refer to *Oracle Database Sample Schemas* for additional information about how these schemas were created and how you can use them yourself.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.



Changes in This Release for Oracle Database Net Services Reference

Review the changes in *Oracle Database Net Services Reference* for Oracle Database 19c.

- New Features
 These are the new features and enhancements available with Oracle Database 19c.
- Deprecated Features
 These features are deprecated in this release and may be desupported in a future release.

New Features

These are the new features and enhancements available with Oracle Database 19c.

Identity and Access Management Integration with Additional Oracle Database Environments

Available for Oracle Database release 19.16, Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM) users can log in to additional Oracle Database Environments.

For a list of the supported Oracle Database environments, see *Oracle Database Security Guide*.

Ability to Use the IAM User Name and IAM Database Password to Retrieve a Database Token

Retrieving an IAM database token using the IAM user name and IAM database password or secure external password store (SEPS) is more secure than using the password verifier method of database access. You can configure the database client to request this token directly from an OCI IAM endpoint.

The new sqlnet.ora or this names.ora parameters enable you to configure this authentication method and specify the IAM endpoint along with additional metadata. These parameters are PASSWORD_AUTH, OCI_IAM_URL, OCI_TENANCY along with optional OCI_COMPARTMENT and OCI_DATABASE.

See Oracle Database Security Guide, PASSWORD_AUTH, OCI_IAM_URL, OCI_TENANCY, OCI_COMPARTMENT, and OCI_DATABASE.

Microsoft Azure Active Directory Integration with Additional Oracle Database Environments

Available for Oracle Database release 19.16, Microsoft Azure Active Directory (Azure AD) users can log in to additional Oracle Database environments with their Azure AD OAuth2 access token.

For a list of the supported Oracle Database environments, see *Oracle Database Security Guide*.



Azure AD Integration with Oracle Autonomous Cloud Databases

Available for Oracle Autonomous Database in June 2022, Azure AD users can log in to Oracle Cloud Infrastructure (OCI) Autonomous Database (Shared Infrastructure) with their Azure AD OAuth2 access token.

OCI Oracle Autonomous Database now can accept Azure AD OAuth2 tokens to access the database. Azure AD users can access the database directly using their Azure AD tokens, and applications can use their service tokens to access the database.

See Oracle Database Net Services Administrator's Guide and Oracle Database Security Guide.

The TOKEN_AUTH parameter allows you to configure Azure AD token-based authentication for Oracle Autonomous Cloud Databases. You must also use the TOKEN_LOCATION parameter to specify the directory location where the Azure AD token is stored for authentication.

See TOKEN_AUTH and TOKEN_LOCATION.

IAM Integration with Oracle Autonomous Cloud Databases

Available for Oracle Database release 19.13, IAM users can log in to Oracle Autonomous Database using either database password or token-based authentication.

An IAM ADMIN user can configure both the authentication and authorization of IAM users and IAM groups. An IAM user can log in to Oracle Autonomous Cloud Databases using tools, such as SQL*Plus or SQLcl.

See Oracle Database Net Services Administrator's Guide and Oracle Database Security Guide.

TOKEN AUTH and TOKEN LOCATION Parameters

The sqlnet.ora or thshames.ora parameter TOKEN_AUTH allows you to configure IAM token-based authentication for Oracle Autonomous Cloud Databases.

The sqlnet.ora or thinnames.ora parameter TOKEN_LOCATION allows you to override the default directory where the database token and private key files are stored for authentication. This is an optional parameter.

See TOKEN_AUTH and TOKEN_LOCATION.

One-Way Transport Layer Security (TLS)

This feature allows you to configure one-way TLS (server authentication). With this method, only the database server authenticates to the client by presenting its certificate issued by Certificate Authority (CA) and the client verifies whether the database server certificate is valid.

An Oracle client wallet with the server certificate is not required if the database server certificate is signed by a trusted common root certificate that is already installed in the local system default certificate store.

See Oracle Database Net Services Administrator's Guide.

Security Update for Native Encryption

The following supported algorithms are improved:

- Encryption algorithms: AES128, AES192, and AES256
- Crypto-checksum algorithms: SHA1, SHA256, SHA384, and SHA512

The following algorithms are deprecated:



- Encryption algorithms: DES, DES40, 3DES112, 3DES168, RC4_40, RC4_56, RC4_128, and RC4_256
- Crypto-checksum algorithm: MD5

To transition your Oracle Database environment to use stronger algorithms, download and install the patch described in My Oracle Support note 2118136.2.

The new sqlnet.ora parameters SQLNET.ALLOW_WEAK_CRYPTO and SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS enable you to review the specified encryption and crypto-checksum algorithms. This ensures that the connection does not encounter compatibility issues and your configuration uses supported strong algorithms.

See Oracle Database Security Guide.

SQLNET.ALLOW_WEAK_CRYPTO Parameter

Use the SQLNET.ALLOW_WEAK_CRYPTO parameter to configure your client-side network connection by reviewing the specified encryption and crypto-checksum algorithms.

See SQLNET.ALLOW_WEAK_CRYPTO.

SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS Parameter

Use the SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS parameter to configure your server-side network connection by reviewing the specified encryption and crypto-checksum algorithms.

See SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS.

COLOCATION TAG Parameter

The <code>COLOCATION_TAG</code> parameter is an alphanumeric string that you can use with the <code>CONNECT_DATA</code> parameter of the TNS connect string. When you set the <code>COLOCATION_TAG</code> parameter, it attempts to route clients with the same <code>COLOCATION_TAG</code> to the same database instance.

Colocation of sessions on the same instance can help decrease inter-instance communication and thereby increase performance for workload that benefits from being executed in the same instance.

See COLOCATION_TAG.

KERBEROS5 PRINCIPAL Parameter

When you configure Kerberos authentication for an Oracle Database client, you can use the KERBEROS5_PRINCIPAL parameter to specify multiple Kerberos principals with a single Oracle Database client. This is an optional parameter. When specified, it is used to verify if the principal name in the credential cache matches the parameter value.

Use this parameter with the <code>CONNECT_DATA</code> parameter. Alternatively, you can specify <code>KERBEROS5_CC_NAME</code> in the connect string along with the <code>KERBEROS5_PRINCIPAL</code> parameter to connect as a different Kerberos principal. Each Kerberos principal must have a valid credential cache.

See KERBEROS5_PRINCIPAL and SQLNET.KERBEROS5_CC_NAME.

Deprecated Features

These features are deprecated in this release and may be desupported in a future release.

Deprecation of the SERVICE_NAMES Initialization Parameter



Starting with Oracle Database 19c, customer use of the <code>SERVICE_NAMES</code> parameter is deprecated. It can be desupported in a future release.

The use of the <code>SERVICE_NAMES</code> parameter is no longer actively supported. It must not be used for high availability (HA) deployments. It is not supported to use service names parameter for any HA operations. This restriction includes FAN, load balancing, <code>FAILOVER_TYPE</code>, <code>FAILOVER_RESTORE</code>, <code>SESSION_STATE_CONSISTENCY</code>, and any other uses.

To manage your services, Oracle recommends that you use the SRVCTL or GDSCTL command line utilities, or the DBMS SERVICE package.



The SERVICE_NAMES parameter that is deprecated is different from the SERVICE_NAME parameter in Oracle Net connect strings. The SERVICE_NAME parameter is still valid.

Deprecation of Weak Native Network Encryption and Integrity Algorithms

The DES, DES40, 3DES112, 3DES168, RC4_40, RC4_56, RC4_128, RC4_256, and MD5 algorithms are deprecated in this release.

As a result of this deprecation, Oracle recommends that you review your network encryption and integrity configuration to check if you have specified any of the deprecated weak algorithms.

To transition your Oracle Database environment to use stronger algorithms, download and install the patch described in My Oracle Support note 2118136.2.

Related Topics

Oracle Database Security Guide



1

Listener Control Utility

This chapter describes the Listener Control Utility commands and syntax. The terms **SQL*Net** and **Net Services** are used interchangeably throughout Oracle documentation.

- Listener Control Utility Overview
- SET and SHOW Commands of the Listener Control utility
 The SET and SHOW commands enable you to alter and view listener configuration parameters.
- Distributed Operations
 The Listener Control utility can perform operations on a local or a remote listener.
- Oracle Net Listener Security
 Authentication for listener administration depends on whether you access the listener locally, or remotely.
- Listener Control Utility Commands
 Use the Listener Control utility commands to manage and configure listeners.

1.1 Listener Control Utility Overview

The Listener Control utility enables you to administer listeners. To perform basic management functions on one or more listeners, you can use the Listener Control utility commands. You can also view and change parameter settings.

The basic syntax of Listener Control utility commands is as follows:

```
lsnrctl command listener name
```

In the preceding command, <code>listener_name</code> is the name of the listener that you want to administer. If you do not specify a specific listener in the command string, then the command is directed to the default listener name, <code>LISTENER</code>.

You can also issue Listener Control utility commands at the LSNRCTL> program prompt. To obtain the prompt, enter lsnrctl with no arguments at the operating system command line. When you run lsnrctl, the program is started. You can then enter the necessary commands from the program prompt. The basic syntax of issuing commands from LSNRCTL> program prompt is as follows:

```
lsnrctl
LSNRCTL> command listener name
```

You can combine commands in a standard text file, and then run them as a sequence of commands. To run in batch mode, use the following format:

```
lsnrctl @file name
```

To identify comments in the batch script, you can use either REM or #. All other lines are considered commands. Any commands that typically require confirmation do not require confirmation during batch processing.

For most commands, the Listener Control utility establishes an Oracle Net connection with the listener that is used to transmit the command. To initiate an Oracle Net connection to the listener, the Listener Control utility must obtain the protocol addresses for the named listener or a listener named LISTENER. This is done by resolving the listener name with one of the following mechanisms:

- listener.ora file in the directory specified by the TNS ADMIN environment variable.
- listener.ora file in the ORACLE HOME/network/admin directory.
- Naming method; for example, a tnsnames.ora file.

If none of the preceding mechanisms resolve the listener name, then the Listener Control utility uses the default listener name LISTENER, resolves the host name IP address, and uses port 1521.

The Listener Control utility supports the following types of commands:

- Operational commands, such as START, and STOP.
- Modifier commands, such as SET TRC LEVEL.
- Informational commands, such as STATUS, and SHOW LOG_FILE.

Related Topics

START

The Listener Control utility command START starts the named listener.

STOP

The Listener Control utility command STOP stops the named listener.

SET TRC LEVEL

The Listener Control utility command SET TRC_LEVEL sets a specific level of tracing for the listener.

STATUS

The Listener Control utility command STATUS displays basic status information about a listener

1.2 SET and SHOW Commands of the Listener Control utility

The SET and SHOW commands enable you to alter and view listener configuration parameters.

You can use the SET command to alter parameter values for a specified listener. You set the name of the listener to administer using the SET CURRENT_LISTENER command. Parameter values remain in effect until the listener is shut down. If you want these settings to persist, then use the SAVE CONFIG command to save changes to the listener.ora file.

You can use the SHOW command to display the current value of a configuration setting.

Related Topics

SET

The Listener Control utility command SET alters the parameter values for the listener.

SET CURRENT LISTENER

The Listener Control utility command SET CURRENT_LISTENER sets the name of the listener that you want to administer.



SAVE CONFIG

The Listener Control utility command SAVE_CONFIG save the current configuration state of the listener to the listener.ora file.

SHOW

The Listener Control utility command SHOW displays the current parameter values for the listener.

1.3 Distributed Operations

The Listener Control utility can perform operations on a local or a remote listener.

Set Up a Computer to Remotely Administer a Listener

Ensure that the Listener Control utility (lsnrctl) executable is installed in the <code>ORACLE_HOME/bin</code> directory. You can resolve the name of the listener that you want to administer either through a listener.ora file, or by a naming method.

When you administer a listener remotely, you can issue all commands except START. However, the Listener Control utility can only start the listener on the same computer from which the utility is running.

When issuing commands, specify the listener name as an argument. If you omit the listener name in the command, then the listener name set with the command SET CURRENT_LISTENER is used. If the listener name is not set with that command, then the command is directed to the default listener name, LISTENER.

Example 1-1 Issuing Commands Using the Listener Control Utility

LSNRCTL> SERVICES lsnr

1.4 Oracle Net Listener Security

Authentication for listener administration depends on whether you access the listener locally, or remotely.

Local listener administration is secure through local operating system authentication, which restricts listener administration to the user account that started the listener, or to the super user. By default, remote listener administration is disabled.

Oracle recommends that you perform listener administration in the default mode, and access the system remotely using a remote login. When you administer the listener remotely, use either Oracle Enterprise Manager Cloud Control or Secure Shell (SSH) to access the remote host.

1.5 Listener Control Utility Commands

Use the Listener Control utility commands to manage and configure listeners.

EXIT

The Listener Control utility command EXIT exits from the Listener Control utility, and returns you to the operating system prompt.

HELP

The Listener Control utility command HELP provides a list of all the Listener Control utility commands, or provides syntax help for a particular Listener Control utility command.

OUIT

The Listener Control utility command QUIT exits from the Listener Control utility and returns you to the operating system prompt.

RELOAD

The Listener Control utility command RELOAD reloads the listener.ora file, so that you can add or change statically configured services without stopping the listener.

SAVE CONFIG

The Listener Control utility command SAVE_CONFIG save the current configuration state of the listener to the listener.ora file.

SERVICES

The Listener Control utility command SERVICES returns detailed information about the database services, instances, and service handlers to which the listener forwards client connection requests.

SET

The Listener Control utility command SET alters the parameter values for the listener.

SET CURRENT LISTENER

The Listener Control utility command SET CURRENT_LISTENER sets the name of the listener that you want to administer.

SET DISPLAYMODE

The Listener Control utility command SET DISPLAYMODE

SET INBOUND CONNECT TIMEOUT

The Listener Control utility command SET INBOUND_CONNECT_TIMEOUT specifies the time, in seconds, for the client to complete its connect request to the listener after establishing the network connection.

SET LOG DIRECTORY

The Listener Control utility command SET LOG_DIRECTORY sets the destination directory where the listener log file is written.

SET LOG_FILE

The Listener Control utility command SET LOG FILE sets the name for the listener log file.

SET LOG STATUS

The Listener Control utility command SET LOG STATUS turns listener logging on or off.

SET SAVE CONFIG ON STOP

The Listener Control utility command SET SAVE_CONFIG_ON_STOP specifies whether changes made to the parameter values for the listener by the SET command are saved to the listener.ora file at the time that the listener is stopped with the STOP command.

SET TRC DIRECTORY

The Listener Control utility command SET TRC_DIRECTORY sets the destination directory where the listener trace files are written.

SET TRC FILE

The Listener Control utility command SET TRC FILE sets the name of the listener trace file.

SET TRC LEVEL

The Listener Control utility command SET TRC_LEVEL sets a specific level of tracing for the listener.

SHOW

The Listener Control utility command SHOW displays the current parameter values for the listener.

SPAWN

The Listener Control utility command SPAWN starts a program stored on the computer on which the listener is running, and that is listed with an alias in the listener.ora file.

START

The Listener Control utility command START starts the named listener.

STATUS

The Listener Control utility command STATUS displays basic status information about a listener.

STOP

The Listener Control utility command STOP stops the named listener.

TRACE

The Listener Control utility command TRACE sets tracing for the listener.

VERSION

The Listener Control utility command VERSION displays the current version of the Listener Control utility.

1.5.1 EXIT

The Listener Control utility command EXIT exits from the Listener Control utility, and returns you to the operating system prompt.

Purpose

To exit from the Listener Control utility, and return to the operating system prompt.

Prerequisites

None

Syntax

From the Listener Control utility:

LSNRCTL> EXIT

Arguments

None

Usage Notes

This command is identical to the QUIT command.

Example

LSNRCTL> EXIT



1.5.2 HELP

The Listener Control utility command HELP provides a list of all the Listener Control utility commands, or provides syntax help for a particular Listener Control utility command.

Purpose

To provide a list of all the Listener Control utility commands or provide syntax help for a particular Listener Control utility command.

Prerequisites

None

Syntax

From the operating system:

```
lsnrctl HELP command
```

From the Listener Control utility:

```
LSNRCTL> HELP command
```

Arguments

command: The Listener Control utility command. Commands are shown in the following example output.

When you enter a command as an argument to \mathtt{HELP} , the Listener Control utility displays information about how to use the command. When you enter \mathtt{HELP} without an argument, the Listener Control utility displays a list of all the commands.

Example

```
LSNRCTL> HELP
The following operations are available
An asterisk (*) denotes a modifier or extended command:
exit
quit
reload
services
set*
show*
spawn
start
status
stop
trace
version
```



1.5.3 QUIT

The Listener Control utility command QUIT exits from the Listener Control utility and returns you to the operating system prompt.

Purpose

To exit from the Listener Control utility and return to the operating system prompt.

Prerequisites

None

Syntax

From the Listener Control utility:

LSNRCTL> QUIT

Arguments

None

Usage Notes

This command is identical to the EXIT command.

Example

LSNRCTL> QUIT

1.5.4 RELOAD

The Listener Control utility command RELOAD reloads the listener.ora file, so that you can add or change statically configured services without stopping the listener.

Purpose

To reload the listener.ora file. This command enables you to add or change statically configured services without actually stopping the listener.

When you run this command, the database services, instances, service handlers, and listening endpoints previously registered dynamically with the listener are unregistered, and subsequently registered again.

To obtain a lightweight reload without dropping registration, use the option <code>-with_ha</code>. Using this option ensures that registered services remain available to clients during reload.

Prerequisites

None



Syntax

From the operating system:

```
lsnrctl RELOAD [-with_ha] listener_name
```

From the Listener Control utility:

```
LSNRCTL> RELOAD [-with ha] listener name
```

Arguments

listener name: The listener name, if the default name of LISTENER is not used.

-with_ha: command option used with RELOAD that indicates that the reload of listener.ora is completed without dropping existing registrations.

Example

```
LSNRCTL> RELOAD

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=sales-server)
(PORT=1521)))

The command completed successfully
```

1.5.5 SAVE_CONFIG

The Listener Control utility command SAVE_CONFIG save the current configuration state of the listener to the listener.ora file.

Purpose

To save the current configuration state of the listener, including trace level, trace file, trace directory, and logging to the listener.ora file. Any changes are stored in listener.ora, preserving formatting, comments, and case as much as possible. Before modification of the listener.ora file, a backup of the file, called listener.bak, is created.

Syntax

From the operating system:

```
lsnrctl SAVE CONFIG listener name
```

From the Listener Control utility:

```
LSNRCTL> SAVE CONFIG listener name
```

Arguments

listener name: The listener name, if the default name of LISTENER is not used.

Usage Notes

This command enables you to save all runtime configuration changes to the listener.ora file.



Example

```
LSNRCTL> SAVE_CONFIG listener
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP) (HOST=sales-server)
(PORT=1521)))
Saved LISTENER configuration parameters.
Listener Parameter File /oracle/network/admin/listener.ora
Old Parameter File /oracle/network/admin/listener.bak
The command completed successfully
```

1.5.6 SERVICES

The Listener Control utility command SERVICES returns detailed information about the database services, instances, and service handlers to which the listener forwards client connection requests.

Purpose

To obtain detailed information about the database services, instances, and service handlers (dispatchers and dedicated servers) to which the listener forwards client connection requests.

Prerequisites

None

Syntax

Arguments

From the operating system:

```
lsnrctl SERVICES listener name
```

From the Listener Control utility:

```
LSNRCTL> SERVICES listener name
```

listener name: The listener name, if the default name of LISTENER is not used.

Usage Notes

The SET DISPLAYMODE command changes the format and the detail level of the output.



Oracle Database Net Services Administrator's Guide for a complete description of ${\tt SERVICES}$ output



Example

This example shows SERVICES output in the default display mode. The output shows the following:

- An instance named sales belonging to two services, sales1.us.example.com and sales2.us.example.com, with a total of three service handlers.
- Service sales1.us.example.com is handled by one dispatcher only.
- Service sales2.us.example.com is handled by one dispatcher and one dedicated server, as specified by in the following output.

```
LSNRCTL> SERVICES
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=net)))
Services Summary...
Service "sales1.us.example.com" has 1 instance(s).
  Instance "sales", status READY, has 1 handler(s) for this service...
   Handler(s):
      "D000" established:0 refused:0 current:0 max:10000 state:ready
         DISPATCHER <machine: sales-server, pid: 5696>
         (ADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=53411))
Service "sales2.us.example.com" has 1 instance(s).
  Instance "sales", status READY, has 2 handler(s) for this service...
   Handler(s):
      "DEDICATED" established: 0 refused: 0 state:ready
         LOCAL SERVER
      "D001" established:0 refused:0 current:0 max:10000 state:ready
         DISPATCHER <machine: sales-server, pid: 5698>
         (ADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=52618))
The command completed successfully
```

Related Topics

SET DISPLAYMODE

The Listener Control utility command SET DISPLAYMODE

1.5.7 SET

The Listener Control utility command SET alters the parameter values for the listener.

Purpose

To alter the parameter values for the listener. Parameter value changes remain in effect until the listener is shut down. To make the changes permanent, use the SAVE_CONFIG command to save changes to the <code>listener.ora</code> file.

Prerequisites

None

Syntax

From the operating system:

lsnrctl SET parameter

From the Listener Control utility:

```
LSNRCTL> SET parameter
```

Arguments

parameter: A SET parameter to modify its configuration setting. Parameters are shown in the example output.

When you enter SET without an argument, the Listener Control utility displays a list of all the parameters.

Usage Notes

If you are using the SET commands to alter the configuration of a listener other than the default LISTENER listener, then use the SET CURRENT_LISTENER command to set the name of the listener to administer.

Example

```
LSNRCTL> SET

The following operations are available with set.

An asterisk (*) denotes a modifier or extended command.

current_listener

displaymode

inbound_connect_timeout

log_file

log_directory

log_status

rawmode

save_config_on_stop

trc_file

trc_directory

trc level
```

Related Topics

SAVE CONFIG

The Listener Control utility command SAVE_CONFIG save the current configuration state of the listener to the listener.ora file.

SET CURRENT_LISTENER

The Listener Control utility command \mathtt{SET} CURRENT_LISTENER sets the name of the listener that you want to administer.

1.5.8 SET CURRENT LISTENER

The Listener Control utility command SET CURRENT_LISTENER sets the name of the listener that you want to administer.

Purpose

To set the name of the listener that you want to administer. After you set the listener name with this command, you can issue subsequent commands that normally require <code>listener_name</code> without specifying the listener.



Syntax

From the Listener Control utility:

```
LSNRCTL> SET CURRENT LISTENER listener name
```

Arguments

listener name: The listener name, if you are not using the default name LISTENER.

Usage Notes

When you specify a listener name using SET CURRENT_LISTENER, the Listener Control utility commands act on the listener name that you specify with this command. You do not have to continue to specify the name of the listener.

Example

```
LSNRCTL> SET CURRENT_LISTENER lsnr
Current Listener is lsnr
```

1.5.9 SET DISPLAYMODE

The Listener Control utility command SET DISPLAYMODE

Purpose

To change the format and level of detail for the SERVICES and STATUS commands.

Syntax

From the Listener Control utility:

```
LSNRCTL> SET DISPLAYMODE {compat | normal | verbose | raw}
```

Arguments

Specify one of the following modes:

compat: Output that is compatible with earlier releases of the listener.

normal: Output that is formatted and descriptive. Oracle recommends this mode.

verbose: All data received from the listener in a formatted and descriptive output.

raw: All data received from the listener without any formatting. This argument should be used only if recommended by Oracle Support Services.

Example

```
LSNRCTL> SET DISPLAYMODE normal Service display mode is NORMAL
```



Related Topics

SERVICES

The Listener Control utility command SERVICES returns detailed information about the database services, instances, and service handlers to which the listener forwards client connection requests.

STATUS

The Listener Control utility command STATUS displays basic status information about a listener.

1.5.10 SET INBOUND CONNECT TIMEOUT

The Listener Control utility command SET INBOUND_CONNECT_TIMEOUT specifies the time, in seconds, for the client to complete its connect request to the listener after establishing the network connection.

Purpose

To specify the time, in seconds, for the client to complete its connect request to the listener after establishing the network connection.

If the listener does not receive the client request in the time specified, then it terminates the connection. In addition, the listener logs the IP address of the client and an ORA-12525:TNS: listener has not received client's request in time allowed error message to the listener.log file.



Oracle Database Net Services Administrator's Guide for additional information about specifying the time out for client connections

Syntax

From the Listener Control utility:

```
LSNRCTL> SET INBOUND CONNECT TIMEOUT time
```

Arguments

time: The time in seconds. Default setting is 60 seconds.

Example

```
LSNRCTL> SET INBOUND_CONNECT_TIMEOUT 2
Connecting to (ADDRESS=(PROTOCOL=TCP) (HOST=sales-server) (PORT=1521))
LISTENER parameter "inbound_connect_timeout" set to 2
The command completed successfully.
```



1.5.11 SET LOG_DIRECTORY

The Listener Control utility command SET LOG_DIRECTORY sets the destination directory where the listener log file is written.

Purpose

To set destination directory where the listener log file is written. By default, the log file is written to the <code>ORACLE HOME/network/log</code> directory.



This command works only if Automatic Diagnostic Repository (ADR) is not enabled. The default is for ADR to be enabled, and to use the log directory ORACLE HOME/log/diag/product type.

Prerequisites

None

Syntax

From the operating system:

lsnrctl SET LOG_DIRECTORY directory

From the Listener Control utility:

LSNRCTL> SET LOG_DIRECTORY directory

Arguments

directory: The directory path of the listener log file.

Example

LSNRCTL> SET LOG_DIRECTORY /usr/oracle/admin
Connecting to (ADDRESS=(PROTOCOL=TCP) (HOST=sales-server) (PORT=1521))
LISTENER parameter "log_directory" set to /usr/oracle/admin
The command completed successfully

1.5.12 SET LOG_FILE

The Listener Control utility command SET LOG FILE sets the name for the listener log file.

Purpose

To set the name for the listener log file. By default, the log file name is listener.log.



This command works only if Automatic Diagnostic Repository (ADR) is not enabled. The default is for ADR to be enabled, and use the log directory <code>ORACLE_HOME/log/diag/product type</code>.

Prerequisites

None

Syntax

From the operating system:

```
lsnrctl SET LOG FILE file name
```

From the Listener Control utility:

```
LSNRCTL> SET LOG_FILE file_name
```

Arguments

file name: The file name of the listener log.

Example

```
LSNRCTL> SET LOG_FILE list.log
Connecting to (ADDRESS=(PROTOCOL=TCP) (HOST=sales-server) (PORT=1521))
LISTENER parameter "log_file" set to list.log
The command completed successfully
```

1.5.13 SET LOG_STATUS

The Listener Control utility command SET LOG STATUS turns listener logging on or off.

Purpose

To turn listener logging on or off.

Prerequisites

None

Syntax

From the operating system:

```
lsnrctl SET LOG STATUS {on | off}
```

From the Listener Control utility:

LSNRCTL> SET LOG STATUS {on | off}

Arguments

on: To turn logging on.

off: To turn logging off.

Example

```
LSNRCTL> SET LOG_STATUS on
Connecting to (ADDRESS=(PROTOCOL=TCP) (HOST=sales-server) (PORT=1521))
LISTENER parameter "log_status" set to ON
The command completed successfully
```

1.5.14 SET SAVE CONFIG ON STOP

The Listener Control utility command SET SAVE_CONFIG_ON_STOP specifies whether changes made to the parameter values for the listener by the SET command are saved to the listener.ora file at the time that the listener is stopped with the STOP command.

Purpose

To specify whether changes made to the parameter values for the listener by the SET command are saved to the listener.ora file at the time that the listener is stopped with the STOP command.

When changes are saved, the Listener Control utility tries to preserve formatting, comments, and letter case. Before the command modifies the listener.ora file, it creates a backup of the file, called listener.bak.

To have all parameters saved immediately, use the SAVE CONFIG command.

Syntax

From the operating system:

```
lsnrctl SET SAVE CONFIG ON STOP {on | off}
```

From the Listener Control utility:

```
LSNRCTL> SET SAVE CONFIG ON STOP {on | off}
```

Arguments

on: To save configuration to listener.ora.

off: To not save configuration to listener.ora.

Example

```
LSNRCTL> SET SAVE_CONFIG_ON_STOP on LISTENER parameter "save_config_on_stop" set to ON The command completed successfully
```



Related Topics

SET

The Listener Control utility command SET alters the parameter values for the listener.

STOP

The Listener Control utility command STOP stops the named listener.

SAVE CONFIG

The Listener Control utility command SAVE_CONFIG save the current configuration state of the listener to the listener.ora file.

1.5.15 SET TRC_DIRECTORY

The Listener Control utility command SET TRC_DIRECTORY sets the destination directory where the listener trace files are written.

Purpose

To set the destination directory where the listener trace files are written. By default, the trace file are written to the <code>ORACLE HOME/network/trace</code> directory.



This command works only if Automatic Diagnostic Repository (ADR) is not enabled. The default is for ADR to be enabled, and use the log directory <code>ORACLE_HOME/log/diag/product type</code>.

Prerequisites

None

Syntax

From the operating system:

lsnrctl SET TRC_DIRECTORY directory

From the Listener Control utility:

LSNRCTL> SET TRC_DIRECTORY directory

Arguments

directory: The directory path of the listener trace files.

Example

LSNRCTL> SET TRC_DIRECTORY /usr/oracle/admin
Connecting to (ADDRESS=(PROTOCOL=TCP) (HOST=sales-server) (PORT=1521))
LISTENER parameter "trc_directory" set to /usr/oracle/admin
The command completed successfully



1.5.16 SET TRC_FILE

The Listener Control utility command \mathtt{SET} \mathtt{TRC} \mathtt{FILE} sets the name of the listener trace file.

Purpose

To set the name of the listener trace file. By default, the trace file name is listener.trc.



This command works only if Automatic Diagnostic Repository (ADR) is not enabled. The default is for ADR to be enabled, and use the log directory <code>ORACLE_HOME/log/diag/product type</code>.

Prerequisites

None

Syntax

From the operating system:

```
lsnrctl SET TRC_FILE file_name
```

From the Listener Control utility:

```
LSNRCTL> SET TRC_FILE file_name
```

Arguments

file name: The file name of the listener trace.

Example

```
LSNRCTL> SET TRC_FILE list.trc

Connecting to (ADDRESS=(PROTOCOL=TCP) (HOST=sales-server) (PORT=1521))

LISTENER parameter "trc_file" set to list.trc

The command completed successfully
```

1.5.17 SET TRC_LEVEL

The Listener Control utility command SET TRC_LEVEL sets a specific level of tracing for the listener.

Purpose

To set a specific level of tracing for the listener.

Prerequisites

None



From the operating system:

```
lsnrctl SET TRC_LEVEL level
```

From the Listener Control utility:

```
LSNRCTL> SET TRC LEVEL level
```

Arguments

level: One of the following trace levels:

- off for no trace output
- user for user trace information
- admin for administration trace information
- support for Oracle Support Services trace information

Usage Notes

This command has the same functionality as the TRACE command.

Example

```
LSNRCTL> SET TRC_LEVEL admin
Connecting to (ADDRESS=(PROTOCOL=TCP) (HOST=sales-server) (PORT=1521))
LISTENER parameter "trc_level" set to admin
The command completed successfully
```

Related Topics

TRACE

The Listener Control utility command TRACE sets tracing for the listener.

1.5.18 SHOW

The Listener Control utility command SHOW displays the current parameter values for the listener.

Purpose

To view the current parameter values for the listener.

All of the SET parameters have equivalent SHOW parameters.

Prerequisites

None



From the operating system:

```
lsnrctl SHOW parameter
```

From the Listener Control utility:

```
LSNRCTL> SHOW parameter
```

Arguments

parameter: A parameter whose configuration settings you want to review. Parameters are shown in the example output.

When you enter SHOW without an argument, the Listener Control utility displays a list of all the parameters.

Example

```
LSNRCTL> SHOW

The following properties are available with SHOW:

An asterisk (*) denotes a modifier or extended command:

current_listener

displaymode
inbound_connect_timeout

log_file

log_directory

log_status

rawmode

save_config_on_stop

trc_file

trc_directory

trc level
```

Related Topics

SET

The Listener Control utility command SET alters the parameter values for the listener.

1.5.19 SPAWN

The Listener Control utility command SPAWN starts a program stored on the computer on which the listener is running, and that is listed with an alias in the listener.ora file.

Purpose

To start a program stored on the computer on which the listener is running, and that is listed with an alias in the listener.ora file.

Prerequisites

None



From the operating system:

```
lsnrctl SPAWN listener_name alias (arguments='arg1, arg2,...')
```

From the Listener Control utility:

```
LSNRCTL> SPAWN listener name alias (arguments='arg1, arg2,...')
```

Arguments

listener name: The listener name, if the default name of LISTENER is not used.

alias: The alias of the program to be spawned off is specified by a listener.ora file entry, similar to the following:

```
alias = (PROGRAM=(NAME=) (ARGS=) (ENVS=))
```

For example:

```
nstest = (PROGRAM=(NAME=nstest) (ARGS=test1) (ENVS='ORACLE_HOME=/usr/oracle'))
```

Example

The nstest program, shown in the preceding section, can then be spawned off using the following command:

```
lsnrctl SPAWN listener name nstest
```

1.5.20 START

The Listener Control utility command START starts the named listener.

Purpose

To start the named listener.

Prerequisites

The listener must not be running.

Syntax

From the operating system:

```
lsnrctl START listener name
```

From the Listener Control utility:

LSNRCTL> START listener_name



On Microsoft Windows, if the database was installed with the Oracle Home User, then the utility can prompt for a password. The password is the operating system password for the Oracle Home User. The prompt is displayed only if the listener service does not exist, and it needs to be created as part of starting the listener.

Arguments

listener name: The listener name, if the default name of LISTENER is not used.

Usage Notes

To start a listener configured in the <code>listener.ora</code> file with a name other than <code>LISTENER</code>, include that name.

For example, if the listener name is top lsnr, enter:

```
lsnrctl START tcp lsnr
```

From the Listener Control utility:

```
LSNRCTL> START tcp lsnr
```

```
LSNRCTL> START
Starting /private/sales group/sales/bin/tnslsnr: please wait...
TNSLSNR for Linux: Version 18.0.0.0.0
System parameter file is $ORACLE HOME/network/admin/listener.ora
Log messages written to $ORACLE_BASE/diag/tnslsnr/node_name/listener/alert/
log.xml
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)
(PORT=1521)))
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=sales-server)
(PORT=1521)))
STATUS of the LISTENER
Alias
                       LISTENER
Version
                      TNSLSNR for Linux: Version 18.0.0.0.0
Start Date
                      21-JAN-2018 21:50:49
                      0 days 0 hr. 0 min. 0 sec
Uptime
Trace Level
                      off
Security
                      ON: Local OS Authetication
                       OFF
SNMP
Listener Log File $ORACLE BASE/diag/tnslsnr/node name/listener/alert/
log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=sales-server)(PORT=1521)))
```

The listener supports no services
The command completed successfully



Oracle Database Administrator's Reference for Microsoft Windows for information about the Oracle Home User

1.5.21 STATUS

The Listener Control utility command STATUS displays basic status information about a listener.

Purpose

To display basic status information about a listener, including a summary of listener configuration settings, listening protocol addresses, and a summary of services registered with the listener.



You can also obtain the status of the listener through the Oracle Enterprise Manager Cloud Control console.

Prerequisites

None

Syntax

From the operating system:

lsnrctl STATUS listener name

From the Listener Control utility:

LSNRCTL> STATUS listener name

Arguments

listener name: The listener name, if the default name of LISTENER is not used.

Usage Notes

The SET DISPLAYMODE command changes the format and level of the detail of the output.





Oracle Database Net Services Administrator's Guide for a complete description of STATUS output

Example

The following example shows STATUS output in the default display mode. The output contains:

- Listener configuration settings
- Listening endpoints summary
- Services summary, which is an abbreviated version of the SERVICES command output

```
LSNRCTL> STATUS
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=net)))
STATUS of the LISTENER
Alias
                        LISTENER
                        TNSLSNR for Linux: Version 18.0.0.0.0 -
Version
Production
Start Date
                        12-JAN-2018 12:02:00
Uptime
                       0 days 0 hr. 5 min. 29 sec
Trace Level
                        support
Security
                        OFF
SNMP
                        OFF
Listener Parameter File /oracle/network/admin/listener.ora
Listener Log File /oracle/network/log/listener.log
Listener Trace File /oracle/network/trace/listener.trc
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=net)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcps)(HOST=sales-server)(PORT=2484)))
Services Summary...
Service "sales1.us.example.com" has 1 instance(s).
  Instance "sales", status READY, has 1 handler(s) for this service...
Service "sales2.us.example.com" has 1 instance(s).
  Instance "sales", status READY, has 2 handler(s) for this service...
The command completed successfully
```

Related Topics

SET DISPLAYMODE

The Listener Control utility command SET DISPLAYMODE

SERVICES

The Listener Control utility command SERVICES returns detailed information about the database services, instances, and service handlers to which the listener forwards client connection requests.

1.5.22 STOP

The Listener Control utility command STOP stops the named listener.

Purpose

To stop the named listener.

Prerequisites

The listener must be running.

Syntax

From the operating system:

```
lsnrctl STOP listener name
```

From the Listener Control utility:

```
LSNRCTL> STOP listener_name
```

Arguments

 ${\it listener_name}. \ \ \textbf{The listener name, if the default name of LISTENER is not used}.$

Example

```
LSNRCTL> STOP
Connecting to (ADDRESS=(PROTOCOL=TCP) (HOST=sales-server) (PORT=1521))
The command completed successfully
```

1.5.23 TRACE

The Listener Control utility command TRACE sets tracing for the listener.

Purpose

To set tracing for the listener.

Syntax

From the operating system:

```
lsnrctl trace level listener name
```

From the Listener Control utility:

LSNRCTL> trace level listener name



Arguments

level: One of the following trace levels:

- off for no trace output
- user for user trace information
- admin for administration trace information
- support for Oracle Support Services trace information

listener name: Specify the listener name, if the default name of LISTENER is not used.

Usage Notes

This command has the same functionality as the SET TRC_LEVEL command.

Example

```
LSNRCTL> TRACE ADMIN lsnr
Connecting to (ADDRESS=(PROTOCOL=TCP)(HOST=sales-server)(PORT=1521))
Opened trace file: /oracle/network/trace/listener.trc
The command completed successfully
```

1.5.24 VERSION

The Listener Control utility command VERSION displays the current version of the Listener Control utility.

Purpose

To display the current version of Listener Control utility.

Prerequisites

None

Syntax

From the operating system:

```
lsnrctl VERSION listener name
```

From the Listener Control utility:

```
LSNRCTL> VERSION listener name
```

Arguments

listener name: The listener name, if the default name of LISTENER is not used.

```
LSNRCTL> version listener
Connecting to ADDRESS=(PROTOCOL=TCP) (HOST=sales-server) (PORT=1521))
```

TNSLSNR for Linux: Version 19.0.0.0.0

TNS for Linux: Version 19.0.0.0.0

Oracle Bequeath NT Protocol Adapter for Linux: Version 19.0.0.0.0

Unix Domain Socket IPC NT Protocol Adaptor for Linux: Version

19.0.0.0.0

TCP/IP NT Protocol Adapter for Linux: Version 19.0.0.0.0

The command completed successfully



2

Oracle Connection Manager Control Utility

This chapter describes the commands and syntax of the Oracle Connection Manager Control Utility.

- Command Modes and Syntax
 The Oracle Connection Manager Control Litility (CMCTL) en
 - The Oracle Connection Manager Control Utility (CMCTL) enables you to start up, configure and alter how client connection requests are managed.
- Oracle Connection Manager Control Utility Overview
- Oracle Connection Manager Control Utility Commands
 Use the Oracle Connection Manager Control utility commands to manage and configure
 Oracle Connection Manager instances.

2.1 Command Modes and Syntax

The Oracle Connection Manager Control Utility (CMCTL) enables you to start up, configure and alter how client connection requests are managed.

The basic syntax of the Oracle Connection Manager Control utility is as follows:

```
cmctl command [argument]
```

The Oracle Connection Manager Control utility supports the following types of commands:

- Initialization and termination commands such as STARTUP and SHUTDOWN
- Alter commands such as SET LOG LEVEL and SET EVENT
- Display commands, such as SHOW STATUS and SHOW RULES
- Gateway commands such as SHOW GATEWAYS and RESUME GATEWAYS

Note:

You can use SET commands to dynamically alter configuration parameters that control how the listener receives client connections. The changes only remain in effect until Oracle Connection Manager is shut down. You cannot save them to the cman.ora file. The one exception is the Oracle Connection Manager password, which you can save using the command SAVE PASSWD.

You can use the Oracle Connection Manager Control utility in command mode, or batch mode.

- Using command mode:
 - From the Oracle Connection Manager Control utility:

Enter cmctl at the command line to obtain the program prompt, and then issue the command:

```
cmctl
CMCTL> command
```

From the operating system:

Enter the entire command from the operating system command prompt:

```
cmctl [command] [argument1 . . . argumentN] [-c instance name]
```

Each command issued this way can have an Oracle Connection Manager instance name appended as an argument. If an Oracle Connection Manager instance name is not provided, then the default instance name is assumed. The default name is <code>cman_hostname</code>. If a password was set in a previous CMCTL session, then you can be prompted for a password. If a password has been set, then to issue commands from an Oracle Connection Manager Control utility session of Oracle Connection Manager, you must enter a password once, at the beginning of the session.



Caution:

There is an option to specify the password on the command line. However, doing so exposes the password on the screen, and is a potential security risk. Oracle recommends not using the password option (-p) on the command line.

Using batch mode:

You can combine commands in a standard text file, and then run them as a sequence of commands. To run in batch mode, use the following syntax:

```
cmctl @input file
```

See Also:

Oracle Database Net Services Administrator's Guide for more information about Oracle Connection Manager architecture

2.2 Oracle Connection Manager Control Utility Overview

The Oracle Connection Manager Control utility (CMCTL) enables you to administer Oracle Connection Manager. You can use Oracle Connection Manager Control utility commands to perform basic management functions on one or more Oracle Connection Manager instances. Additionally, you can view and change parameter settings.

2.3 Oracle Connection Manager Control Utility Commands

Use the Oracle Connection Manager Control utility commands to manage and configure Oracle Connection Manager instances.

ADMINISTER

The ADMINISTER command can be issued only from the Oracle Connection Manager utility and lets you select the Oracle Connection Manager instance to administer.

CLOSE CONNECTIONS

It is an Oracle Connection Manager utility command to terminate connections but Oracle Connection Manager must be running at that time.

EXIT

Oracle Connection Manager Control utility command EXIT is used to exit from the Oracle Connection Manager utility.

- HELP
- QUIT
- RELOAD

Reload Oracle Connection Manager Control utility command dynamically rereads parameters and rules.

- RESUME GATEWAYS
- SAVE_PASSWD
- SET
- SET ASO_AUTHENTICATION_FILTER
- SET CONNECTION_STATISTICS
- SET EVENT
- SET IDLE TIMEOUT
- SET INBOUND_CONNECT_TIMEOUT
- SET LOG_DIRECTORY
- SET LOG_LEVEL
- SET OUTBOUND_CONNECT_TIMEOUT
- SET PASSWORD



- SET SESSION TIMEOUT
- SET TRACE_DIRECTORY
- SET TRACE_LEVEL
- SHOW
- SHOW ALL
- SHOW CONNECTIONS
- SHOW DEFAULTS
- SHOW EVENTS
- SHOW GATEWAYS
- SHOW PARAMETERS
- SHOW RULES
- SHOW SERVICES
- SHOW STATUS
- SHOW VERSION
- SHUTDOWN
- STARTUP
- SUSPEND GATEWAY

2.3.1 ADMINISTER

The ADMINISTER command can be issued only from the Oracle Connection Manager utility and lets you select the Oracle Connection Manager instance to administer.

Purpose

To select an Oracle Connection Manager instance.

Prerequisites

None

Syntax

From the Oracle Connection Manager Control utility:

CMCTL> ADMINISTER [-c] instance name

Arguments

instance_name: The instance name of Oracle Connection Manager that you would like to administer. Instances are defined in the <code>cman.ora</code> file.

Usage Notes

You can issue the ADMINISTER command only within the utility. You cannot issue the command from the operating system.

ADMINISTER enables you to choose which Oracle Connection Manager instance to administer. To start the Oracle Connection Manager instance, you must issue the STARTUP command.



When you omit the instance name from the command, the instance administered defaults to the local instance.

Use the -c option when to administer an instance that is not the local instance.

A password is required only if one was provided at installation time or during a previous session of the Oracle Connection Manager.

Example

```
CMCTL> ADMINISTER cman_indl040ad Enter CMAN password: password Current instance cman_indl040ad is already started Connections refer to (address=(protocol=TCP) (host=indl040ad) (port=1560)). The command completed successfully
```

Related Topics

STARTUP

2.3.2 CLOSE CONNECTIONS

It is an Oracle Connection Manager utility command to terminate connections but Oracle Connection Manager must be running at that time.

Purpose

To terminate connections, using specific qualifiers to select connections.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl CLOSE CONNECTIONS [in state] [gt time] [from source] [to destination]
[for service] [using gateway_process_id] [connect_identifier_list]
[-c cman_name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> CLOSE CONNECTIONS [in state] [gt time] [from source] [to destination] [for service] [using gateway_process_id] [connect_identifier_list
```

Arguments

state: One of the following values to specify the connection state:

- idle: Connections that are inactive in the established state.
- connecting: Connections that are in the process of connecting.
- established: Connections that are connected and are transferring data.
- terminating: Connections that are disconnecting.



If no state is specified, then CLOSE CONNECTIONS defaults to all possible states. If the time qualifier is included under these conditions, then the time specified is the amount of time that has elapsed since a client initiated a connection.

time: The time format. Use the following format to specify connections greater than the time indicated:

```
gt[hh:mm:]ss
```

source: The source address. Use one of the following formats to specify the source address:

- from IP
- from hostname
- from subnet

destination: The destination address. Use one of the following formats to specify the destination address:

- to TP
- to hostname
- to subnet

service: The service name. Use the <code>service_name</code> parameter to specify the service, such as <code>sales.us.example.com</code>.

gateway_process_id: The gateway process identifier is a number. Use this number to specify connections that are proxied by the gateway process indicated. To determine the gateway process identifier, use the Oracle Connection Manager control utility show gateways command.

connect_identifier_list: The connection identifiers. Use a space between multiple connection identifiers in a list.

Usage Notes

Because the CLOSE CONNECTIONS command terminates connections, it might generate error messages on both client and server sides.

The IDLE state qualifier always requires a time qualifier.

Issuing CLOSE CONNECTIONS without an argument closes all connections.

Examples

The following example shuts down connections in any state. The elapsed time of the connection must be greater than 1 hour and 30 minutes. The connection source is the specified subnet, and the destination is the specified host name.

```
CMCTL> CLOSE CONNECTIONS gt 1:30:00 from 192.0.2.32/24 to host1
```

The following example shuts down those connections proxied by gateway process 0 that have been in the idle state more than 30 minutes:

```
CMCTL> CLOSE idle CONNECTIONS gt 30:00 using 0
```

The following example shuts down connections that are connected to the service sales.us.example.com:

CMCTL> CLOSE established CONNECTIONS for sales.us.example.com



2.3.3 EXIT

Oracle Connection Manager Control utility command EXIT is used to exit from the Oracle Connection Manager utility.

Purpose

To exit from the Oracle Connection Manager Control utility.

Prerequisites

None

Syntax

From the operating system:

```
cmctl EXIT [-c instance name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> EXIT
```

Usage Notes

This command is identical to the QUIT command.

Example 2-1 Example

CMCTL> EXIT

2.3.4 HELP

Purpose

To provide a list of all commands for the Oracle Connection Manager Control utility or to provide help with the syntax of a particular command.

Prerequisites

None

Syntax

From the operating system:

```
cmctl HELP [command] [-c instance name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> HELP [command]
```

Arguments

command: Specify a HELP command. Commands are shown in the following sample output.

When you enter a command as an argument to HELP, the Oracle Connection Manager Control utility displays information about how to use the command. When you enter HELP without an argument, the Oracle Connection Manager Control utility displays a list of all the commands.



Example

```
CMCTL> HELP
```

The following operations are available

An asterisk (*) denotes a modifier or extended command:

administer close* exit
resume* save_passwd set*
shutdown sleep startup
show_version quit reload show* suspend*

2.3.5 QUIT

Purpose

To exit the Oracle Connection Manager Control utility and return to the operating system prompt.

Prerequisites

None

Syntax

From the operating system:

cmctl QUIT

From the Oracle Connection Manager Control utility:

CMCTL> QUIT

Usage Notes

This command is identical to the EXIT command.

Example

CMCTL> QUIT

2.3.6 RELOAD

Reload Oracle Connection Manager Control utility command dynamically rereads parameters and rules.

Purpose

To dynamically reread parameters and rules.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

cmctl RELOAD [-with_ha] [-c instance_name]



From the Oracle Connection Manager Control utility:

```
CMCTL> RELOAD [-with ha]
```

Arguments

-with ha: It is used to reload cman.ora without dropping registrations

Usage Notes

Configuration information modified using the RELOAD command applies only to new connections. Existing connections are unaffected. The SET RELOAD command restores configurations set in cman.ora, and override the SET command.

RELOAD reregisters gateways with the Oracle Connection Manager listener during which some new connections might be refused until the registration process is complete.

-with_ha option can be used with RELOAD to not drop registrations, thus providing high service availability during reload.

Example

```
CMCTL> RELOAD
The command completed successfully
```

2.3.7 RESUME GATEWAYS

Purpose

To resume gateway processes that have been suspended.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl RESUME GATEWAYS [gateway_process_id] [cman_name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> RESUME GATEWAYS [gateway_process_id]
```

Arguments

gateway_process_id: One or more gateway processes to reopen. Separate multiple gateway processes using a space between the process identifiers.

Usage Notes

Issuing RESUME GATEWAYS without an argument reopens all gateway processes that have been closed.

```
CMCTL> RESUME GATEWAYS 1
The command completed successfully
```



2.3.8 SAVE_PASSWD

Purpose

To save the current password to the cman.ora file, the configuration file for Oracle Connection Manager.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SAVE_PASSWD [-c instance_name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SAVE PASSWD
```

Usage Notes

If you run this command, then the next session of Oracle Connection Manager uses the password. The password is stored in an encrypted format in the <code>cman.ora</code> file.

Example

CMCTL> SAVE PASSWD

2.3.9 SET

Purpose

To display a list of parameters that can be modified using this command.

Prerequisites

None

Syntax

From the operating system:

cmctl SET

From the Oracle Connection Manager Control utility:

CMCTL> SET

```
CMCTL> SET
The following operations are available after set
An asterisk (*) denotes a modifier or extended command:
aso_authentication_filter outbound_connect_timeout
connection_statistics password
event session timeout
```



idle_timeout
inbound_connect_timeout
trace_level
log_directory
log level

trace directory

2.3.10 SET ASO_AUTHENTICATION_FILTER

Purpose

To indicate whether the client must use Oracle Database security to authenticate.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SET ASO AUTHENTICATION FILTER {on | off}[-c instance name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET ASO AUTHENTICATION FILTER {on | off}
```

Arguments

on: To reject connections that are not using Secure Network Service (SNS) to perform client authentication. SNS is part of Oracle Database security.

off: To specify whether no authentication is required for client connections. This is the default.

Example

```
CMCTL> set aso_authentication_filter ON
CMAN_user.us.example.com parameter aso_authentication_filter set to ON
The command completed successfully
```

2.3.11 SET CONNECTION_STATISTICS

Purpose

To specify whether gateway processes collect connection statistics.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SET CONNECTION_STATISTICS {yes | no}[-c instance_name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET CONNECTION_STATISTICS {yes | no}
```



Arguments

yes: To have gateway processes collect connection statistics.

no: To not have gateway processes collect connection statistics. This is the default.

Usage Notes

If SET CONNECTION_STATISTICS is set to yes, then you can obtain statistics by issuing the SHOW CONNECTIONS command.

Example

```
CMCTL> set connection_statistics ON CMAN_user.us.example.com parameter connection_statistics set to ON The command completed successfully
```

2.3.12 SET EVENT

Purpose

To log information for a particular event.

Syntax

From the operating system:

```
cmctl SET EVENT event group [-c instance name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET EVENT event_group {on | off}
```

Arguments

event_group: Specify one of the following event groups:

- init and term: Initialization and termination event group.
- memory ops: Memory operations event group.
- conn hdlg: Connection handling event group.
- proc mgmt: Process management event group.
- reg and load: Registration and load update event group.
- wake_up: Events related to Connection Manager Administration (CMADMIN) wakeup queue event group.
- timer: Gateway timeouts event group.
- cmd proc: Command processing event group.
- relay: Events associated with connection control blocks event group.

on: To turn an event group on.

off: To turn an event group off.



Usage Notes

The SET EVENT command accepts only one argument. To log multiple events, you must issue the command for each event separately.

Example

```
CMCTL> set event memory_ops off cman11 event memory_ops set to OFF. The command completed successfully
```

2.3.13 SET IDLE_TIMEOUT

Purpose

To specify the amount of time a client can be idle without transmitting data.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SET IDLE TIMEOUT [time] [-c instance name]
```

From the From the Oracle Connection Manager Control utility:

```
CMCTL> SET IDLE TIMEOUT [time]
```

Arguments

time: Specify the idle timeout in seconds. The default is 0 (zero), which disables this feature.

Example

```
CMCTL> SET IDLE_TIMEOUT 30
CMAN_user.us.example.com parameter idle_timeout set to 30
The command completed successfully
```

2.3.14 SET INBOUND_CONNECT_TIMEOUT

Purpose

To specify the maximum amount of time the Oracle Connection Manager listener waits for a valid connection request from the client before timing out.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SET INBOUND CONNECT TIMEOUT [time] [-c instance name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET INBOUND CONNECT TIMEOUT [time]
```

Arguments

time: The inbound connect timeout in seconds. The default is 0 (zero), which disables this feature.

Example

```
CMCTL> SET INBOUND_CONNECT_TIMEOUT 30
CMAN_user.us.example.com parameter inbound_connect_timeout set to 30
The command completed successfully
```

2.3.15 SET LOG_DIRECTORY



This command works only if Automatic Diagnostic Repository (ADR) is not enabled. The default is for ADR to be enabled, and use the log directory <code>ORACLE HOME/log</code>.

Purpose

To designate where the log files for Oracle Connection Manager are written.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SET LOG_DIRECTORY [directory_path] [-c instance_name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET LOG_DIRECTORY [directory_path]
```

Arguments

directory_path: The location of the log directory. The default path is as follows:

Linux and UNIX:

```
ORACLE_HOME/network/log directory
```

Microsoft Windows:

```
ORACLE HOME\network\log directory
```

Usage Notes

Use the SHOW PARAMETERS command to determine the location of the log files.

```
CMCTL>
SET LOG_DIRECTORY /disk1/user_cman_test/oracle/network/admin
```

 $\begin{tabular}{ll} $\sf CMAN_user.us.example.com parameter log_directory set to \\ $\sf /disk1/user cman test/oracle/network/admin \end{tabular}$

The command completed successfully

2.3.16 SET LOG_LEVEL

Purpose

To set the log level for Oracle Connection Manager.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SET LOG LEVEL [level] [-c instance name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET LOG_LEVEL [level]
```

Arguments

level: Specify one of the following log levels:

- off: No logging.
- user: User log information.
- admin: Administrative log information.
- support: Oracle Support Services log information. This is the default.

Usage Notes

Specify off to capture the minimum amount of log information. Specify support to capture the maximum amount.

Example

```
CMCTL> SET LOG_LEVEL SUPPORT
CMAN_user.us.example.com parameter log_level set to SUPPORT
The command completed successfully
```

2.3.17 SET OUTBOUND_CONNECT_TIMEOUT

Purpose

To specify the maximum amount of time the Oracle Connection Manager instance waits for a valid connection with the server before timing out.

Prerequisites

Oracle Connection Manager must be running.



From the operating system:

```
cmctl SET OUTBOUND_CONNECT_TIMEOUT [time] [-c instance_name]
```

From the From the Oracle Connection Manager Control utility:

```
CMCTL> SET OUTBOUND CONNECT TIMEOUT [time]
```

Arguments

time: The outbound connect timeout in seconds. The default is 0.

Example

```
CMCTL> SET OUTBOUND_CONNECT_TIMEOUT 30

CMAN_user.us.example.com parameter outbound_connect_timeout set to 30

The command completed successfully
```

2.3.18 SET PASSWORD

Purpose

To assign a password to the Oracle Connection Manager instance.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SET PASSWORD
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET PASSWORD
```

Arguments

None.

Usage Notes

This command may be used either to set a password for the first time or to change an existing one.

This command does not save the password to cman.ora. As a result the password is valid only for the current session. To save the password after you have set it, run the SAVE_PASSWD command.

```
CMCTL> SET PASSWORD
```

```
Enter Old password: old_password
Enter New password: new_password
Reenter New password: new password
```



The command completed successfully

2.3.19 SET SESSION_TIMEOUT

Purpose

To specify the maximum amount of time for a session of Oracle Connection Manager.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SET SESSION TIMEOUT [time] [-c instance name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET SESSION TIMEOUT [time]
```

Arguments

time: The session timeout in seconds. The default is 0 (zero), which disables this feature.

Example

```
CMCTL> SET SESSION_TIMEOUT 60
CMAN_user.us.example.com parameter session_timeout set to 60
The command completed successfully
```

2.3.20 SET TRACE_DIRECTORY



This command works only if Automatic Diagnostic Repository (ADR) is not enabled. The default is for ADR to be enabled.

Purpose

To designate where the trace files for an Oracle Connection Manager instance are written.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SET TRACE_DIRECTORY [directory_path] [-c instance_name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET TRACE DIRECTORY [directory path]
```



Arguments

directory_path: The location of the trace directory. The default path is <code>ORACLE_HOME/network/trace</code>.

Usage Notes

Use the SHOW PARAMETERS command to determine the location of the trace files.

Example

```
CMCTL> SET TRACE_DIRECTORY /disk1/mpurayat_newtest/oracle/network/trace cman1 parameter trace_directory set to /disk1/mpurayat_newtest/oracle/network/trace
The command completed successfully
```

2.3.21 SET TRACE_LEVEL

Purpose

To set the trace level for an Oracle Connection Manager instance.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SET TRACE LEVEL [level] [-c instance name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SET TRACE_LEVEL [level]
```

Arguments

level: Specify one of the following log levels:

- off: No tracing. This is the default.
- user: User trace information.
- admin: Administrative trace information.
- support: Oracle Support Services trace information.

Usage Notes

Specify off to capture the minimum amount of trace information. Specify support to capture the maximum amount.

Use the SHOW PARAMETERS command to determine the current trace level.

```
CMCTL> SET TRACE_LEVEL USER

CMAN_user.us.example.com parameter trace_level set to USER

The command completed successfully
```



2.3.22 SHOW

Purpose

To display a list of parameters that may be used as arguments for this command. Entering one of these parameters with the command displays the parameter value or values.

Prerequisites

None

Syntax

From the operating system:

```
cmctl SHOW [-c instance name]
```

From the Oracle Connection Manager Control utility:

CMCTL> SHOW

Example

```
CMCTL> SHOW

The following operations are available after show

An asterisk (*) denotes a modifier or extended command:
```

```
all gateways status connections parameters version defaults rules events services
```

2.3.23 SHOW ALL

Purpose

To combine and display output from the SHOW PARAMETERS and SHOW RULES commands.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SHOW ALL [-c instance_name]
```

From the Oracle Connection Manager Control utility:

CMCTL> SHOW ALL



```
| /disk1/user_cman_test/oracle/network/log/
log directory
log level
                      SUPPORT
max connections
                          256
                            0
idle timeout
inbound_connect_timeout |
                            0
session timeout
                           0
outbound connect timeout |
                           0
max_gateway_processes |
                          16
min_gateway_processes
                      max_cmctl_sessions
                      4
password
                         OFF
                      trace_directory
trace_level
                     | /disk1/user cman test/oracle/network/trace/
                     | OFF
trace_timestamp
                     | OFF
trace filelen
                     0
trace fileno
                           0
                     (rule_list=
 (rule=
  (src=*)
  (dst=*)
  (srv=*)
  (act=accept)
)
The command completed successfully
```

2.3.24 SHOW CONNECTIONS

Purpose

To display information about specific connections or all connections.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SHOW CONNECTIONS [information] [in state] [gt time] [from source]
[to destination] [for service] [using gateway_process_id]
[connect_identifier_list] [-c instance_name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SHOW CONNECTIONS [information][in state] [gt time] [from source] [to destination] [for service] [using gateway_process_id] [connect_identifier_list]
```

Arguments

information: Specify one of the following values to display information about connections. Information categories include connection identifier, source, destination, service, current state, total idle time, and total elapsed time.

- count: The total number of connections that meet the criteria specified by the other qualifiers. This is the default.
- detail: All information about connections specified by the other qualifiers.

state: Specify one of the following values to specify the connection state:

- idle: Connections that are inactive in the established state.
- connecting: Connections that are in the process of connecting.
- established: Connections that are connected and are transferring data.
- terminating: Connections that are disconnecting.

If no state is specified, then SHOW CONNECTIONS defaults to all possible states. If the time qualifier is included under these conditions, then the time specified is the amount of time that has elapsed since a client initiated a connection.



This argument is not supported with Oracle Connection Manager in Traffic Director mode.

time: Use the following format to specify connections greater than the time indicated:

gt[hh:mm:]ss



This argument is not supported with Oracle Connection Manager in Traffic Director mode.

source: Specify one of the following formats to specify the source address:

- from IP
- from hostname
- from subnet

destination: Specify one of the following formats to specify the destination address:

- to IP
- to hostname
- to subnet

service: Use the service_name format to request a service:

gateway_process_id: Use the following format to specify connections that are proxied by the gateway process indicated:

using gateway process id

connect_identifier_list: Separate multiple connection identifiers using a space.

Usage Notes

Connections are sorted by gateway process identifier and connection identifier, in ascending order.

Issuing Show Connections without an argument displays all connections.

Examples

The following command displays a detailed description of connections in any state. The elapsed time of the connection must be greater than 1 hour and 30 minutes. The connection source is the specified subnet, and the destination the specified host name.

```
CMCTL> SHOW CONNECTIONS gt 1:30:00 from 192.0.2.32/24 to host1
```

The following command displays the number of connections proxied by Oracle Connection Manager using the gateway process identifier 0 that have been in the idle state more than 30 minutes:

```
CMCTL> SHOW idle CONNECTIONS count gt 30:00 using 0
```

The following command displays a detailed description of connections that are connected to the service sales.us.example.com:

CMCTL> SHOW established CONNECTIONS detail for sales.us.example.com

Additional Statistics Shown in Traffic Director Mode

Each connection to Oracle Connection Manager in Traffic Director mode displays these additional statistics:

- Source Host Name: Host name of the client connection.
- Source Process Id: Process Id of the connected client.
- Source Program Name: The name of the connected client program.
- Destination Hostname: Host name of the database server to which the client is connected through Oracle Connection Manager.
- State: State of the inbound connection with one of the following values
 - THREAD WAIT: Connection is waiting for a worker thread, not seen in dedicated threads mode
 - ACTIVE: Connection is transferring data, occupying the thread
 - IDLE: Connection is established but inactive, can still occupy the thread if tdm_bind_thread=true in cman.ora
- Idle time: Cumulative time in \(\mu \) s the connection is in IDLE state.
- Thread Wait time: Cumulative time in μs the connection is in THREAD WAIT state. It is always 0 in dedicated threads mode.
- Active time: Cumulative time in \(\mu \) s the connection is in ACTIVE state.
- PRCP State: State of the inbound connection with respect to the Proxy Resident Connection Pool (PRCP) and can be one of the following values
 - WAIT: Connection is waiting for a session from the PRCP
 - CHECKED-OUT: Connection is holding an outbound session from PRCP but not making any OCI calls
 - ACTIVE: Connection is holding an outbound session from PRCP and busy with OCI calls
 - CHECKED-IN: Connection released the CHECKED-OUT session back to the PRCP
 - NO STATE: Clients to a service without a configured PRCP configured have this state



- PRCP Wait time, PRCP Checked-out time, and PRCP Active time: Cumulative time in properties the connection is in PRCP WAIT, CHECKED-OUT, and ACTIVE states. All these three states are zero in case of non-PRCP service.
- Total Session Gets: Total count of PRCP session get requests from this connection. It is always 1 if PRCP is not configured.
- Session Get Hits: Number of times a session is found existing in the PRCP out of all the requests. It is always 0 if PRCP is not configured.

2.3.25 SHOW DEFAULTS

Purpose

To display default parameter settings.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SHOW DEFAULTS [-c instance name]
```

From the Oracle Connection Manager Control utility:

CMCTL> SHOW DEFAULTS

```
CMCTL> SHOW DEFAULTS
listener address |
(address=(protocol=tcp) (host=users.us.example.com) (port=1521))
aso authentication filter | OFF
connection_statistics | OFF
event_group | OFF
log_directory | /disk1/user_cman_test/oracle/network/log/
log_level | SUPPORT
max_connections | 256
idle timeout | 0
idle_timeout
inbound_connect_timeout |
                                   0
session timeout
outbound connect timeout |
max gateway_processes |
                                   16
min_gateway_processes |
                            max_cmctl_sessions
                                     4
password | OFF
trace_directory | /disk1/user_cman_test/oracle/network/trace/
trace_level | OFF
trace_timestamp | OFF
trace_filelen | 0
trace_filen | OFF

trace_filen | O

trace_filen | O
The command completed successfully
```



2.3.26 SHOW EVENTS

Purpose

To display the events that are in operation.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SHOW EVENTS [-c instance name]
```

From the Oracle Connection Manager Control utility:

CMCTL> SHOW EVENTS

Example

CMCTL> SHOW EVENTS
Event Groups:
memory_ops
The command completed successfully

2.3.27 SHOW GATEWAYS

Purpose

To display the current status of a specific gateway process or processes. Statistics displayed include number of active connections, number of peak active connections, total number of connections handled, and number of connections refused.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SHOW GATEWAYS [gateway] [-c instance name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SHOW GATEWAYS [gateway]
```

Arguments

gateway: The identifier of the gateway or gateways whose status to display.

Issuing Show Gateways without an argument displays the status of all gateway processes.

Usage Notes

To display multiple gateways, use a space to separate the identifiers when entering the command.



Example

CMCTL> SHOW GATEWAYS 1	
Gateway ID	1
Gateway state	READY
Number of active connections	0
Peak active connections	0
Total connections	0
Total connections refused	0
The command completed successfully	

2.3.28 SHOW PARAMETERS

Purpose

To display current parameter settings for an instance.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SHOW PARAMETERS [-c instance name]
```

From the Oracle Connection Manager Control utility:

CMCTL> SHOW PARAMETERS

Usage Notes

Several configuration parameters can be dynamically modified using the SET command. Therefore, the information that SHOW PARAMETERS displays might be different from what appears in the <code>cman.ora</code> file.

```
CMCTL> SHOW PARAMETERS
listener_address |
(address=(protocol=tcp) (host=users.us.example.com) (port=1630))
aso_authentication_filter | ON
connection_statistics | ON
event_group | (memory_ops)
log_directory | /disk1/user_cman_test/oracle/network/log/
log_level | SUPPORT
max_connections | 256
idle_timeout | O
inbound_connect_timeout | O
session_timeout | O
outbound_connect_timeout | O
max_gateway_processes | 16
min_gateway_processes | 2
max_cmctl_sessions | 4
password | OFF
trace_directory | /disk1/user_cman_test/oracle/network/trace/
trace_level | SUPPORT
trace_timestamp | OFF
```



2.3.29 SHOW RULES

Purpose

To display the access control list currently used by the instance.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SHOW RULES [-c instance name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SHOW RULES
```

Usage Notes

You can update the rules list by issuing the RELOAD command.

```
CMCTL> SHOW RULES
Number of filtering rules currently in effect: 5
(rule list=
  (rule=
    (src=usunnae12)
    (dst=usunnae13)
    (srv=*)
    (act=accept)
    (action list=(mit=120) (mct=1800) (conn stats=on) (aut=off))
  (rule=
    (src=usunnae12)
    (dst=usunnae14)
    (srv=service2)
    (act=accept)
  (rule=
    (src=*)
    (dst=usunnae15)
    (srv=*)
    (act=accept)
    (action list=(mit=120) (mct=3000) (moct=200) (aut=on))
  (rule=
    (src=*)
    (dst=usunnae16)
    (srv=*)
    (act=reject)
    (action list=(moct=20)(aut=on))
```

```
(rule=
   (src=users.us.example.com)
   (dst=users.us.example.com)
   (srv=cmon)
   (act=accept)
   (action_list=(mit=100) (mct=1130) (moct=200) (aut=on))
)
```

2.3.30 SHOW SERVICES

Purpose

To display comprehensive information about the Oracle Connection Manager instance. The information displayed includes number of handlers for gateway and CMADMIN processes, listening ports of handlers, and number of connections, both refused and current.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SHOW SERVICES [-c instance name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SHOW SERVICES
```

Example

```
CMCTL> SHOW SERVICES
Services Summary...
Proxy service "cmgw" has 1 instance(s).
 Instance "cman", status READY, has 2 handler(s) for this service...
   Handler(s):
      "cmgw001" established:0 refused:0 current:0 max:256 state:ready
         <machine: user-sun, pid: 29190>
         (ADDRESS=(PROTOCOL=tcp) (HOST=user-sun) (PORT=33175))
      "cmgw000" established:0 refused:0 current:0 max:256 state:ready
         <machine: user-sun, pid: 29188>
         (ADDRESS=(PROTOCOL=tcp) (HOST=user-sun) (PORT=33174))
Service "cmon" has 1 instance(s).
 Instance "cman", status READY, has 1 handler(s) for this service...
   Handler(s):
      "cmon" established: 0 refused: 0 current: 0 max: 4 state: ready
         <machine: user-sun, pid: 29184>
         (ADDRESS=(PROTOCOL=tcp) (HOST=users) (PORT=33168))
The command completed successfully
```

2.3.31 SHOW STATUS

Purpose

To display basic information about the instance, including version, start time, and current statistics.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

cmctl SHOW STATUS

From the Oracle Connection Manager Control utility:

CMCTL> SHOW STATUS

Example

2.3.32 SHOW VERSION

Purpose

To display the current version and name of the Oracle Connection Manager Control utility.

Prerequisites

None

Syntax

From the operating system:

cmctl SHOW VERSION [-c instance name]

From the Oracle Connection Manager Control utility:

CMCTL> SHOW VERSION

Examples

```
CMCTL> SHOW VERSION
CMAN for Linux: Version 18.0.0.0.0
The command completed successfully
```



2.3.33 SHUTDOWN

Purpose

To shut down specific gateway processes or the entire Oracle Connection Manager instance.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SHUTDOWN [gateways gateway] [normal | abort] [-c instance_name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SHUTDOWN [gateways gateway] {normal | abort}
```

Arguments

gateways: To shut down a specific gateway.

normal: To reject new connections and terminate after existing connections close. This is the default.

abort: To shut down Oracle Connection Manager immediately, and close all open connections.

To specify more than one gateway, separate gateways using a space.

Usage Notes

Issuing Shutdown without an argument shuts down all gateways.

Example

```
CMCTL> SHUTDOWN GATEWAYS 0
The command completed successfully
```

2.3.34 STARTUP

Purpose

To start Oracle Connection Manager.

Prerequisites

Another Oracle Connection Manager instance configured with the same protocol address must not be running.

Syntax

From the operating system:

```
cmctl STARTUP [-c instance_name]
```

From the Oracle Connection Manager Control utility:

CMCTL> STARTUP



Usage Notes

Before issuing this command, you must use the ADMINISTER command to select an instance to start.

Issuing this command starts all instance components, which are the listener, CMADMIN, and the gateway processes. The command fails if any one of these components is already running.

The utility may prompt for a password if Oracle Connection Manager was installed with secure installation option.

Example

2.3.35 SUSPEND GATEWAY

Purpose

To specify which gateway processes will no longer accept new client connections.

Prerequisites

Oracle Connection Manager must be running.

Syntax

From the operating system:

```
cmctl SUSPEND GATEWAY [gateway process id] [-c instance name]
```

From the Oracle Connection Manager Control utility:

```
CMCTL> SUSPEND GATEWAY [gateway_process_id]
```

Arguments

gateway_process_id: The gateway process that will no longer accept new connections. Specify multiple gateway processes by putting a space between entries.

Issuing SUSPEND GATEWAY without an argument suspends all gateway processes.



Usage Notes

Use the RESUME GATEWAYS command to enable gateway processes to accept new connections.

Example

CMCTL> SUSPEND GATEWAY 1
The command completed successfully



Syntax Rules for Configuration Files

Learn how to follow the syntax configuration rules for Oracle Net Services parameters, keywords, addresses, and naming methods.

- Overview of Configuration File Syntax
 Construct your Oracle Net Services configuration files in accordance with syntax rules and standard conventions.
- Syntax Rules for Configuration Files
 Follow the structure, hierarchy, and character requirements for configuration files.
- Network Character Set for Keywords
 Use the permitted character set for keyword values and network character sets.
- Character Set for Listener and Net Service Names
 Learn how to create listener names and net service names for clients that comply with character set requirements.

3.1 Overview of Configuration File Syntax

Construct your Oracle Net Services configuration files in accordance with syntax rules and standard conventions.

The Oracle Net Services configuration files consist of parameters that include keyword-value pairs. Keyword-value pairs are surrounded by parentheses:

```
parameter=(keyword=value)
```

Some keywords have other keyword-value pairs as their values:

```
(keyword=
     (keyword1=value1)
     (keyword2=value2))
```

For example, the address portion of a local naming configuration file (tnsnames.ora) can include lines such as the following:

```
(ADDRESS=
    (PROTOCOL=tcp)
    (HOST=sales-server)
    (PORT=1521))
```

Set up your configuration files so that indentation reflects what keyword is the parent or owner of other keyword-value pairs. If you do not choose to indent your files in this way, then you

must still indent a wrapped line by at least one space, or it will be misread as a new parameter. The following syntax is acceptable:

```
(ADDRESS=(PROTOCOL=tcp)
  (HOST=sales-server) (PORT=1521))
```

The following syntax is not acceptable:

```
(ADDRESS=(PROTOCOL=tcp)
(HOST=sales-server)(PORT=1521))
```

3.2 Syntax Rules for Configuration Files

Follow the structure, hierarchy, and character requirements for configuration files.

The following rules apply to the syntax of configuration files:

- Any keyword in a configuration file that begins a parameter that includes one or more keyword-value pairs must be in the far left column of a line. If it is indented by one or more spaces, then it is interpreted as a continuation of the previous line.
- All characters must belong to the network character set.
- Keywords are not case sensitive. However, values can be case-sensitive, depending on your operating system and protocol.
- In keyword-value pairs, spaces around the equal sign (=) are optional.
- There is a hierarchy of keywords, which requires that some keywords are always followed by others. At any level of the hierarchy, keywords can be listed in any order. For example, the following entries are equally valid:

```
(ADDRESS=
    (PROTOCOL=TCP)
    (HOST=sales-server)
    (PORT=1521))

(ADDRESS=
    (PROTOCOL=tcp)
    (PORT=1521)
    (HOST=sales-server))
```

- Keywords cannot contain spaces.
- Values must not contain spaces, unless the values with spaces are enclosed within double quotation marks (") or single quotation marks (').
- If the keyword-value pair consists of a single word, or a concatenation of words on either side of the equal sign, then no parentheses are needed.
- The maximum length of a connect descriptor is 4 KB.
- You can include comments by using the number sign (#) at the beginning of a line.
 Anything following the number sign to the end of the line is considered a comment.

3.3 Network Character Set for Keywords

Use the permitted character set for keyword values and network character sets.

The network character set for keyword values consists of the following characters. Connect descriptors must be made up of single-byte characters.

```
A-Z, a-Z
0-9
() < > / \
, . : ; ' "=-
$ + * # & ! % ? @
```

Within this character set, the following symbols are reserved:

```
( ) = \ " ' #
```

Reserved symbols are used as delimiters, not as part of a keyword or a value, unless the keyword or value has quotation marks. If you have a value that contains reserved symbols, then use either single or double quotation marks to enclose the value. To include a quotation marks within a value that is surrounded by quotation marks, use different quotation marks. The backslash (\) is used as an escape character.

You can use the following characters within a connect descriptor, but not in a keyword or value:

- Space
- Tab
- Carriage return
- Newline

3.4 Character Set for Listener and Net Service Names

Learn how to create listener names and net service names for clients that comply with character set requirements.

Listener names and net service names are limited to the following character set:

```
[a...z] [A...Z] [0...9]
```

The first character in the listener name or net service name must be an alphanumeric character. In general, names up to 64 characters in length are acceptable. In addition, a database service name must match the global database name defined by the database administrator, which consists of a database name, and the database domain. Both net service names and global database names are not case-sensitive.



4

Protocol Address Configuration

Learn how to configure connections for Oracle Database instances and clients.

A network object is identified by a protocol address. When a connection is made, the client and the receiver of the request (listener or Oracle Connection Manager) are configured with identical protocol addresses. The client uses this address to send the connection request to a particular network object location, and the recipient "listens" for requests on this address, and grants a connection based on its address information matching the client information.

Protocol Addresses

The protocol address is comprised of ADDRESS and ADDRESS LIST elements.

Protocol Parameters

The listener and Oracle Connection Manager are identified by protocol addresses.

Recommended Port Numbers

Oracle recommends that you use the default port numbers for client and Oracle Connection Manager connections.

Port Number Limitations

Understand limitations for port numbers. If it is necessary to configure a listener to use a system port number (in the 1 to 1024 range), then use the procedure described here to configure the listener.

4.1 Protocol Addresses

The protocol address is comprised of ADDRESS and ADDRESS LIST elements.

ADDRESS

The ADDRESS networking parameter specifies the protocol address under the ADDRESS_LIST or DESCRIPTION parameter.

ADDRESS LIST

The ADDRESS_LIST networking parameter specifies the number of protocol addresses sharing common characteristics.

4.1.1 ADDRESS

The ADDRESS networking parameter specifies the protocol address under the ADDRESS_LIST or DESCRIPTION parameter.

Purpose

To define a protocol address.

Usage Notes

Put this parameter under an ADDRESS_LIST or DESCRIPTION parameter. A DESCRIPTION is used in a tnsnames.ora or a listener.ora file.

Example

```
(ADDRESS=
  (PROTOCOL=tcp)
  (HOST=sales-server)
  (PORT=1521))
```

Related Topics

- Protocol Parameters
 - The listener and Oracle Connection Manager are identified by protocol addresses.
- Oracle Database Global Data Services Concepts and Administration Guide

4.1.2 ADDRESS_LIST

The ADDRESS_LIST networking parameter specifies the number of protocol addresses sharing common characteristics.

Purpose

To define a list of protocol addresses that share common characteristics.

Usage Notes

This parameter is not mandatory when specifying multiple addresses.

Example

```
(ADDRESS_LIST=
  (LOAD_BALANCE=on)
  (ADDRESS=
     (PROTOCOL=tcp)
     (HOST=sales-server)
     (PORT=1521))
  (ADDRESS=
     (PROTOCOL=tcp)
     (HOST=hr-server)
     (PORT=1521)))
```

4.2 Protocol Parameters

The listener and Oracle Connection Manager are identified by protocol addresses.

The following table lists the parameters used by the Oracle protocol support:

Table 4-1 Protocol-Specific Parameters

Protocol	Parameter	Description
IPC	PROTOCOL	Specify ipc as the value.



Table 4-1 (Cont.) Protocol-Specific Parameters

Protocol	Parameter	Description
IPC	KEYPATH	On UNIX variants, IPC protocol uses the UNIX domain socket and this socket creates an internal file for client/server communication. The parameter <code>keypath</code> specifies the location where this file is created. If <code>keypath</code> is used, then use the same value of version greater than 18 on the client and listener sides.
IPC	KEY	Specify a unique name for the service. Oracle recommends using the service name or the Oracle system identifier (SID) of the service. Example:
		(PROTOCOL=ipc) (KEY=sales)
Named Pipes	PROTOCOL	Specify nmp as the value.
Named Pipes	SERVER	Specify the name of the Oracle server.
Named Pipes	PIPE	Specify the pipe name used to connect to the database server. This is the same PIPE keyword specified on server with Named Pipes. This name can be any name. Example:
		(PROTOCOL=nmp) (SERVER=sales) (PIPE=dbpipe0)
SDP	PROTOCOL	Specify sdp as the value.
SDP	HOST	Specify the host name or IP address of the computer.
SDP	PORT	Specify the listening port number. Example:
		(PROTOCOL=sdp) (HOST=sales-server) (PORT=1521) (PROTOCOL=sdp) (HOST=192.0.2.204) (PORT=1521)
TCP/IP	PROTOCOL	Specify tcp as the value.
TCP/IP	HOST	Specify the host name or IP address of the computer.
TCP/IP	PORT	Specify the listening port number. Example:
		(PROTOCOL=tcp) (HOST=sales-server) (PORT=1521) (PROTOCOL=tcp) (HOST=192.0.2.204) (PORT=1521)
TCP/IP with TLS	PROTOCOL	Specify tcps as the value.
TCP/IP with TLS	HOST	Specify the host name or IP address of the computer.
TCP/IP with TLS	PORT	Specify the listening port number. Example:
		(PROTOCOL=tcps) (HOST=sales-server) (PORT=2484) (PROTOCOL=tcps) (HOST=192.0.2.204) (PORT=2484)
Exadirect	PROTOCOL	Specify exadirect as the value.
Exadirect	HOST	Specify the IP address of the InfiBand interface.



Table 4-1 (Cont.) Protocol-Specific Parameters

Protocol	Parameter	Description
Exadirect	PORT	Specify the listening port number.
		Example:
		(PROTOCOL=exadirect) (HOST=sales-server) (PORT=2484)
		(PROTOCOL=tcps) (HOST=192.0.2.204) (PORT=1522)
Websocket	PROTOCOL	Specify \mbox{ws} as the value; use this protocol only to serve as web server backend database server.
Websocket	HOST	Specify the host name or IP address of the computer.
Websocket	PORT	Specify the listening port number.
		Example:
		(protocol=ws) (host=sales-server) (port=1524)
Secure Websocket	PROTOCOL	Specify ws as the value; use this protocol on the client side to connect to a webserver with websocket protocol support. The web server should be configured to make a websocket connection to the database listener. Wallet should be configured in sqlnet.ora.
		Use SQLNET.URI for mapping on web server.
Secure Websocket	HOST	Specify the host name or IP address of the web server with websocket support.
Secure Websocket	PORT	Specify the listening port number.
		Example:
		(protocol=wss) (host=sales-server) (port=1524)

4.3 Recommended Port Numbers

Oracle recommends that you use the default port numbers for client and Oracle Connection Manager connections.

Table 4-2 Recommended Port Numbers

Port	Description
1521	Default listening port for client connections to the listener.
	This port number can change to the officially registered port number of 2483 for TCP/IP, and 2484 for TCP/IP with TLS.
1521	Default and officially registered listening port for client connections to Oracle Connection Manager.
1830	Default and officially registered listening port for administrative commands to Oracle Connection Manager.

4.4 Port Number Limitations

Understand limitations for port numbers. If it is necessary to configure a listener to use a system port number (in the 1 to 1024 range), then use the procedure described here to configure the listener.



Oracle allows port numbers from 1 to 65535. However, the port numbers below 1024 (the **well-known ports** or **system ports**) are typically reserved. Normally, only privileged processes can listen for TCP connections on ports below 1024.

If you need to configure a listener to listen on a port number less than 1024, then complete the following procedure:



This procedure is a guideline. Your operating system can require a different procedure.

- 1. Use Oracle Net Configuration Assistant or Oracle Net Manager to configure the listener with protocol addresses and other configuration parameters.
- 2. Log in as the root user on the machine that has the listener.
- 3. Set file ownership and access permissions for the listener executable (tnslsnr) and the dependent shared libraries, so that these files can be modified only by the root user.
- 4. Starting with the root directory, ensure that the permissions of the individual directories found in the path names to these files share the same ownership and access permissions.
- 5. Start the listener as the root user.
- 6. Enter the following command at the system prompt:

```
tnslsnr listener name -user user -group group
```

In the preceding command, the following options are used:

Table 4-3 tnslsnr Utility Options

Options	Description
listener_name	Specify the name of the listener that you want to configure. If omitted, then the default name LISTENER is used.
user	Specify the user whose privileges you want the listener to use when super user (root) privileges are not needed. After performing the privileged operations, the listener gives up root privileges irreversibly.
group	Specify the group whose privileges you want the listener to use when super user (root) group privileges are not needed. After performing the privileged operations, the listener gives up root group privileges irreversibly.

During this step, the listener changes from root to the user and group privileges that you specify. All operations are done with the specified user and group privileges, except the system calls necessary to listen on configured endpoints. The listener reverts to the root user to listen on reserved addresses, such as TCP ports less than 1024.

After the listener starts listening on all of its endpoints configured in the <code>listener.ora</code> file, it switches to the specified user and group irreversibly. At that point, the listener gives up the <code>root</code> privilege that it initially had. The <code>-user</code> and <code>-group</code> command line arguments only accept user and group identifiers specified in numeric form.



For example, to run a listener called mylsnr with root privileges, and to have it use privileges of the Oracle user with the user identifier (UID) of 37555, and with OSDBA group dba membership, with a group identifier (GID) of 16, enter the following command at the operating system prompt:

tnslsnr mylsnr -user 37555 -group 16

7. After the listener starts, you can administer it with the Listener Control utility.

\mathbf{A}

Caution:

- Oracle recommends that the user under whose privileges the listener process runs is the oracle user, or a similarly privileged user with whose privileges the listener process normally runs on the operating system.
- Do not leave the listener process running as the root user. Running processes as the super user is a security vulnerability.



Parameters for the sqlnet.ora File

This chapter provides a complete listing of the sqlnet.ora file configuration parameters.

- Overview of Profile Configuration File
- sglnet.ora Profile Parameters

These are the sqlnet.ora profile configuration parameters that you use to administer database clients and servers.

- ADR Diagnostic Parameters in sqlnet.ora
 Diagnostic data for critical errors is stored in the sqlnet.ora Automatic Diagnostic Repository (ADR).
- Non-ADR Diagnostic Parameters in sqlnet.ora Files
 Learn about sqlnet.ora parameters that you use when you disable ADR.

5.1 Overview of Profile Configuration File

The sqlnet.ora file is the profile configuration file. It resides on the client machines and the database server. Profiles are stored and implemented using this file. The database server can be configured with access control parameters in the sqlnet.ora file. These parameters specify whether clients are allowed or denied access based on the protocol.

The sqlnet.ora file enables you to do the following:

- Specify the client domain to append to unqualified names
- Prioritize naming methods
- Enable logging and tracing features
- Route connections through specific processes
- Configure parameters for external naming
- Configure Oracle Advanced Security
- Use protocol-specific parameters to restrict access to the database

By default, the sqlnet.ora file is located in the <code>ORACLE_HOME/network/admin</code> directory. The sqlnet.ora file can also be stored in the directory specified by the <code>TNS_ADMIN</code> environment variable.

Note:

- The settings in the sqlnet.ora file apply to all pluggable databases (PDBs) in a multitenant container database environment.
- Oracle Net Services supports the IFILE parameter in the sqlnet.ora file, with up to three levels of nesting. The parameter is added manually to the file. The following is an example of the syntax:

```
IFILE=/tmp/listener_em.ora
IFILE=/tmp/listener_cust1.ora
IFILE=/tmp/listener_cust2.ora
```

Refer to Oracle Database Reference for additional information.

- In the read-only Oracle home mode,, the sqlnet.ora file default location is ORACLE BASE HOME/network/admin.
- In the read-only Oracle home mode, the parameters that default to <code>ORACLE_HOME</code> location change to default to <code>ORACLE_BASE</code> HOME location.

5.2 sqlnet.ora Profile Parameters

These are the sqlnet.ora profile configuration parameters that you use to administer database clients and servers.

Note:

The SQLNET.ENCRYPTION_WALLET_LOCATION sqlnet.ora parameter is deprecated in Oracle Database 19c.

The <code>SQLNET.ENCRYPTION_WALLET_LOCATION</code> parameter defines the location of the software keystores for Transparent Data Encryption (TDE). To configure the software keystore location, instead of setting <code>SQLNET.ENCRYPTION_WALLET_LOCATION</code>, Oracle recommends that you set the <code>WALLET_ROOT</code> initialization parameter, and the <code>TDE CONFIGURATION</code> dynamic initialization parameter.

These parameters are described in Oracle Database Advanced Security Guide.

- ACCEPT_MD5_CERTS
 - The sqlnet.ora profile parameter ACCEPT MD5 CERTS accepts MD5 signed certificates.
- ACCEPT_SHA1_CERTS
- ADD SSLV3 TO DEFAULT

The sqlnet.ora profile parameter ADD_SSLV3_TO_DEFAULT sets the Transport Layer Security (TLS) versions that your server accepts.

BEQUEATH DETACH

Use the sqlnet.ora parameter to enable and disable signal handling on Linux and UNIX systems.



DEFAULT SDU SIZE

Use the sqlnet.ora profile parameter to specify the session data unit size (SDU) for connections.

• DISABLE INTERRUPT

Use the sqlnet.ora profile parameter DISABLE_INTERRUPT to disable Oracle Net handling of a SIGINIT signal in client applications.

DISABLE OOB

DISABLE_OOB is a networking parameter of the sqlnet.ora file and is used to enable or disable Oracle Net to send or receive out-of-band break messages using urgent data provided by the underlying protocol.

DISABLE OOB AUTO

The DISABLE_OOB_AUTO networking parameter of the sqlnet.ora file checks the server path for out-of-band break messages support at the connection time.

- EXADIRECT FLOW CONTROL
- EXADIRECT RECVPOLL
- IPC.KEYPATH
- NAMES.DEFAULT DOMAIN

NAMES.DIRECTORY PATH

Use the sqlnet parameter NAMES.DIRECTORY_PATH to specify the order of the naming methods for client name resolution lookups.

- NAMES.LDAP AUTHENTICATE BIND
- NAMES.LDAP_CONN_TIMEOUT
- NAMES.LDAP_PERSISTENT_SESSION
- NAMES.NIS.META MAP

OCI COMPARTMENT

Use the <code>OCI_COMPARTMENT</code> parameter to specify Oracle Cloud Identifier (OCID) of the compartment that holds database instances for client connections.

OCI DATABASE

Use the $OCI_DATABASE$ parameter to specify Oracle Cloud Identifier (OCID) of the database that you want to access for the client connection.

OCI IAM URL

Use the OCI_IAM_URL parameter to specify an endpoint URL that the database client must connect with to get the database token for authenticating Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM) users on OCI Database as a Service (DBaaS).

OCI TENANCY

Use the $OCI_{TENANCY}$ parameter to specify Oracle Cloud Identifier (OCID) of the user's tenancy.

- PASSWORD_AUTH
- RECV BUF SIZE

Use the sqlnet.ora parameter RECV_BUF_SIZE to specify buffer space limit for session receive operations.

- SDP.PF_INET_SDP
- SEC USER AUDIT ACTION BANNER
- SEC USER UNAUTHORIZED ACCESS BANNER

SEND BUF SIZE

Use the sqlnet parameter $SEND_BUF_SIZE$ to specify the buffer space limit for session send operations.

SQLNET.ALLOW WEAK CRYPTO

Use the sqlnet.ora compatibility parameter $SQLNET.ALLOW_WEAK_CRYPTO$ to configure your client-side network connection by reviewing the specified encryption and crypto-checksum algorithms.

SQLNET.ALLOW WEAK CRYPTO CLIENTS

Use the sqlnet.ora compatibility parameter $SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS$ to configure your server-side network connection by reviewing the specified encryption and crypto-checksum algorithms.

SQLNET.ALLOWED_LOGON_VERSION_CLIENT

Use the sqlnet parameter SQLNET.ALLOWED_LOGON_VERSION_CLIENT to define minimum authentication protocols that servers acting as clients to other servers can use for connecting to Oracle Database instances.

SQLNET.ALLOWED LOGON VERSION SERVER

Use the sqlnet.ora parameter $SQLNET.ALLOWED_LOGON_VERSION_SERVER$ to set the minimum authentication protocol that is permitted when connecting to Oracle Database instances.

SQLNET.AUTHENTICATION SERVICES

Use the sqlnet.ora parameter SQLNET.AUTHENTICATION_SERVICES to enable one or more authentication services.

- SQLNET.CLIENT_REGISTRATION
- SQLNET.CLOUD_USER
- SOLNET.COMPRESSION
- SQLNET.COMPRESSION ACCELERATION
- SQLNET.COMPRESSION LEVELS
- SQLNET.COMPRESSION_THRESHOLD
- SQLNET.CRYPTO_CHECKSUM_CLIENT

Use the sqlnet.ora parameter SQLNET.CRYPTO_CHECKSUM_CLIENT to specify the desired data integrity behavior when this client or server acting as a client connects to a server.

SQLNET.CRYPTO CHECKSUM SERVER

Use the sqlnet.ora parameter SQLNET.CRYPTO_CHECKSUM_SERVER to specify the data integrity behavior when a client or another server acting as a client connects to this server.

SQLNET.CRYPTO CHECKSUM TYPES CLIENT

Use the sqlnet.ora parameter SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT to specify a list of data integrity algorithms that this client or server acting as a client uses.

SQLNET.CRYPTO CHECKSUM TYPES SERVER

Use the sqlnet.ora parameter $SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER$ to specify the data integrity algorithms that this server or client to another server uses, in order of intended use.

- SQLNET.DBFW PUBLIC KEY
- SQLNET.DOWN HOSTS TIMEOUT

SQLNET.ENCRYPTION CLIENT

Use the sqlnet.ora parameter SQLNET.ENCRYPTION_CLIENT to set the encryption behavior when this client or server acting as a client connects to a server.

SQLNET.ENCRYPTION SERVER

The sqlnet.ora parameter SQLNET.ENCRYPTION_SERVER specifies the encryption behavior when a client or a server acting as a client connects to this server.

SQLNET.ENCRYPTION TYPES CLIENT

Use the sqlnet.ora parameter $SQLNET.ENCRYPTION_TYPES_CLIENT$ to list encryption algorithms for clients to use.

SQLNET.ENCRYPTION TYPES SERVER

Use the sqlnet.ora parameter SQLNET.ENCRYPTION_TYPES_SERVER to list the encryption algorithms for the database to use.

SOLNET.EXPIRE TIME

Use the sqlnet.ora parameter SQLNET.EXPIRE_TIME to specify how often, in minutes, to verify that client and server connections are active.

SQLNET.IGNORE AND ENCRYPTION FOR TCPS

The SQLNET.IGNORE_ANO_ENCRYPTION_FOR_TCPS parameter is used on the server-side to ignore the value set in SQLNET.ENCRYPTION_SERVER for TCPS connections (effectively disabling ANO encryption on the TCPS listener).

- SQLNET.INBOUND_CONNECT_TIMEOUT
- SQLNET.FALLBACK AUTHENTICATION

SQLNET.KERBEROS5 CC NAME

Use the sqlnet.ora parameter $SQLNET.KERBEROS5_CC_NAME$ to specify the complete path name to the Kerberos credentials cache file.

- SQLNET.KERBEROS5 CLOCKSKEW
- SQLNET.KERBEROS5 CONF
- SOLNET.KERBEROS5 CONF LOCATION
- SQLNET.KERBEROS5 KEYTAB
- SQLNET.KERBEROS5 REALMS
- SQLNET.KERBEROS5 REPLAY CACHE

SQLNET.OUTBOUND_CONNECT_TIMEOUT

Use the sqlnet.ora parameter SQLNET.OUTBOUND_CONNECT_TIMEOUT to specify the amount of time, in milliseconds, seconds, or minutes, in which clients must establish Oracle Net connections to database instances.

- SQLNET.RADIUS ALTERNATE
- SQLNET.RADIUS ALTERNATE PORT
- SQLNET.RADIUS_ALTERNATE_RETRIES
- SQLNET.RADIUS ALTERNATE TIMEOUT

Use the sqlnet.ora parameter SQLNET.RADIUS_ALTERNATE_TIMEOUT to set the time for an alternate RADIUS server to wait for a response.

- SQLNET.RADIUS_AUTHENTICATION
- SQLNET.RADIUS AUTHENTICATION INTERFACE
- SQLNET.RADIUS AUTHENTICATION PORT
- SQLNET.RADIUS AUTHENTICATION RETRIES
- SQLNET.RADIUS AUTHENTICATION TIMEOUT



SQLNET.RADIUS CHALLENGE KEYWORD

Use the sqlnet.ora parameter SQLNET.RADIUS_CHALLENGE_KEYWORD to set the keyword for requesting a challenge from the RADIUS server.

SQLNET.RADIUS CHALLENGE RESPONSE

Use the sqlnet.ora parameter SQLNET.RADIUS_CHALLENGE_RESPONSE to enable or disable challenge responses.

SOLNET.RADIUS CLASSPATH

Use the sqlnet.ora parameter SQLNET.RADIUS_CLASSPATH to set the path for Java classes and JDK Java libraries.

SQLNET.RADIUS_SECRET

SQLNET.RADIUS_SEND_ACCOUNTING

SQLNET.RECV_TIMEOUT

Use the sqlnet.ora parameter SQLNET.RECV_TIMEOUT to specify the duration of time that a database client or server should wait for data from a peer after establishing a connection.

SOLNET.SEND TIMEOUT

Use the sqlnet.ora parameter SQLNET.SEND_TIMEOUT to specify the duration of time for a database server to complete a send operation to clients after establishing a connection.

SOLNET.URI

SQLNET.URI networking parameter of the sqlnet.ora file specifies a database client URI mapping on the web server.

SQLNET.USE HTTPS PROXY

SQLNET.WALLET OVERRIDE

Use the sqlnet.ora parameter $SQLNET.WALLET_OVERRIDE$ to determine whether a client should override strong authentication credentials with the password credential from the stored wallet.

SSL CERT REVOCATION

Use the sqlnet.ora parameter SSL_CERT_REVOCATION to configure revocation checks for certificates.

SSL CRL FILE

Use the sqlnet.ora parameter SSL_CRL_FILE to specify the name of the file in which you assemble the certificate revocation list (CRL) for client authentication.

SSL CRL PATH

Use the sqlnet.ora parameter SSL_CRL_PATH to specify the destination directory of the certificate revocation list (CRL) for client authentication.

SSL CIPHER SUITES

Use the SSL_CIPHER_SUITES parameter to control the combination of authentication, encryption, and data integrity algorithms used by Transport Layer Security (TLS).

SSL CLIENT AUTHENTICATION

Use the SSL_CLIENT_AUTHENTICATION parameter to specify whether a client is authenticated using Transport Layer Security (TLS).

SSL DISABLE WEAK EC CURVES

Use the SSL_DISABLE_WEAK_EC_CURVES parameter to disable the use of weak Elliptic Curve Cryptography (ECC) curves.

SSL EXTENDED KEY USAGE

SSL SERVER DN MATCH

Use the $SSL_SERVER_DN_MATCH$ parameter to enforce server-side certificate validation through distinguished name (DN) matching.

SSL VERSION

Use the SSL_VERSION parameter to define valid Transport Layer Security (TLS) versions to be used for connections.

- TCP.CONNECT TIMEOUT
- TCP.EXCLUDED NODES
- TCP.INVITED NODES
- TCP.NODELAY
- TCP.QUEUESIZE
- TCP.VALIDNODE CHECKING
- TNSPING.TRACE_DIRECTORY
- TNSPING.TRACE_LEVEL
- TOKEN AUTH

Use the TOKEN_AUTH parameter to configure token-based authentication for Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM) or Microsoft Azure users of Microsoft Entra ID (previously called Microsoft Azure Active Directory).

TOKEN LOCATION

Use the <code>TOKEN_LOCATION</code> parameter to specify the directory location where token file is stored for token-based authentication.

- USE_CMAN
- USE DEDICATED SERVER
- WALLET_LOCATION

Use the sqlnet.ora parameter WALLET LOCATION to specify the location of Oracle wallets.

5.2.1 ACCEPT_MD5_CERTS

The sqlnet.ora profile parameter ACCEPT MD5 CERTS accepts MD5 signed certificates.

Purpose

To enable sqlnet to accept MD5 signed certificates. In addition to sqlnet.ora, you must also set this parameter in listener.ora.

Default

FALSE

Values

- TRUE to accept MD5 signed certificates
- FALSE to not accept MD5 signed certficates

5.2.2 ACCEPT_SHA1_CERTS

Purpose

To not accept SHA1 signed certificates, in addition to sqlnet.ora, this parameter must also be set in listener.ora.



Default

TRUE

Values

- TRUE to accept SHA1 signed certificates
- FALSE to not accept SHA1 signed certificates

5.2.3 ADD SSLV3 TO DEFAULT

The sqlnet.ora profile parameter ADD_SSLV3_TO_DEFAULT sets the Transport Layer Security (TLS) versions that your server accepts.

Purpose

To set the TLS versions that your server accepts.

Usage Notes

To use SSL_VERSION=3.0 in your SSL_VERSION default list, set the value to TRUE. In addition to setting this parameter in sqlnet.ora, you must also set this parameter in listener.ora.

Default

FALSE

Values

- If set to TRUE and SSL_VERSION is not specified or is set to "undetermined", then SSL VERSION includes versions 1.2, 1.1, 1.0, and 3.0.
- If set to FALSE and SSL_VERSION is not specified or is set to "undetermined", then SSL VERSION includes versions 1.2, 1.1, and 1.0

5.2.4 BEQUEATH_DETACH

Use the sqlnet.ora parameter to enable and disable signal handling on Linux and UNIX systems.

Purpose

To turn signal handling on or off for Linux and UNIX systems

Default

no

Values

- yes to turn signal handling off
- no to leave signal handling on

Example

BEQUEATH_DETACH=yes



5.2.5 DEFAULT_SDU_SIZE

Use the sqlnet.ora profile parameter to specify the session data unit size (SDU) for connections.

Purpose

To specify the SDU size, in bytes, for connections.

Usage Notes

Oracle recommends setting this parameter in both the client-side and server-side sqlnet.ora files to ensure that the same SDU size is used throughout a connection. When the configured values of client and database server do not match for a session, the lower of the two values is used.

You can override this parameter for a particular client connection by specifying the SDU parameter in the connect descriptor for a client.

Default

8192 bytes (8 KB)

Value

512 to 2097152 bytes

Example 5-1 Example

DEFAULT SDU SIZE=4096

5.2.6 DISABLE_INTERRUPT

Use the sqlnet.ora profile parameter DISABLE_INTERRUPT to disable Oracle Net handling of a SIGINIT signal in client applications.

Purpose

To disable Oracle Net handling of a SIGINIT signal in client applications.

Usage Notes

Oracle Net installs a signal handler to catch a SIGINT signal. By default, the action on receipt of a SIGINIT signal is to cancel the current operation. If you set this parameter to TRUE, then you can override the default behavior and ignore Oracle Net handling of SIGINT signals.

For details on installing and uninstalling your own signal handlers in addition to Oracle Net, see Oracle Database Administrator's Reference for Linux and UNIX-Based Operating Systems.

Default

FALSE

Example

DISABLE INTERRUPT=TRUE



5.2.7 DISABLE_OOB

DISABLE_OOB is a networking parameter of the sqlnet.ora file and is used to enable or disable Oracle Net to send or receive out-of-band break messages using urgent data provided by the underlying protocol.

Purpose

To enable or disable Oracle Net to send or receive out-of-band break messages using urgent data provided by the underlying protocol.

Usage Notes

If turned off, then the parameter enables Oracle Net to send and receive break messages. If turned on, then the parameter disables the ability to send and receive break messages. Once enabled, this feature applies to all protocols used by this client.

Default

off

Example 5-2 Example

DISABLE OOB=on

5.2.8 DISABLE_OOB_AUTO

The <code>DISABLE_OOB_AUTO</code> networking parameter of the <code>sqlnet.ora</code> file checks the server path for out-of-band break messages support at the connection time.

Purpose

Disable automatic out-of-band (OOB) support checks the server path at connection time.

Usage Notes

By default, the client checks if the server path supports out-of-band break messages or not at the connection time. If this parameter is set to \mathtt{TRUE} , then the client does not perform this check at the connection time.

Default

FALSE

Example 5-3 Example

DISABLE OOB AUTO = TRUE

5.2.9 EXADIRECT_FLOW_CONTROL

Purpose

To enable or disable Exadirect flow control.

Usage Notes

If turned on, the parameter enables Oracle Net to broadcast available receive window to the sender. The sender limits the sends based on the receiver broadcast window.



Default

off

Example

EXADIRECT FLOW CONTROL=on

5.2.10 EXADIRECT RECVPOLL

Purpose

To specify the time that a receiver polls for incoming data.

Usage Notes

The parameter can be set to a fixed value or AUTO for auto tuning of the polling value.

Default

0

Example

```
EXADIRECT_RECVPOLL = 10

EXADIRECT RECVPOLL = AUTO
```

5.2.11 IPC.KEYPATH

Purpose

To specify the destination directory where the internal file is created for UNIX domain sockets.

Usage Notes

This parameter applies only to Oracle Net's usage of UNIX domain socket and does not apply to other usages of UNIX domain sockets in the database, such as clusterware. If keypath is used, then the same value should be used on both the client and the listener sides with version greater than 18.

Default

The directory path is either /var/tmp/.oracle for Oracle Linux, Oracle Solaris or /tmp/.oracle for other UNIX variants.

Example

ipc.keypath=/home/oracleuser.

5.2.12 NAMES.DEFAULT_DOMAIN

Purpose

To set the domain from which the client most often looks up names resolution requests.



Usage Notes

When this parameter is set, the default domain name is automatically appended to any unqualified net service name or service name.

For example, if the default domain is set to us.example.com, then the connect string CONNECT scott@sales gets searched as sales.us.example.com. If the connect string includes the domain extension, such as CONNECT scott@sales.us.example.com, then the domain is not appended to the string.

Default

None

Example

NAMES.DEFAULT_DOMAIN=example.com

5.2.13 NAMES.DIRECTORY_PATH

Use the sqlnet parameter NAMES.DIRECTORY_PATH to specify the order of the naming methods for client name resolution lookups.

Purpose

To specify the order of the naming methods for client name resolution lookups.

Default

NAMES.DIRECTORY PATH=(tnsnames, ezconnect, ldap)

Values

The following table shows the NAMES.DIRECTORY_PATH values for the naming methods.

Naming Method Value	Description
tnsnames (local naming method)	Set to resolve a network service name through the tnsnames.ora file on the client.
ldap (directory naming method)	Set to resolve a database service name, net service name, or network service alias through a directory server.
ezconnect or hostname (Easy Connect naming method)	Select to enable clients to use a TCP/IP connect identifier that consists of a host name and optional port and service name.
nis (external naming method)	Set to resolve service information through an existing Network Information Service (NIS).

Example

NAMES.DIRECTORY_PATH=(tnsnames)



5.2.14 NAMES.LDAP_AUTHENTICATE_BIND

Purpose

To specify whether the LDAP naming adapter should attempt to authenticate using a specified wallet when it connects to the LDAP directory to resolve the name in the connect string.

Usage Notes

The parameter value is Boolean.

If the parameter is set to TRUE, then the LDAP connection is authenticated using a wallet whose location must be specified in the WALLET LOCATION parameter.

If the parameter is set to FALSE, then the LDAP connection is established using an anonymous bind.

Default

false

Example

NAMES.LDAP AUTHENTICATE BIND=true

5.2.15 NAMES.LDAP_CONN_TIMEOUT

Purpose

To specify number of seconds for a non-blocking connect timeout to the LDAP server.

Usage Notes

The parameter value -1 is for infinite timeout.

Default

15 seconds

Values

Values are in seconds. The range is -1 to the number of seconds acceptable for your environment. There is no upper limit.

Example

 $names.ldap_conn_timeout = -1$

5.2.16 NAMES.LDAP_PERSISTENT_SESSION

Purpose

To specify whether the LDAP naming adapter should leave the session with the LDAP server open after name lookup is complete.



Usage Notes

The parameter value is Boolean.

If the parameter is set to TRUE, then the connection to the LDAP server is left open after the name lookup is complete. The connection will effectively stay open for the duration of the process. If the connection is lost, then it is re-established as needed.

If the parameter is set to FALSE, then the LDAP connection is terminated as soon as the name lookup completes. Every subsequent lookup opens the connection, performs the lookup, and closes the connection. This option prevents the LDAP server from having a large number of clients connected to it at any one time.

Default

false

Example

NAMES.LDAP PERSISTENT SESSION=true

5.2.17 NAMES.NIS.META MAP

Purpose

To specify the map file to be used to map Network Information Service (NIS) attributes to an NIS mapname.

Default

sqlnet.maps

Example

NAMES.NIS.META_MAP=sqlnet.maps

5.2.18 OCI_COMPARTMENT

Use the OCI_COMPARTMENT parameter to specify Oracle Cloud Identifier (OCID) of the compartment that holds database instances for client connections.

Purpose

To define the scope of your database token request. This value instructs the database client to initiate a token request to databases within the specified compartment only.

Usage Notes

You can use this parameter along with the PASSWORD_AUTH, OCI_IAM_URL, and OCI_TENANCY parameters while configuring token-based authentication for Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM) users on OCI Database as a Service (DBaaS). You can also use the optional OCI_DATABASE parameter to specify a database instance within the compartment for your connection.

With this configuration, the database client can only request an IAM database token using the IAM user name and IAM database password. The client cannot request an IAM database token for an API-key, delegation token, security token, resource principal, service principal, or instance principal.



The OCI_COMPARTMENT parameter is optional if OCI_DATABASE is not set. If you choose to set OCI_DATABASE, then you must also set OCI_COMPARTMENT so that your token request is for the specified database in that compartment.

If you do not set both <code>OCI_COMPARTMENT</code> and <code>OCI_DATABASE</code>, then the entire tenancy is the scope of your token request.

Use this parameter under the SECURITY section of the tnsnames.ora file, sqlnet.ora file, or directly as part of the command-line connect string. The parameter value specified in the connect string takes precedence over the other specified values.

Default

None

Value

OCID for the IAM compartment to allow access for the database token. You can get the OCID value for your compartment from the Compartments information page in the OCI console.

The compartment OCID uses this syntax:

```
OCI COMPARTMENT=compartment OCID
```

For details on the syntax options, see Oracle Cloud IDs (OCIDs).

Examples

```
In the tnsnames.ora file:
```

In the sqlnet.ora file:

```
SSL_SERVER_DN_MATCH=TRUE
PASSWORD_AUTH=OCI_TOKEN
OCI_IAM_URL=https://auth.us-region-1.example.com/v1/actions/
generateScopedAccessBearerToken
OCI_TENANCY=ocid1.tenancy..12345
OCI_COMPARTMENT=ocid1.compartment..12345
OCI_DATABASE=ocid1.autonomousdatabase.oc1.12345
```

Related Topics

Oracle Database Security Guide



PASSWORD AUTH

5.2.19 OCI_DATABASE

Use the <code>OCI_DATABASE</code> parameter to specify Oracle Cloud Identifier (OCID) of the database that you want to access for the client connection.

Purpose

To define the scope of your database token request. The database OCID value instructs the database client to initiate a token request to the specified database within your compartment.

Usage Notes

This parameter is optional. You can use this parameter along with the PASSWORD_AUTH, OCI_IAM_URL, OCI_TENANCY, and OCI_COMPARTMENT parameters while configuring token-based authentication for Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM) users on OCI Database as a Service (DBaaS).

With this configuration, the database client can only request an IAM database token using the IAM user name and IAM database password. The client cannot request an IAM database token for an API-key, delegation token, security token, resource principal, service principal, or instance principal.

The OCI_DATABASE value limits your token request to the specified database only. If you set OCI_DATABASE, then you must also set OCI_COMPARTMENT so that your token request is for the specified database in that compartment.

Use this parameter under the SECURITY section of the tnsnames.ora file, sqlnet.ora file, or directly as part of the command-line connect string. The parameter value specified in the connect string takes precedence over the other specified values.

Default

None

Value

OCID of the database that you want to access for the client connection. You can get the OCID value for your database from the Database details page in the OCI console.

The database OCID uses this syntax:

```
OCI DATABASE=database OCID
```

For details on the syntax options, see Oracle Cloud IDs (OCIDs).

Examples

In the tnsnames.ora file:



Related Topics

- Oracle Database Security Guide
- PASSWORD_AUTH

5.2.20 OCI IAM URL

Use the <code>OCI_IAM_URL</code> parameter to specify an endpoint URL that the database client must connect with to get the database token for authenticating Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM) users on OCI Database as a Service (DBaaS).

Purpose

To specify the IAM URL for your REST API requests. The database client connects to this URL to retrieve the database token from IAM.

Usage Notes

You set the <code>OCI_IAM_URL</code> parameter along with the <code>PASSWORD_AUTH</code> and <code>OCI_TENANCY</code> parameters while configuring IAM token-based authentication (using the IAM user name and IAM database password to retrieve the database token). These parameters are mandatory.

With this configuration, the database client can only request an IAM database token using the IAM user name and IAM database password. The client cannot request an IAM database token for an API-key, delegation token, security token, resource principal, service principal, or instance principal.

You can also set the optional <code>OCI_COMPARTMENT</code> and <code>OCI_DATABASE</code> parameters to specify the scope of your token request.

Use this parameter under the SECURITY section of the tnsnames.ora file, sqlnet.ora file, or directly as part of the command-line connect string. The parameter value specified in the connect string takes precedence over the other specified values.

Default

None



Value

OCI IAM endpoint URL that the database client must connect with to get the database token. This URL is specific to your region and uses this syntax:

```
<authentication_regional_endpoint>/v1/actions/generateScopedAccessBearerToken
```

You can derive this value by replacing <authentication_regional_endpoint> with the API endpoint URL for your region. To obtain the appropriate API endpoint URL, see Identity and Access Management Data Plane API.

For example, if you want to use the URL as https://auth.us-region-1.example.com, then your OCI_IAM_URL value is:

```
https://auth.us-region-1.example.com/v1/actions/generateScopedAccessBearerToken
```

Examples

```
In the tnsnames.ora file:
```

```
net service name=
  (DESCRIPTION=
     (ADDRESS=(PROTOCOL=tcps) (HOST=salesserver1) (PORT=1522))
     (SECURITY=
        (SSL SERVER DN MATCH=TRUE)
        (SSL SERVER CERT DN="C=US,O=example,CN=OracleContext")
        (PASSWORD AUTH=OCI TOKEN)
        (OCI IAM URL=https://auth.us-region-1.example.com/v1/actions/
generateScopedAccessBearerToken)
        (OCI TENANCY=ocid1.tenancy..12345))
     (CONNECT DATA=(SERVICE NAME=sales.us.example.com))
In the sqlnet.ora file:
SSL SERVER DN MATCH=TRUE
PASSWORD AUTH=OCI TOKEN
OCI IAM URL=https://auth.us-region-1.example.com/v1/actions/
generateScopedAccessBearerToken
OCI TENANCY=ocid1.tenancy..12345
```

In these examples, the optional <code>OCI_COMPARTMENT</code> and <code>OCI_DATABASE</code> parameters are not specified and thus the entire tenancy is set as the scope of the token request.

Related Topics

- Oracle Database Security Guide
- PASSWORD AUTH

5.2.21 OCI_TENANCY

Use the OCI_TENANCY parameter to specify Oracle Cloud Identifier (OCID) of the user's tenancy.

Purpose

To specify OCID of the user's tenancy (root compartment).

Usage Notes

You set this parameter along with the mandatory PASSWORD_AUTH and OCI_IAM_URL parameters while configuring token-based authentication for Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM) users on OCI Database as a Service (DBaaS).

With this configuration, the database client can only request an IAM database token using the IAM user name and IAM database password. The client cannot request an IAM database token for an API-key, delegation token, security token, resource principal, service principal, or instance principal.

You can also set the optional OCI_COMPARTMENT and OCI_DATABASE parameters to specify the scope of your token request. If you do not set the OCI_COMPARTMENT and OCI_DATABASE parameter values, then the entire tenancy is the scope of your token request.

Use this parameter under the SECURITY section of the tnsnames.ora file, sqlnet.ora file, or directly as part of the command-line connect string. The parameter value specified in the connect string takes precedence over the other specified values.

Default

None

Value

OCID of the user's tenancy. You can get the OCID value for your tenancy from the Tenancy information page in the OCI console.

The tenancy OCID uses this syntax:

```
OCI TENANCY=tenancy OCID
```

For details on the syntax options, see Oracle Cloud IDs (OCIDs).

Examples

In the tnsnames.ora file:

```
(CONNECT_DATA=(SERVICE_NAME=sales.us.example.com))
)
In the sqlnet.ora file:

SSL_SERVER_DN_MATCH=TRUE
PASSWORD_AUTH=OCI_TOKEN
OCI_IAM_URL=https://auth.us-region-1.example.com/v1/actions/generateScopedAccessBearerToken
OCI_TENANCY=ocid1.tenancy..12345
```

In these examples, the optional <code>OCI_COMPARTMENT</code> and <code>OCI_DATABASE</code> parameters are not specified and thus the entire tenancy is set as the scope of the token request.

Related Topics

- Oracle Database Security Guide
- PASSWORD AUTH

5.2.22 PASSWORD_AUTH

Use the PASSWORD_AUTH parameter to configure an authentication method for Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM) users on OCI Database as a Service (DBaaS). With this setting, client connections use the IAM user name and IAM database password for logging in users to the database.

Purpose

To configure either IAM database password verifier authentication or IAM token-based authentication, using the IAM user name and IAM database password for the access.

For password verifier authentication, the database server retrieves an IAM database password verifier from IAM. For token-based authentication, the database client requests a database token (db-token) from IAM.

Usage Notes

- Use this parameter under the SECURITY section of the tnsnames.ora file, sqlnet.ora file, or directly as part of the command-line connect string. The parameter value specified in the connect string takes precedence over the other specified values.
- This setting instructs the database client to either use the existing password login process
 with the database server (password verifier authentication) or to get a token with the IAM
 user name and IAM database password (token-based authentication). This IAM database
 password is different from the OCI console password. An IAM user can set this password
 from the OCI console.

See Create an OCI IAM password to use for Autonomous Databases User Authentication and Authorization.

• By default, this parameter is set to PASSWORD_VERIFIER. The PASSWORD_AUTH=PASSWORD_VERIFIER setting configures IAM database password verifier authentication. The database server retrieves an IAM database password verifier (an encrypted hash of password) from IAM to authenticate users.

When an IAM user logs in with the IAM user name and IAM database password using <code>@connect_identifier</code>, the <code>PASSWORD_AUTH=PASSWORD_VERIFIER</code> setting along with

<code>@connect_identifier</code> instructs the database client to follow the existing user name and password login process with the database server.

You can use the PASSWORD_AUTH parameter to override the tnsnames.ora or sqlnet.ora setting by specifying a different value in the connect string.

• To configure IAM token-based authentication with the IAM user name and IAM database password, set PASSWORD_AUTH=OCI_TOKEN. The database client requests a database token (db-token) from IAM for the user to access the database.

This db-token obtained by the client is a bearer token with an expiration time and scope, and does not come with a private key. These tokens are transmitted over secure channels. You must use only the TCP/IP with Transport Layer Security (TLS) protocol, otherwise an error message appears indicating that non-TLS connections are disallowed.

When an IAM user logs in with the IAM user name and IAM database password using / @connect_identifier, the PASSWORD_AUTH=OCI_TOKEN setting along with / @connect_identifier instructs the database client to get the token directly from an OCI IAM endpoint using a REST API request. If the IAM user is mapped to a database schema (exclusively or shared), then the login is completed.

For the database client to retrieve the token from IAM, you must set additional parameters so that the database client can find the IAM endpoint along with additional meta-data. The additional parameters are <code>OCI_IAM_URL</code> and <code>OCI_TENANCY</code> along with the optional <code>OCI_COMPARTMENT</code> and <code>OCI_DATABASE</code>. These values enable the database client to make appropriate calls to the specified endpoint.

The <code>OCI_IAM_URL</code> parameter specifies the API endpoint URL that the database client must connect with. The <code>OCI_TENANCY</code> parameter specifies the OCID (Oracle Cloud Identifier) of the user's tenancy. The optional <code>OCI_COMPARTMENT</code> and <code>OCI_DATABASE</code> parameters limit the scope of your request.

This authentication method is more secure than using a password verifier because a password verifier is considered sensitive. Also, only the database client can retrieve the database token. Applications or tools cannot pass these types of tokens through the database client API.

Note:

You can also use other IAM user credentials (such as API-key, security token, resource principal, service principal, instance principal, or delegation token) to get the db-token. This db-token is a proof-of-possession (PoP) token. In this case, you use a different parameter setting (TOKEN AUTH=OCI TOKEN).

Unlike the IAM database password that can only be used by the database client to retrieve the token, these credentials require an application or tool to retrieve the token. See TOKEN AUTH.

Default

PASSWORD VERIFIER



Values and Examples

Value **Example**

For IAM database password verifier authentication: In the tnsnames.ora file:

PASSWORD AUTH=PASSWORD VERIFIER

Note: Use of IAM user name and IAM database password with the IAM database password verifier is the default configuration, and you do not need to set any additional parameters for the client.

However, if PASSWORD AUTH is set to OCI TOKEN in the client-side sqlnet.ora file, then the client tries to connect with OCI IAM to retrieve a database token using the IAM user name and IAM database password. In this case, you can override this setting for a particular connection using PASSWORD_AUTH=PASSWORD_VERIFIER.

```
net service name=
  (DESCRIPTION =
     (ADDRESS=(PROTOCOL=tcps)
(HOST=sales-svr) (PORT=1521))
     (SECURITY=
        (SSL SERVER DN MATCH=TRUE)
(SSL SERVER CERT DN="C=US,O=example,C
N=OracleContext")
(PASSWORD AUTH=PASSWORD VERIFIER))
(CONNECT DATA=(SERVICE NAME=sales.us.
example.com))
In the sqlnet.ora file:
```

PASSWORD AUTH=PASSWORD VERIFIER



Value

For IAM token-based authentication with the IAM user name and IAM database password:

PASSWORD AUTH=OCI TOKEN

Note: You must configure the TCPS protocol (PROTOCOL=tcps) and set the SSL_SERVER_DN_MATCH parameter to TRUE for token-based authentication.

Example

In the tnsnames.ora file:

```
net service name=
  (DESCRIPTION=
     (ADDRESS=(PROTOCOL=tcps)
(HOST=salesserver1) (PORT=1522))
     (SECURITY=
        (SSL_SERVER_DN_MATCH=TRUE)
(SSL SERVER CERT DN="C=US,O=example,C
N=OracleContext")
        (PASSWORD AUTH=OCI_TOKEN)
        (OCI IAM URL=https://auth.us-
region-1.example.com/v1/actions/
generateScopedAccessBearerToken)
(OCI TENANCY=ocid1.tenancy..12345))
(CONNECT DATA=(SERVICE NAME=sales.us.
example.com))
  )
In the sqlnet.ora file:
SSL SERVER DN MATCH=TRUE
```

In these examples, the optional OCI_COMPARTMENT and OCI_DATABASE parameters are not specified and thus the entire tenancy is set as the scope of the token request.

PASSWORD_AUTH=OCI_TOKEN
OCI_IAM_URL=https://auth.usregion-1.example.com/v1/actions/
generateScopedAccessBearerToken
OCI_TENANCY=ocid1.tenancy..12345

Related Topics

- Oracle Database Security Guide
- OCI IAM URL

Use the <code>OCI_IAM_URL</code> parameter to specify an endpoint URL that the database client must connect with to get the database token for authenticating Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM) users on OCI Database as a Service (DBaaS).

OCI TENANCY

Use the OCI_TENANCY parameter to specify Oracle Cloud Identifier (OCID) of the user's tenancy.

OCI COMPARTMENT

Use the <code>OCI_COMPARTMENT</code> parameter to specify Oracle Cloud Identifier (OCID) of the compartment that holds database instances for client connections.

OCI DATABASE

Use the $OCI_DATABASE$ parameter to specify Oracle Cloud Identifier (OCID) of the database that you want to access for the client connection.

5.2.23 RECV_BUF_SIZE

Use the sqlnet.ora parameter RECV_BUF_SIZE to specify buffer space limit for session receive operations.

Purpose

To specify the buffer space limit for receive operations of sessions.

Usage Notes

You can override this parameter for a particular client connection by specifying the RECV_BUF_SIZE parameter in the connect descriptor for a client.

This parameter is supported by the TCP/IP, TCP/IP with TLS, and SDP protocols.



Additional protocols might support this parameter on certain operating systems. Refer to the operating system-specific documentation for additional information about additional protocols that support this parameter.

Default

The default value for this parameter is operating system specific. The default for Linux 2.6 operating system is 87380 bytes.

Example

RECV_BUF_SIZE=11784

Related Topics

Oracle Database Net Services Administrator's Guide

5.2.24 SDP.PF_INET_SDP

Purpose

To specify the protocol family or address family constant for the SDP protocol on your system.

Default

27

Values

Any positive integer

Example

SDP.PF_INET_SDP=30



5.2.25 SEC_USER_AUDIT_ACTION_BANNER

Purpose

To specify a text file containing the banner contents that warn the user about possible user action auditing.

Usage Notes

The complete path of the text file must be specified in the sqlnet.ora file on the server. Oracle Call Interface (OCI) applications can make use of OCI features to retrieve this banner and display it to the user.

Default

None

Values

Name of the file for which the database owner has read permissions.

Example

SEC_USER_AUDIT_ACTION_BANNER=/opt/oracle/admin/data/auditwarning.txt

5.2.26 SEC_USER_UNAUTHORIZED_ACCESS_BANNER

Purpose

To specify a text file containing the banner contents that warn the user about unauthorized access to the database.

Usage Notes

The complete path of the text file must be specified in the sqlnet.ora file on the server. OCI applications can make use of OCI features to retrieve this banner and display it to the user.

Default

None

Values

Name of the file for which the database owner has read permissions.

Example

SEC USER UNAUTHORIZED ACCESS BANNER=/opt/oracle/admin/data/unauthwarning.txt

5.2.27 SEND BUF SIZE

Use the sqlnet parameter $SEND_BUF_SIZE$ to specify the buffer space limit for session send operations.

Purpose

To specify the buffer space limit for send operations of sessions.



Usage Notes

You can override this parameter for a particular client connection by specifying the SEND_BUF_SIZE parameter in the connect descriptor for a client.

This parameter is supported by the TCP/IP, TCP/IP with TLS, and SDP protocols.



Additional protocols might support this parameter on certain operating systems. Refer to the operating system-specific documentation for additional information about additional protocols that support this parameter.

Default

The default value for this parameter is operating system specific. The default for Linux 2.6 operating system is 16 KB.

Example

SEND BUF SIZE=11784

Related Topics

Oracle Database Net Services Administrator's Guide

5.2.28 SQLNET.ALLOW WEAK CRYPTO

Use the sqlnet.ora compatibility parameter $SQLNET.ALLOW_WEAK_CRYPTO$ to configure your client-side network connection by reviewing the specified encryption and crypto-checksum algorithms.

Purpose

To configure your client-side network connection by reviewing the encryption and crypto-checksum algorithms enabled on the client and server. This ensures that the connection does not encounter compatibility issues and your configuration uses supported strong algorithms.

Usage Notes

• The DES, DES40, 3DES112, 3DES168, RC4_40, RC4_56, RC4_128, RC4_256, and MD5 algorithms are deprecated in this release.

As a result of this deprecation, Oracle recommends that you review your network encryption and integrity configuration to check if you have specified any of the deprecated weak algorithms.

To transition your Oracle Database environment to use stronger algorithms, download and install the patch described in My Oracle Support note 2118136.2.

- If you set this parameter to TRUE, then you can specify deprecated algorithms for backward compatibility. This configuration allows patched clients to connect to unpatched servers, and thus such a connection is less secure.
- If you set this parameter to FALSE, then you can specify only supported algorithms so that clients and servers can communicate in a fully patched environment. The server enforces key fold-in for all Kerberos and JDBC thin clients. This configuration strengthens the

connection between clients and servers by using strong native network encryption and integrity capabilities.

Using this setting, if native network encryption or checksumming is enabled and a patched server or client attempts to communicate with an unpatched old client or server, then the connection fails with an error message.

Values

- TRUE
- FALSE

Default Value

TRUE

Recommended Value

FALSE



Before setting this parameter to FALSE, you must remove all deprecated algorithms listed in the server and client sqlnet.ora files.

Example

SQLNET.ALLOW WEAK CRYPTO = FALSE

Related Topics

- Oracle Database Security Guide
- SQLNET.ENCRYPTION_TYPES_CLIENT
 Use the sqlnet.ora parameter SQLNET.ENCRYPTION_TYPES_CLIENT to list encryption
 algorithms for clients to use.
- SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT

 Use the sqlnet.ora parameter SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT to specify a list of data integrity algorithms that this client or server acting as a client uses.

5.2.29 SQLNET.ALLOW WEAK CRYPTO CLIENTS

Use the sqlnet.ora compatibility parameter SQLNET.ALLOW_WEAK_CRYPTO_CLIENTS to configure your server-side network connection by reviewing the specified encryption and cryptochecksum algorithms.

Purpose

To configure your server-side network connection by reviewing the encryption and cryptochecksum algorithms enabled on the client and server. This ensures that the connection does not encounter compatibility issues and your configuration uses supported strong algorithms.



Usage Notes

• The DES, DES40, 3DES112, 3DES168, RC4_40, RC4_56, RC4_128, RC4_256, and MD5 algorithms are deprecated in this release.

As a result of this deprecation, Oracle recommends that you review your network encryption and integrity configuration to check if you have specified any of the deprecated weak algorithms.

To transition your Oracle Database environment to use stronger algorithms, download and install the patch described in My Oracle Support note 2118136.2.

- If you set this parameter to TRUE, then you can specify deprecated algorithms for backward compatibility. This configuration allows patched servers to connect to unpatched clients, and thus such a connection is less secure.
- If you set this parameter to FALSE, then you can specify only supported algorithms so that
 clients and servers can communicate in a fully patched environment. The server enforces
 key fold-in for all Kerberos and JDBC thin clients. This configuration strengthens the
 connection between clients and servers by using strong native network encryption and
 integrity capabilities.

Using this setting, if native network encryption or checksumming is enabled and a patched server or client attempts to communicate with an unpatched old client or server, then the connection fails with an error message.

Values

- TRUE
- FALSE

Default Value

TRUE

Recommended Value

FALSE



Before setting this parameter to FALSE, you must remove all deprecated algorithms listed in the server and client sqlnet.ora files.

Example

SQLNET.ALLOW WEAK CRYPTO CLIENTS = FALSE

Related Topics

- Oracle Database Security Guide
- SQLNET.ENCRYPTION_TYPES_SERVER
 Use the sqlnet.ora parameter SQLNET.ENCRYPTION_TYPES_SERVER to list the encryption algorithms for the database to use.



SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER

Use the sqlnet.ora parameter SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER to specify the data integrity algorithms that this server or client to another server uses, in order of intended use.

5.2.30 SQLNET.ALLOWED LOGON VERSION CLIENT

Use the sqlnet parameter SQLNET.ALLOWED_LOGON_VERSION_CLIENT to define minimum authentication protocols that servers acting as clients to other servers can use for connecting to Oracle Database instances.

Purpose

To set the minimum authentication protocol allowed for clients when a server is acting as a client, such as connecting over a database link, when connecting to Oracle Database instances.

Usage Notes

The term **VERSION** in the parameter name refers to the version of the authentication protocol, not the Oracle Database release.

If the version does not meet or exceed the value defined by this parameter, then authentication fails with an ORA-28040: No matching authentication protocol error.

Values

12a for Oracle Database 12c Release 1 (12.1.0.2) or later (strongest protection)



Using this setting, the clients can only authenticate using a de-optimized password version. For example, the 12C password version.

 12 for the critical patch updates CPUOct2012 and later Oracle Database 11g authentication protocols (stronger protection)



Using this setting, the clients can only authenticate using a verifier that uses salt. For example, the 11G or 12C password versions.

- 11 for Oracle Database 11q authentication protocols (default)
- 10 for Oracle Database 10g authentication protocols
- 8 for Oracle8*i* authentication protocol

Default

11



Examples

• If an Oracle Database 19c database hosts a database link to an Oracle Database 18c database, then set the SQLNET.ALLOWED_LOGON_VERSION_CLIENT parameter as follows for the database link connection to proceed:

SQLNET.ALLOWED LOGON VERSION CLIENT=12

In this case, you cannot configure the more secure

SQLNET.ALLOWED_LOGON_VERSION_CLIENT setting of 12a on the 19c server hosting the database link because the account on the Oracle Database 18c database might not have its password changed and thus might only have the 11g verifier.

If an Oracle Database 19c database hosts a database link to an old server, such as Oracle
Database 11g database, then set the SQLNET.ALLOWED_LOGON_VERSION_CLIENT parameter
as follows to allow the database link connection to proceed using the 11g verifier:

SQLNET.ALLOWED LOGON VERSION CLIENT=11

Related Topics

- Oracle Database Reference
- Oracle Database Security Guide

5.2.31 SQLNET.ALLOWED_LOGON_VERSION_SERVER

Use the sqlnet.ora parameter SQLNET.ALLOWED_LOGON_VERSION_SERVER to set the minimum authentication protocol that is permitted when connecting to Oracle Database instances.

Purpose

To set the minimum authentication protocol for connecting to Oracle Database instances.

Usage Notes

Authentication Protocol Versions:

The term **VERSION** in the parameter name refers to the version of the authentication protocol, not the Oracle Database release.

A value that appears higher up in Table 5-1 is less compatible (in terms of the protocol that clients must understand in order to authenticate) but simultaneously more secure than a value that appears lower down. The server is also more restrictive in terms of the password version that must exist to authenticate any specific account. Whether a client can authenticate to a specific account depends on both the server's setting of its SQLNET.ALLOWED_LOGON_VERSION_SERVER parameter, as well as on the password versions which exist for the specified account. The list of password versions can be seen in DBA USERS.PASSWORD VERSIONS.

The authentication fails with an ORA-28040: No matching authentication protocol or an ORA-03134: Connections to this server version are no longer supported error if the client does not have the ability listed in the "Ability Required of the Client" column corresponding to the row matching the value of the

SQLNET.ALLOWED LOGON VERSION SERVER parameter in Table 5-1.



A setting of 12 (the default) enables only the 11G and 12C password versions. A setting of 12a enables only the 12C password version.

Note the following implications of setting the value to 12 or 12a:

A value of FALSE for the SEC_CASE_SENSITIVE_LOGON Oracle instance initialization parameter must not be used because password case insensitivity requires the use of the 10g password version. If the SEC_CASE_SENSITIVE_LOGON Oracle instance initialization parameter is set to FALSE, then user accounts and secure roles become unusable because Exclusive Mode excludes the use of the 10g password version. The SEC_CASE_SENSITIVE_LOGON Oracle instance initialization parameter enables or disables password case sensitivity. However, since Exclusive mode is enabled by default in this release, disabling the password case sensitivity is not supported.

Note:

- * The use of the Oracle instance initialization parameter SEC_CASE_SENSITIVE_LOGON is deprecated in favor of setting the SQLNET.ALLOWED_LOGON_VERSION_SERVER parameter to 12 to ensure that passwords are treated in a case-sensitive fashion.
- * Disabling password case sensitivity is not supported in Exclusive mode (when SQLNET.ALLOWED LOGON VERSION SERVER is set to 12 or 12a.)
- Releases of OCI clients earlier than Oracle Database 10g cannot authenticate to the Oracle database using password-based authentication.
- If the client uses Oracle Database 10g, then the client will receive an ORA-03134: Connections to this server version are no longer supported error message. To allow the connection, set the SQLNET.ALLOWED_LOGON_VERSION_SERVER value to 8. Ensure the DBA_USERS.PASSWORD_VERSIONS value for the account contains the value 10g. It may be necessary to reset the password for that account.
- To take advantage of the 12c password version introduced in Oracle Database release 12.2, user passwords should be expired to encourage users to change their passwords and cause the new 12c password version to be generated for their account. By default in this release, new passwords are treated in a case-sensitive fashion. When an account password is changed, the earlier 10g case-insensitive password version and the 11g password version are both automatically removed, and the new 12c password version is generated.
- JDBC Thin Client Support:

In Oracle Database release 12.1.0.2 and later, if you set the sqlnet.ora parameter SQLNET.ALLOWED_LOGON_VERSION_SERVER to 12a and you create a new account or change the password of an existing account, then only the new 12C password version is generated. The 12C password version is based on a SHA-2 (Secure Hash Algorithm) SHA-512 salted cryptographic hash deoptimized using the PBKDF2 (Password-Based Key Derivation Function 2) algorithm. When the database server is running with ALLOWED_LOGON_VERSION_SERVER set to 12a, it is running in Exclusive Mode. In this mode, to log in using a JDBC client, the JRE version must be at least version 8. The JDBC client enables its 07L_MR capability flag only when it is running with at least version 8 of the JRE.



Note:

Check the PASSWORD_VERSIONS column of the DBA_USERS catalog view in Oracle Database Reference to see the list of password versions for any given account.

If you set the sqlnet.ora parameter SQLNET.ALLOWED_LOGON_VERSION_SERVER to 12, the server runs in Exclusive Mode and only the 11g and 12c password versions (the SHA-1 and PBKDF2 SHA-2 based hashes of the password, respectively) are generated and allowed to be used. In such cases, fully-patched JDBC clients having the CPUOct2012 patch can connect because these JDBC clients provide the O5L_NP client ability.

Older JDBC clients which do not have the CPUOct2012 containing the fix for the stealth password cracking vulnerability CVE-2012-3132, do not provide the OSL_NP client ability. Therefore, ensure that all the JDBC clients are patched properly.

Client Capabilities:

The client must support certain abilities of the authentication protocol before the server will authenticate. If the client does not support a specified authentication ability, then the server rejects the connection with an ORA-28040: No matching authentication protocol error message.

The following is the list of all client abilities. Some clients do not have all the listed abilities. Clients that are more recent have all of the capabilities of the older clients, but older clients tend to have fewer abilities than more recent clients. An ability that appears at the top in this list is more recent and secure than an ability that appears lower toward the bottom:

- 08L LI: The ability to support long identifiers (user names up to 128 bytes).
- O7L_MR: The ability to perform the Oracle Database 10g authentication protocol using the 12C password version. For JDBC clients, only those running on at least JRE version 8 offer the O7L_MR capability.
- O5L_NP: The ability to perform the Oracle Database 10g authentication protocol using the 11g password version, and generating a session key encrypted for critical patch update CPUOct2012.
- 05L: The ability to perform the Oracle Database 10g authentication protocol using the 10g password version.
- 04L: The ability to perform the Oracle9i database authentication protocol using the 10G password version.
- O3L: The ability to perform the Oracle8i database authentication protocol using the 10G password version.

Using the Gradual Database Password Rollover Feature

When the gradual database password rollover feature is enabled for an account, the ${\tt LOGON_INFO}$ clause in the audit record enables you to see whether the user has logged in with the old password or whether an application has not yet been updated to log in using the new password.

For example:

```
(TYPE=(DATABASE));
(CLIENT ADDRESS=((PROTOCOL=ipc)(HOST=0.0.0.0)));
```



(LOGON_INFO=((VERIFIER=11G-OLD)
(CLIENT_CAPABILITIES=05L_NP,07L_MR,08L_LI)));

Allowed Parameter Settings:

The following table describes the allowed settings of the <code>SQLNET.ALLOWED_LOGON_VERSION_SERVER</code> parameter, its effect on the generated password versions when an account is created or a password is changed, the ability flag required of the client to authenticate while the server has this setting, and whether the setting is considered to be an Exclusive Mode.

Table 5-1 SQLNET.ALLOWED_LOGON_VERSION_SERVER Settings

Value of the ALLOWED_LOG ON_VERSION_S ERVER Parameter	Generated Password Version	Ability Required of the Client	Meaning for Clients	Server Runs in Exclusive Mode
12a	12C	O7L_MR	Only Oracle Database 12c release 1 (12.1.0.2 or later) clients can connect to the server.	Yes because it excludes the use of both 10G and 11G password versions.
12	11G, 12C	O5L_NP	Oracle Database 11 <i>g</i> release 2 (11.2.0.3 or later) clients can connect to the server.	Yes because it excludes the use of the 10G password version.
			Older clients need the critical patch update CPUOct2012 or later, to gain the O5L_NP ability.	
			Only older clients which have applied critical patch update CPUOct2012 or later can connect to the server.	



Table 5-1 (Cont.) SQLNET.ALLOWED_LOGON_VERSION_SERVER Settings

Value of the ALLOWED_LOG ON_VERSION_S ERVER Parameter	Generated Password Version	Ability Required of the Client	Meaning for Clients	Server Runs in Exclusive Mode
11	10G, 11G, 12C	05L	Clients using Oracle Database 10g and later can connect to the server.	No
			Clients using releases earlier than Oracle Database release 11.2.0.3 that have not applied critical patch update CPUOct2012 or later patches must use the 10G password version.	
10	10G, 11G, 12C	05L	It has the same meaning as the earlier row.	No
9	10G, 11G, 12C	O4L	It has the same meaning as the earlier row.	No
8	10G, 11G, 12C	O3L	It has the same meaning as the earlier row.	No

Values

- 12a for Oracle Database 12c release 12.1.0.2 or later authentication protocols (strongest protection)
- 12 for Oracle Database 12c release 12.1 authentication protocols (default and recommended value)
- 11 for Oracle Database 11*g* authentication protocols
- 10 for Oracle Database 10g authentication protocols
- 9 for Oracle9*i* Database authentication protocol
- 8 for Oracle8*i* Database authentication protocol

Note:

- Starting with Oracle Database 12c Release 2 (12.2), the default value is 12.
- For earlier releases, the value 12 can be used after the critical patch updates CPUOct2012 and later are applied.

Default

12

Example

SQLNET.ALLOWED_LOGON_VERSION_SERVER=12

Related Topics

- Ensuring Against Password Security Threats by Using the 12C Password Version
- Managing Gradual Database Password Rollover for Applications

5.2.32 SQLNET.AUTHENTICATION SERVICES

Use the sqlnet.ora parameter SQLNET.AUTHENTICATION_SERVICES to enable one or more authentication services.

Purpose

To enable one or more authentication services. If you have installed authentication, then Oracle recommends that you set <code>SQLNET.AUTHENTICATION_SERVICES</code> to either <code>NONE</code> or to one of the listed authentication methods.

Usage Notes

You can set this parameter in the sqlnet.ora file.

You can also set this value in the <code>tnsnames.ora</code> file or directly as part of the connect string. The <code>SQLNET.AUTHENTICATION_SERVICES</code> parameter is equivalent to the <code>tnsnames.ora</code> parameter <code>AUTHENTICATION_SERVICE</code>. The parameter value specified in the connect string takes precedence.

- When you set SQLNET.AUTHENTICATION_SERVICES to ALL, the server attempts to authenticate using each of the following methods.
 - Authentication based on a service external to the database, such as a service on the network layer, Kerberos, or RADIUS.
 - Authentication based on the operating system user's membership in an administrative operating system group. Group names are platform-specific. This authentication applies to administrative connections only.
 - 3. Authentication performed by the database.
 - 4. Authentication based on credentials stored in a directory server.

The server falls back to the authentication methods that appear further down on the list if attempts to use the authentication methods appearing higher on the list were unsuccessful.



- When using local database password authentication (no external authentication), set SQLNET.AUTHENTICATION SERVICES= (NONE) for better client performance.
- Operating system authentication enables access to the database using any user name and any password when an administrative connection is attempted to the CDB root, such as using the AS SYSDBA clause when connecting using SQL*Plus.

An example of a connection to the CDB root is as follows.

```
sqlplus ignored_username/ignored_password AS SYSDBA
```

When the operating-system user who issued the preceding command is already a member of the appropriate administrative operating system group, then the connection is successful. This is because Oracle checks the group membership first, and thus the user name and password are ignored by the server.

Values

Authentication methods that are available with Oracle Net Services:

- NONE for no authentication methods, including Microsoft Windows native operating system authentication. When you set SQLNET.AUTHENTICATION_SERVICES to NONE, then the user can use a valid user name and password to access the database.
- ALL for all authentication methods.
- BEQ for native operating system authentication for operating systems other than Microsoft Windows.
- KERBEROS5 for Kerberos authentication.
- RADIUS for Remote Authentication Dial-In User Service (RADIUS) authentication.
- TCPS for TLS authentication.
- NTS for Microsoft Windows native operating system authentication. In this case, the user must authenticate to the database (CDB root) with OS credentials using Windows native authentication. No external password is needed. NTS checks the group membership for an OS user. For example, if an OS user is a member of the ORA_DBA group, then the user can log in to the database as SYSDBA.

Note:

With the SQLNET.AUTHENTICATION_SERVICES value NTS, if you try to connect through SQL*Plus using NTS authentication and specify an external password (for example, SQL*Plus SYSTEM/password), then the connection fails with an ORA-12638: credential retrieval failed error. For regular user name and password based authentication, set the value to NONE.

Default

TCPS





When installing Oracle Database with Database Configuration Assistant (DBCA), you can set this parameter to NTS in the sqlnet.ora file.

Examples

When specifying multiple authentication services, you must enclose the values within parentheses as follows:

SQLNET.AUTHENTICATION_SERVICES=(BEQ, KERBEROS5)

If you want to specify only a single authentication service, then parentheses are not required:

SQLNET.AUTHENTICATION SERVICES=KERBEROS5

Related Topics

- Configuring Authentication
- Setting the SQLNET.AUTHENTICATION_SERVICES Parameter in sqlnet.ora
- AUTHENTICATION_SERVICE
 Use the tnsnames.ora parameter AUTHENTICATION_SERVICE to enable an authentication service.

5.2.33 SQLNET.CLIENT_REGISTRATION

Purpose

To set a unique identifier for the client computer.

Usage Notes

This identifier is passed to the listener with any connection request, and is included in the audit trail. The identifier can be any alphanumeric string up to 128 characters long.

Default

None

Example

SQLNET.CLIENT REGISTRATION=1432

5.2.34 SQLNET.CLOUD USER

Purpose

To specify a user name for the web server HTTP basic authentication.



Usage Notes

When secure websocket protocol is used, the client uses this user as the user name for authentication. The password for this user should be stored in a wallet using mkstore commands

Configuration steps to use HTTP basic authentication with secure websockets:

1. Create wallet using orapki utility.

```
orapki wallet create -wallet wallet_directory
Example
orapki wallet create -wallet /app/wallet
```

2. Add web server public certificate.

```
orapki wallet -wallet wallet_directory -trusted_cert -cert
web_server_public_certificate_in_pem_format
Example
orapki wallet -wallet /app/wallet -trusted_cert -cert server_cert.txt
```

3. Add web server user name to sqlnet.ora. This user name is only used for authenticating the web server. This is not a database user name. After web server authentication, the web server makes connection to the backend database server and usual database authentication happens.

Example

```
sqlnet.cloud user = dbuser1
```

4. Add web server user password to wallet.

```
mkstore -wrl wallet_location -createEntry username password
Example
mkstore -wrl /app/wallet -createEntry dbuser1 Secretdb#
```

5. Make wallet auto login and protect this wallet directory using operating system file permissions or any other means, so that ONLY database client can have read access to it. Refer to the operating system utilities for information about changing file permissions.

```
orapki wallet create -wallet wallet_directory -auto_login
Example
orapki wallet create -wallet /app/wallet -auto login
```

6. Update sqlnet.ora with wallet entry.

Example

```
wallet_location=(SOURCE= (METHOD=file) (METHOD_DATA= (DIRECTORY=/app/
wallet)))
```

Default

None

5.2.35 SQLNET.COMPRESSION

Purpose

To enable or disable data compression. If both the server and client have this parameter set to ON, then compression is used for the connection.



The SQLNET.COMPRESSION parameter applies to all database connections, except for Oracle Data Guard streaming redo and SecureFiles LOBs (Large Objects).

Default

off

Values

- on to enable data compression.
- off to disable data compression.

Example

SQLNET.COMPRESSION=on

5.2.36 SQLNET.COMPRESSION_ACCELERATION

Purpose

To specify the use of hardware accelerated version of compression using this parameter if it is available for that platform.

Usage Notes

This parameter can be specified under Oracle Connection Manager alias description.

Default

on

Values

- on
- off
- 0
- 1

Example 5-4 Example

compression acceleration = on

5.2.37 SQLNET.COMPRESSION_LEVELS

Purpose

To specify the compression level.

Usage Notes

The compression levels are used at time of negotiation to verify which levels are used at both ends, and to select one level.

For Database Resident Connection Pooling (DRCP), only the compression level low is supported.

Default

low

Values

- low to use low CPU usage and low compression ratio.
- high to use high CPU usage and high compression ratio.

Example

SQLNET.COMPRESSION LEVELS=(high)

5.2.38 SQLNET.COMPRESSION_THRESHOLD

Purpose

To specify the minimum data size, in bytes, for which compression is needed.

Usage Notes

Compression is not be done if the size of the data to be sent is less than this value.

Default

1024 bytes

Example

SQLNET.COMPRESSION THRESHOLD=1024

5.2.39 SQLNET.CRYPTO CHECKSUM CLIENT

Use the sqlnet.ora parameter SQLNET.CRYPTO_CHECKSUM_CLIENT to specify the desired data integrity behavior when this client or server acting as a client connects to a server.

Purpose

To specify the checksum behavior for the client. The behavior partially depends on the SQLNET.CRYPTO CHECKSUM SERVER setting at the other end of the connection.



Default

accepted

Values

- accepted to enable the security service if required or requested by the other side.
- rejected to disable the security service, even if required by the other side.
- requested to enable the security service if the other side allows it.
- required to enable the security service and disallow the connection if the other side is not enabled for the security service.

Example

SQLNET.CRYPTO CHECKSUM CLIENT=accepted

Related Topics

Oracle Database Security Guide

5.2.40 SQLNET.CRYPTO_CHECKSUM_SERVER

Use the sqlnet.ora parameter SQLNET.CRYPTO_CHECKSUM_SERVER to specify the data integrity behavior when a client or another server acting as a client connects to this server.

Purpose

To specify the checksum behavior for the database. The behavior partially depends on the SQLNET.CRYPTO CHECKSUM CLIENT setting at the other end of the connection.

Default

accepted

Values

- accepted to enable the security service if required or requested by the other side.
- rejected to disable the security service, even if required by the other side.
- requested to enable the security service if the other side allows it.
- required to enable the security service and disallow the connection if the other side is not enabled for the security service.

Example

SQLNET.CRYPTO_CHECKSUM_SERVER=accepted

Related Topics

Oracle Database Security Guide



5.2.41 SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT

Use the sqlnet.ora parameter SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT to specify a list of data integrity algorithms that this client or server acting as a client uses.

Purpose

To specify a list of crypto-checksum algorithms for the client to use.

This list is used to negotiate a mutually acceptable algorithm with the other end of the connection. If an algorithm that is not installed on this side is specified, the connection terminates with the ORA-12650: No common encryption or data integrity algorithm error error message.

Default

All available algorithms

Values

MD5 for the RSA Data Security MD5 algorithm.

The MD5 algorithm is deprecated in this release. To transition your Oracle Database environment to use stronger algorithms, download and install the patch described in My Oracle Support note 2118136.2.

- SHA1 for the Secure Hash Algorithm.
- SHA256 for SHA-2 uses 256 bits with the hashing algorithm.
- SHA384 for SHA-2 uses 384 bits with the hashing algorithm.
- SHA512 for SHA-2 uses 512 bits with the hashing algorithm.

Example

SQLNET.CRYPTO_CHECKSUM_TYPES_CLIENT=(SHA256, MD5)

Related Topics

Oracle Database Security Guide

5.2.42 SQLNET.CRYPTO CHECKSUM TYPES SERVER

Use the sqlnet.ora parameter SQLNET.CRYPTO_CHECKSUM_TYPES_SERVER to specify the data integrity algorithms that this server or client to another server uses, in order of intended use.

Purpose

To specify a list of crypto-checksum algorithms for the database to use.

This list is used to negotiate a mutually acceptable algorithm with the other end of the connection. Each algorithm is checked against the list of available client algorithm types until a match is found. If an algorithm is specified that is not installed on this side, the connection terminates with the ORA-12650: No common encryption or data integrity algorithm error error message.



Default

All available algorithms

Values

MD5 for the RSA Data Security's MD5 algorithm

The MD5 algorithm is deprecated in this release. To transition your Oracle Database environment to use stronger algorithms, download and install the patch described in My Oracle Support note 2118136.2.

- SHA1 for the Secure Hash algorithm.
- SHA256 for SHA-2 uses 256 bits with the hashing algorithm.
- SHA384 for SHA-2 uses 384 bits with the hashing algorithm.
- SHA512 for SHA-2 uses 512 bits with the hashing algorithm.

Example

SQLNET.CRYPTO CHECKSUM TYPES SERVER=(SHA256, MD5)

Related Topics

Oracle Database Security Guide

5.2.43 SQLNET.DBFW_PUBLIC_KEY

Purpose

To provide Oracle Database Firewall public keys to Advanced Security Option (ASO) by specifying the file that stores the Oracle Database Firewall public keys.

Default

None

Values

Full path name of the operating system file that has the public keys.

Example

SQLNET.DBFW_PUBLIC_KEY="/path_to_file/dbfw_public_key_file.txt"



"SQLNET.ENCRYPTION_TYPES_SERVER"



5.2.44 SQLNET.DOWN_HOSTS_TIMEOUT

Purpose

To specify the amount of time in seconds that information about the down state of server hosts is kept in client process cache.

Usage Notes

Clients discover the <code>down</code> state of server hosts when attempting connections. When a connection attempt fails, the information about the <code>down</code> state of the server host is added to the client process cache. Subsequent connection attempts by the same client process move the <code>down</code> hosts to the end of the address list, thereby reducing the priority of such hosts. When the time <code>specified</code> by the <code>SQLNET.DOWN_HOSTS_TIMEOUT</code> parameter has passed, the host is purged from the process cache, and its priority in the address list is restored.

Default

600 seconds (10 minutes)

Values

Any positive integer

Example

SQLNET.DOWN HOSTS TIMEOUT=60

5.2.45 SQLNET.ENCRYPTION_CLIENT

Use the sqlnet.ora parameter SQLNET.ENCRYPTION_CLIENT to set the encryption behavior when this client or server acting as a client connects to a server.

Purpose

To turn encryption on for the client. Setting the tnsnames.ora parameter IGNORE ANO ENCRYPTION FOR TCPS to TRUE disables SQLNET.ENCRYPTION CLIENT.

The behavior of the client partially depends on the value set for SQLNET.ENCRYPTION_SERVER at the other end of the connection.

Default

accepted

Values

- accepted to enable the security service if required or requested by the other side.
- rejected to disable the security service, even if required by the other side.
- requested to enable the security service if the other side allows it.
- required to enable the security service and disallow the connection if the other side is not enabled for the security service.



Example

SQLNET.ENCRYPTION CLIENT=accepted

Related Topics

Oracle Database Security Guide

5.2.46 SQLNET.ENCRYPTION_SERVER

The sqlnet.ora parameter SQLNET.ENCRYPTION_SERVER specifies the encryption behavior when a client or a server acting as a client connects to this server.

Purpose

To turn encryption on for the database server. Setting

SQLNET.IGNORE_ANO_ENCRYPTION_FOR_TCPS to FALSE disables SQLNET.ENCRYPTION_SERVER.

The behavior of the server partially depends on the SQLNET.ENCRYPTION_CLIENT setting at the other end of the connection.

Default

accepted

Values

- accepted to enable the security service if required or requested by the other side.
- rejected to disable the security service, even if required by the other side.
- requested to enable the security service if the other side allows it.
- required to enable the security service and disallow the connection if the other side is not enabled for the security service.

Example

SQLNET.ENCRYPTION_SERVER=accepted

Related Topics

Oracle Database Security Guide

5.2.47 SQLNET.ENCRYPTION TYPES CLIENT

Use the sqlnet.ora parameter $SQLNET.ENCRYPTION_TYPES_CLIENT$ to list encryption algorithms for clients to use.

Purpose

To specify a list of encryption algorithms for clients to use.

Default

All available algorithms.

Values

One or more of the following:



- 3des112 for triple DES with a two-key (112-bit) option
- 3des168 for triple DES with a three-key (168-bit) option
- aes128 for AES (128-bit key size)
- aes192 for AES (192-bit key size)
- aes256 for AES (256-bit key size)
- des for standard DES (56-bit key size)
- des40 for DES (40-bit key size)
- rc4 40 for RSA RC4 (40-bit key size)
- rc4 56 for RSA RC4 (56-bit key size)
- rc4 128 for RSA RC4 (128-bit key size)
- rc4 256 for RSA RC4 (256-bit key size)

The DES, DES40, 3DES112, 3DES168, RC4_40, RC4_56, RC4_128, and RC4_256 algorithms are deprecated in this release.

To transition your Oracle Database environment to use stronger algorithms, download and install the patch described in My Oracle Support note 2118136.2.

Example

SQLNET.ENCRYPTION TYPES CLIENT=(rc4 56)

See Also:

Oracle Database Security Guide

5.2.48 SQLNET.ENCRYPTION_TYPES_SERVER

Use the sqlnet.ora parameter SQLNET.ENCRYPTION_TYPES_SERVER to list the encryption algorithms for the database to use.

Purpose

To specify a list of encryption algorithms for the database to use.

Default

All available algorithms.

Values

One or more of the following:

- 3des112 for triple DES with a two-key (112-bit) option
- 3des168 for triple DES with a three-key (168-bit) option
- aes128 for AES (128-bit key size)
- aes192 for AES (192-bit key size)



- aes256 for AES (256-bit key size)
- des for standard DES (56-bit key size)
- des40 for DES40 (40-bit key size)
- rc4 40 for RSA RC4 (40-bit key size)
- rc4 56 for RSA RC4 (56-bit key size)
- rc4 128 for RSA RC4 (128-bit key size)
- rc4 256 for RSA RC4 (256-bit key size)

The DES, DES40, 3DES112, 3DES168, RC4_40, RC4_56, RC4_128, and RC4_256 algorithms are deprecated in this release.

To transition your Oracle Database environment to use stronger algorithms, download and install the patch described in My Oracle Support note 2118136.2.

Example

SQLNET.ENCRYPTION TYPES SERVER=(rc4 56, des, ...)

See Also:

Oracle Database Security Guide

5.2.49 SQLNET.EXPIRE_TIME

Use the sqlnet.ora parameter SQLNET.EXPIRE_TIME to specify how often, in minutes, to verify that client and server connections are active.

Purpose

To specify time intervals, in minutes, for how often to verify that client and server connections are active.

Usage Notes

Setting a value greater than 0 ensures that connections are not left open indefinitely, due to an unusual client termination. If the system supports TCP keepalive tuning, then Oracle Net Services automatically uses the enhanced detection model, and tunes the TCP keepalive parameters

If the probe finds a terminated connection, or a connection that is no longer in use, then it returns an error, causing the server process to exit.

This parameter is primarily intended for the database server, which typically handles multiple connections at any one time.

Limitations on using this terminated connection detection feature are:

- It is not allowed on bequeathed connections.
- Though very small, a probe packet generates additional traffic that may downgrade network performance.

Depending on which operating system is in use, the server may need to perform additional
processing to distinguish the connection probing event from other events that occur. This
can also result in degraded network performance.

Default

0

Minimum Value

0

Recommended Value

10

Example

SQLNET.EXPIRE TIME=10

5.2.50 SQLNET.IGNORE_ANO_ENCRYPTION_FOR_TCPS

The SQLNET.IGNORE_ANO_ENCRYPTION_FOR_TCPS parameter is used on the server-side to ignore the value set in SQLNET.ENCRYPTION_SERVER for TCPS connections (effectively disabling ANO encryption on the TCPS listener).

Purpose

Used on the server-side to ignore the value set in SQLNET.ENCRYPTION_SERVER for TCPS connections (effectively disabling ANO encryption on the TCPS listener).

Usage Notes

If you set the SQLNET.ENCRYPTION_CLIENT parameter on the client to required and SQLNET.ENCRYPTION_SERVER on the server to required, and if a TCPS listener is used, then the ORA-12696 Double Encryption Turned On, login disallowed error appears. Starting with this release, you can set a new parameter, SQLNET.IGNORE_ANO_ENCRYPTION_FOR_TCPS, to TRUE to ignore the SQLNET.ENCRYPTION_CLIENT OR SQLNET.ENCRYPTION_SERVER when there is a conflict between the use of a TCPS client and these two parameters are set to required.

Default

FALSE

Example 5-5 Example

SQLNET.IGNORE_ANO_ENCRYPTION_FOR_TCPS=TRUE

5.2.51 SQLNET.INBOUND_CONNECT_TIMEOUT

Purpose

To specify the time, in ms, sec, or min, for a client to connect with the database server and provide the necessary authentication information.



Usage Notes

If the client fails to establish a connection and complete authentication in the time specified, then the database server terminates the connection. In addition, the database server logs the IP address of the client and an ORA-12170: TNS:Connect timeout occurred error message to the sqlnet.log file. The client receives either an ORA-12547: TNS:lost contact or an ORA-12637: Packet receive failed error message.

The default value of this parameter is appropriate for typical usage scenarios. However, if you need to explicitly set a different value, then Oracle recommends setting this parameter in combination with the <code>INBOUND_CONNECT_TIMEOUT_listener_name</code> parameter in the <code>listener.ora</code> file. When specifying the values for these parameters, note the following recommendations:

- Set both parameters to an initial low value.
- Set the value of the INBOUND_CONNECT_TIMEOUT_listener_name parameter to a lower value than the SQLNET.INBOUND CONNECT TIMEOUT parameter.

Default

60 seconds

Example

SQLNET.INBOUND_CONNECT_TIMEOUT=3ms

5.2.52 SQLNET.FALLBACK_AUTHENTICATION

Purpose

To specify whether password-based authentication is going to be attempted if Kerberos authentication fails. This is relevant for direct connections as well as database link connections.

Default

FALSE

Example

SQLNET.FALLBACK AUTHENTICATION=TRUE



Oracle Database Security Guide



5.2.53 SQLNET.KERBEROS5_CC_NAME

Use the sqlnet.ora parameter SQLNET.KERBEROS5_CC_NAME to specify the complete path name to the Kerberos credentials cache file.

Purpose

To specify the complete path name to the Kerberos CC file.

Usage Notes

You can set this parameter in the sqlnet.ora file.

You can also set this value in the <code>tnsnames.ora</code> file or directly as part of the connect string. The <code>SQLNET.KERBEROS5_CC_NAME</code> parameter is equivalent to the <code>tnsnames.ora</code> parameter <code>KERBEROS5_CC_NAME</code>. The parameter value specified in the connect string takes precedence.

Values and Examples

You can use the following formats to specify a value for SQLNET. KERBEROS5 CC NAME:

- If the Oracle database is using a directory cache:
 - SQLNET.KERBEROS5 CC NAME=complete path to cc file

For example:

```
SQLNET.KERBEROS5_CC_NAME=/tmp/kcache
SQLNET.KERBEROS5 CC NAME=D:\tmp\kcache
```

SQLNET.KERBEROS5_CC_NAME=FILE:complete_path_to_cc_ file

For example:

```
SQLNET.KERBEROS5 CC NAME=FILE:/tmp/kcache
```

- If the Oracle database is using the native Windows cache:
 - SQLNET.KERBEROS5 CC NAME=OSMSFT://
 - SQLNET.KERBEROS5 CC NAME=MSLSA:

The OSMSFT and MSLSA options specify that the file is on Microsoft Windows and is running Microsoft Kerberos Key Distribution Center (KDC).



If you want to authenticate multiple Kerberos principals, then you can specify additional Kerberos principals either through the connect string directly or in the tnsnames.ora file.

Default

The default value is operating system-dependent, as follows:

- On Linux and UNIX operating systems: /tmp/krb5cc userid
- On Microsoft Windows operating systems: c:\tmp\krbcache



Related Topics

KERBEROS5 CC NAME

Use the tnsnames.ora parameter KERBEROS5_CC_NAME to specify the complete path name to the Kerberos credentials cache (CC) file.

KERBEROS5 PRINCIPAL

Use the KERBEROS5_PRINCIPAL parameter to set the Kerberos principal name associated with the Kerberos credentials cache (CC) file.

Oracle Database Security Guide

5.2.54 SQLNET.KERBEROS5_CLOCKSKEW

Purpose

To specify how many seconds can pass before a Kerberos credential is considered out of date.

Default

300

Example

SQLNET.KERBEROS5 CLOCKSKEW=1200



Oracle Database Security Guide

5.2.55 SQLNET.KERBEROS5_CONF

Purpose

To specify the complete path name to the Kerberos configuration file, which contains the realm for the default Key Distribution Center (KDC) and maps realms to KDC hosts.

Usage Notes

The KDC maintains a list of user principals and is contacted through the kinit program for the user's initial ticket.

The AUTO_DISCOVER option allows the automatic discovery of KDC and realms. It is the default configuration for Kerberos clients. If there are multiple realms to be specified, then Oracle recommends creating configuration files instead of using the AUTO_DISCOVER option. This option is supported for all operating systems with such a feature.

Default

/krb5/krb.conf on Linux and UNIX operating systems

c:\krb5\krb.conf on Microsoft Windows operating systems



Values

- Directory path to krb.conf file
- AUTO DISCOVER

Example

SQLNET.KERBEROS5 CONF=/krb5/krb.conf



Oracle Database Security Guide

5.2.56 SQLNET.KERBEROS5_CONF_LOCATION

Purpose

To specify the directory for the Kerberos configuration file. The parameter also specifies the file is created by the system, and not by the client.

Usage Notes

The configuration file uses DNS lookup to obtain the realm for the default KDC, and maps realms to the KDC hosts. This option is supported for all operating systems with such a feature.

Default

/krb5 on Linux and UNIX operating systems

c:\krb5 on Microsoft Windows operating systems

Example

SQLNET.KERBEROS5 CONF LOCATION=/krb5

5.2.57 SQLNET.KERBEROS5_KEYTAB

Purpose

To specify the complete path name to the Kerberos principal/secret key mapping file, which is used to extract keys and decrypt incoming authentication information.

Default

/etc/v5srvtab on Linux and UNIX operating systems

c:\krb5\v5srvtab on Microsoft Windows operating systems

Example

 ${\tt SQLNET.KERBEROS5_KEYTAB=/etc/v5srvtab}$





Oracle Database Security Guide

5.2.58 SQLNET.KERBEROS5_REALMS

Purpose

To specify the complete path name to the Kerberos realm translation file, which provides a mapping from a host name or domain name to a realm.

Default

/krb5/krb.realms on Linux and UNIX operating systems

c:\krb5\krb.realms on Microsoft Windows operating systems

Example

SQLNET.KERBEROS5 REALMS=/krb5/krb.realms



Oracle Database Security Guide

5.2.59 SQLNET.KERBEROS5_REPLAY_CACHE

Purpose

To specify replay cache is stored in operating system-managed memory on the server, and that file-based replay cache is not used.

Usage Notes

The OS_MEMORY option specifies the replay cache is stored in operating system-managed memory on the server, and file-based replay cache is not used.

Example

SQLNET_KERBEROS5_REPLAY_CACHE=OS_MEMORY

5.2.60 SQLNET.OUTBOUND_CONNECT_TIMEOUT

Use the sqlnet.ora parameter $SQLNET.OUTBOUND_CONNECT_TIMEOUT$ to specify the amount of time, in milliseconds, seconds, or minutes, in which clients must establish Oracle Net connections to database instances.

Purpose

To specify the time, in ms, sec, or min, for a client to establish an Oracle Net connection to the database instance.



Usage Notes

If an Oracle Net connection is not established in the time specified, then the connect attempt is terminated. The client receives an ORA-12170: TNS:Connect timeout occurred error.

The outbound connect timeout interval is a superset of the TCP connect timeout interval, which specifies a limit on the time taken to establish a TCP connection. Additionally, the outbound connect timeout interval includes the time taken to be connected to an Oracle instance providing the requested service. It accepts different timeouts with or without space between the value and the unit.

Without this parameter, a client connection request to the database server may block for the default TCP connect timeout duration (60 seconds) when the database server host system is unreachable. In case, no unit is mentioned, the default unit is sec.

The outbound connect timeout interval is only applicable for TCP, TCP with TLS, and IPC transport connections.

This parameter is overridden by the CONNECT TIMEOUT parameter in the address description.

Default

None

Example

SQLNET.OUTBOUND_CONNECT_TIMEOUT=10 ms

Related Topics

CONNECT TIMEOUT

Use the thinnames.ora parameter CONNECT_TIMEOUT to specify the amount of time, in milliseconds, seconds, or minutes, in which clients must establish Oracle Net connections to database instances.

5.2.61 SQLNET.RADIUS_ALTERNATE

Purpose

To specify an alternate RADIUS server to use in case the primary server is unavailable.

Usage Notes

The value can be either the IP address or host name of the server.

Default

None

Example

SQLNET.RADIUS_ALTERNATE=radius2





Oracle Database Security Guide

5.2.62 SQLNET.RADIUS_ALTERNATE_PORT

Purpose

To specify the listening port of the alternate RADIUS server.

Default

1645

Example

SQLNET.RADIUS_ALTERNATE_PORT=1667



Oracle Database Security Guide

5.2.63 SQLNET.RADIUS_ALTERNATE_RETRIES

Purpose

To specify the number of times the database server should resend messages to the alternate RADIUS server.

Default

3

Example

SQLNET.RADIUS ALTERNATE RETRIES=4



Oracle Database Security Guide



5.2.64 SQLNET.RADIUS_ALTERNATE_TIMEOUT

Use the sqlnet.ora parameter SQLNET.RADIUS_ALTERNATE_TIMEOUT to set the time for an alternate RADIUS server to wait for a response.

Purpose

To set the time, in seconds, for an alternate RADIUS server to wait for a response.

Syntax

```
SQLNET.RADIUS ALTERNATE TIMEOUT=time in seconds
```

Default

5

Example

SQLNET.RADIUS ALTERNATE TIMEOUT=5

Related Topics

Oracle Database Security Guide

5.2.65 SQLNET.RADIUS_AUTHENTICATION

Purpose

To specify the location of the primary RADIUS server, either by its host name or IP address.

Default

Local host

Example

 ${\tt SQLNET.RADIUS_AUTHENETICATION=} of fice a cct$



Oracle Database Security Guide

5.2.66 SQLNET.RADIUS_AUTHENTICATION_INTERFACE

Purpose

To specify the class containing the user interface used to interact with the user.

Default

DefaultRadiusInterface

Example

SQLNET.RADIUS_AUTHENTICATION_INTERFACE=DefaultRadiusInterface



Oracle Database Security Guide

5.2.67 SQLNET.RADIUS_AUTHENTICATION_PORT

Purpose

To specify the listening port of the primary RADIUS server.

Default

1645

Example

SQLNET.RADIUS_AUTHENTICATION_PORT=1667



Oracle Database Security Guide

5.2.68 SQLNET.RADIUS_AUTHENTICATION_RETRIES

Purpose

To specify the number of times the database server should resend messages to the primary RADIUS server.

Default

3

Example

SQLNET.RADIUS AUTHENTICATION RETRIES=4



Oracle Database Security Guide

5.2.69 SQLNET.RADIUS_AUTHENTICATION_TIMEOUT

Purpose

To specify the time, in seconds, that the database server should wait for a response from the primary RADIUS server.

Default

5

Example

SQLNET.RADIUS_AUTHENTICATION_TIMEOUT=10



Oracle Database Security Guide

5.2.70 SQLNET.RADIUS_CHALLENGE_KEYWORD

Use the sqlnet.ora parameter SQLNET.RADIUS_CHALLENGE_KEYWORD to set the keyword for requesting a challenge from the RADIUS server.

Purpose

To set the keyword for requesting a challenge from the RADIUS server. By setting the challenge keyword, you let the user avoid using a password on the client to verify identity.

Syntax

SQLNET.RADIUS CHALLENGE KEYWORD=keyword

Default

challenge

Example

SQLNET.RADIUS_CHALLENGE_KEYWORD=challenge

Related Topics

Oracle Database Security Guide



5.2.71 SQLNET.RADIUS_CHALLENGE_RESPONSE

Use the sqlnet.ora parameter SQLNET.RADIUS_CHALLENGE_RESPONSE to enable or disable challenge responses.

Purpose

To turn the challenge responses on or off.

Default

off

Values

on | off

Example

SQLNET.RADIUS_CHALLENGE_RESPONSE=on

5.2.72 SQLNET.RADIUS_CLASSPATH

Use the sqlnet.ora parameter SQLNET.RADIUS_CLASSPATH to set the path for Java classes and JDK Java libraries.

Purpose

To set the path for Java classes for a graphical interface, and to set the path to JDK Java libraries.

If you use the challenge-response authentication mode, then RADIUS displays a Java-based graphical interface. This interface first requests a password and then additional information, for example, a dynamic password that the user obtains from a token card.

Syntax

SQLNET.RADIUS CLASSPATH=path to GUI Java classes

Default

\$ORACLE HOME/jlib/netradius.jar:\$ORACLE HOME/JRE/lib/sparc/native threads

Example

SQLNET.RADIUS CLASSPATH=/jre1.1

Related Topics

Oracle Database Security Guide

5.2.73 SQLNET.RADIUS_SECRET

Purpose:

To specify the location of the RADIUS secret key.



Default

The ORACLE HOME/network/security/radius.key file.

Example

SQLNET.RADIUS_SECRET=oracle/bin/admin/radiuskey



Oracle Database Security Guide

5.2.74 SQLNET.RADIUS_SEND_ACCOUNTING

Purpose

To turn accounting on and off. If enabled, then packets are sent to the active RADIUS server at listening port plus one.

Usage Notes

The default port is 1646.

Default

off

Values

on \mid off

Example

SQLNET.RADIUS_SEND_ACCOUNTING=on



Oracle Database Security Guide

5.2.75 SQLNET.RECV_TIMEOUT

Use the sqlnet.ora parameter SQLNET.RECV_TIMEOUT to specify the duration of time that a database client or server should wait for data from a peer after establishing a connection.

Purpose

To specify the time for a database client or server to wait for data from the peer after establishing a connection. The peer must send some data within the time interval.



You can specify the time in hours, minutes, seconds, or milliseconds by using the hr, min, sec, or ms keyword respectively. If you do not specify a unit of measurement, then the default unit is sec.

Usage Notes

Setting this parameter for clients ensure that receive operation is not left in wait state indefinitely or for a long period due to an unusual termination of server process or server busy state. If a client does not receive response data in time specified, then it logsORA-12535: TNS:operation timed out and ORA-12609: TNS: Receive timeout occurred messages to the sqlnet.log file. If you choose to set the value, then set the value to an initial low value and adjust according to the system and network capacity. If necessary, use this parameter with the SQLNET.SEND_TIMEOUT parameter.

You can also set this parameter on the server-side to specify the time, in ms, sec, or min, for a server to wait for client data after connection establishment. If a client does not send any data in time specified, then the database server logs <code>ORA-12535</code>: <code>TNS:operation timed out and ORA-12609</code>: <code>TNS: Receive timeout occurred messages to the sqlnet.log file. Without this parameter, the database server may continue to wait for data from clients that may be down or are experiencing difficulties. The server usually blocks on input from the client and gets these timeouts frequently if set to a low value.</code>

Default Value

None

Minimum Value

1 ms

Allowed Range

Any number greater than the minimum value of 1 ms up to 4294967295 ms.

Example

SQLNET.RECV TIMEOUT=10 ms

Related Topics

Oracle Database Net Services Administrator's Guide

5.2.76 SQLNET.SEND TIMEOUT

Use the sqlnet.ora parameter SQLNET.SEND_TIMEOUT to specify the duration of time for a database server to complete a send operation to clients after establishing a connection.

Purpose

To specify the time for a database server to complete a send operation to clients after establishing a connection.

You can specify the time in hours, minutes, seconds, or milliseconds by using the hr, min, sec, or ms keyword respectively. If you do not specify a unit of measurement, then the default unit is sec.



Usage Notes

Setting this parameter is recommended for environments in which clients shut down occasionally or unusually.

If the database server cannot complete a send operation in the time specified, then it logs ORA-12535: TNS:operation timed out and ORA-12608: TNS: Send timeout occurred messages to the sqlnet.log file. Without this parameter, the database server may continue to send responses to clients that are unable to receive data due to a downed computer or a busy state.

You can also set this parameter on the client-side to specify the time, in ms, sec, or min, for a client to complete send operations to the database server after connection establishment. It accepts different timeouts with or without space between the value and the unit. Without this parameter, the client may continue to send requests to a database server already saturated with requests. If you choose to set the value, then set the value to an initial low value and adjust according to system and network capacity.

If necessary, use this parameter with the SQLNET.RECV_TIMEOUT parameter.

Default Value

None

Minimum Value

1 ms

Allowed Range

Any number greater than the minimum value of 1 ms up to 4294967295 ms.

Example

SQLNET.SEND TIMEOUT=3 ms

Related Topics

Oracle Database Net Services Administrator's Guide

5.2.77 SQLNET.URI

SQLNET.URI networking parameter of the sqlnet.ora file specifies a database client URI mapping on the web server.

Purpose

To specify a database client URI mapping on the web server.

Usage Notes

You can use this parameter to customize URI for mapping the database websocket requests coming onto web server to the backend database server. Secure websocket handshaking requests are sent with this URI.

Default

/sqlnet



Example 5-6 Example

sqlnet.uri="/my uri prefix/database/"

5.2.78 SQLNET.USE_HTTPS_PROXY

Purpose

To enable forward HTTP proxy tunneling client connections.

Usage Notes

If turned on, the clients can tunnel secure connections over forward HTTP proxy using HTTP CONNECT method. This helps in accessing the public cloud database service as it eliminates the requirement to open an outbound port on a client side firewall.

This parameter is applicable with Oracle Connection Manager on the server side.

Default

on

Example

SQLNET.USE HTTPS PROXY=on

5.2.79 SQLNET.WALLET_OVERRIDE

Use the sqlnet.ora parameter SQLNET.WALLET_OVERRIDE to determine whether a client should override strong authentication credentials with the password credential from the stored wallet.

Purpose

To determine whether a client should override strong authentication credentials with the password credential from the stored wallet to log in to a database.



This is a client-side parameter. The <code>SQLNET.WALLET_OVERRIDE=TRUE</code> setting on the database server may break external procedures.

Usage Notes

When you use wallets for authentication, the database credentials for user name and password are securely stored in an Oracle wallet. The auto-login feature of the wallet is enabled so that the database does not need a password to open the wallet. From the wallet, the database gets the credentials to access the database for the user.

Wallet use can simplify large-scale deployments that rely on password credentials for connecting to databases. When this feature is configured, application code, batch jobs, and scripts do not need embedded user names and passwords. Risk is reduced because such passwords are no longer exposed, and password management policies are enforced without changing application code whenever user names or passwords change.



Users connect using the <code>connect /@database_name</code> command instead of specifying a user name and password explicitly. This simplifies the maintenance of the scripts and secures the password management for the applications.

Middle-tier applications create an Oracle Applications wallet during installation to store an application's identity. The password may be randomly generated rather than hardcoded. When an Oracle application accesses the database, it sets appropriate values for SQLNET.AUTHENTICATION_SERVICES and WALLET_LOCATION. The new wallet-based password authentication code uses the password credential in the Oracle Applications wallet to log in to the database.

Values

true | false

Example

SQLNET.WALLET OVERRIDE=true

Related Topics

- My Oracle Support Note 340559.1
- Oracle Database Security Guide

5.2.80 SSL_CERT_REVOCATION

Use the sqlnet.ora parameter $SSL_CERT_REVOCATION$ to configure revocation checks for certificates.

Purpose

To configure a revocation check for a certificate.



Oracle Database Security Guide

Default

none

Values

none disables certificate revocation status checking. This is the default value.



Note:

Oracle recommends that you do not set the SSL_CERT_REVOCATION parameter to none because this removes a critical component in certificate-based authentication. Without certificate revocation status checking, you cannot protect against stolen certificates that are used for authentication. Set the none value only in cases where mitigating controls safeguard the use of certificates for authentication, such as network access control lists or Oracle Database Vault policies that limit the database connection to trusted clients.

- requested to perform certificate revocation in case a Certificate Revocation List (CRL) is available. Reject TLS connection if the certificate is revoked. If no appropriate CRL is found to determine the revocation status of the certificate and the certificate is not revoked, then accept the TLS connection.
- required to perform certificate revocation when a certificate is available. If a certificate is
 revoked and no appropriate CRL is found, then reject the TLS connection. If no appropriate
 CRL is found to ascertain the revocation status of the certificate and the certificate is not
 revoked, then accept the TLS connection.

Example

SSL CERT REVOCATION=required

5.2.81 SSL_CRL_FILE

Use the sqlnet.ora parameter SSL_CRL_FILE to specify the name of the file in which you assemble the certificate revocation list (CRL) for client authentication.

Purpose

To specify the name of the file where you can assemble the CRL for client authentication.

Usage Notes

This file contains the PEM-encoded CRL files, in order of preference. You can use this file alternatively or in addition to the SSL_CRL_PATH parameter. This parameter is only valid if SSL CERT REVOCATION is set to either requested or required.

Syntax

SSL CRL FILE=certificate revocation list filename

Default

None

Example

SSL CRL FILE=crl.txt

Related Topics

SSL CERT REVOCATION

Use the sqlnet.ora parameter SSL_CERT_REVOCATION to configure revocation checks for certificates.

SSL CRL PATH

Use the sqlnet.ora parameter SSL_CRL_PATH to specify the destination directory of the certificate revocation list (CRL) for client authentication.

5.2.82 SSL_CRL_PATH

Use the sqlnet.ora parameter SSL_CRL_PATH to specify the destination directory of the certificate revocation list (CRL) for client authentication.

Purpose

To specify the destination directory of the CRL of certificate authority (CA).

Usage Notes

The files in this directory are hashed symbolic links created by Oracle Wallet Manager.

This parameter is only valid if SSL CERT REVOCATION is set to either requested or required.

Syntax

SSL_CRL_PATH=certificate_revocation_list_path

Default

None

Example

SSL_CRL_PATH=/home/user1/crldir

Related Topics

SSL CERT REVOCATION

Use the sqlnet.ora parameter SSL_CERT_REVOCATION to configure revocation checks for certificates.

SSL CRL FILE

Use the sqlnet.ora parameter SSL_CRL_FILE to specify the name of the file in which you assemble the certificate revocation list (CRL) for client authentication.

5.2.83 SSL CIPHER SUITES

Use the SSL_CIPHER_SUITES parameter to control the combination of authentication, encryption, and data integrity algorithms used by Transport Layer Security (TLS).

Purpose

To control the combination of authentication, encryption, and data integrity algorithms used by Transport Layer Security (TLS). By default, the strongest protocol and cipher are negotiated between the database client and server. Setting this parameter will override the default behavior. You must use this parameter only if you have internal security controls that dictate the usage of certain protocol versions.

Usage Notes

Enclose the SSL_CIPHER_SUITES parameter value in parentheses. Otherwise, the cipher suite setting does not parse correctly.

Default

None

Values

Approved ciphers compatible with TLS 1.2:

- TLS ECDHE RSA WITH AES 256 GCM SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS DHE RSA WITH AES 256 GCM SHA384
- TLS DHE RSA WITH AES 128 GCM SHA256

Deprecated ciphers compatible with TLS 1.2:

- TLS ECDHE RSA WITH AES 256 CBC SHA384
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS ECDHE ECDSA WITH AES 256 CBC SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS RSA WITH AES 256 GCM SHA384
- TLS RSA WITH AES 256 CBC SHA256
- TLS_RSA_WITH_AES_128_GCM_SHA256
- TLS RSA WITH AES 128 CBC SHA256
- TLS ECDH RSA WITH AES 256 GCM SHA384
- TLS ECDH RSA WITH AES 128 GCM SHA256
- TLS ECDH RSA WITH AES 256 CBC SHA384
- TLS ECDH RSA WITH AES 128 CBC SHA256
- TLS ECDH ECDSA WITH AES 256 GCM SHA384
- TLS ECDH ECDSA WITH AES 128 GCM SHA256
- TLS ECDH ECDSA WITH AES 256 CBC SHA384
- TLS ECDH ECDSA WITH AES 128 CBC SHA256
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
- TLS DHE RSA WITH AES 128 CBC SHA256
- TLS_DH_anon_WITH_AES_256_GCM_SHA384
- TLS DH anon WITH AES 128 GCM SHA256

Deprecated ciphers compatible with TLS 1.0, TLS 1.1, and TLS 1.2:

- TLS ECDHE RSA WITH AES 256 CBC SHA
- TLS ECDHE RSA WITH AES 128 CBC SHA
- TLS ECDHE ECDSA WITH AES 256 CBC SHA



- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS RSA WITH AES 128 CBC SHA
- TLS_ECDH_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDH_RSA_WITH_AES_128_CBC_SHA
- TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA
- TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA
- TLS DHE RSA WITH AES 256 CBC SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS ECDHE RSA WITH RC4 128 SHA
- TLS ECDHE ECDSA WITH RC4 128 SHA
- TLS_ECDH_RSA_WITH_RC4_128_SHA
- TLS_ECDH_ECDSA_WITH_RC4_128_SHA
- SSL_RSA_WITH_RC4_128_SHA
- SSL RSA WITH RC4 128 MD5
- TLS ECDHE ECDSA WITH NULL SHA
- TLS_ECDHE_RSA_WITH_NULL_SHA
- TLS ECDH ECDSA WITH NULL SHA
- TLS ECDH_RSA_WITH_NULL_SHA
- SSL RSA WITH NULL SHA
- SSL_RSA_WITH_NULL_MD5
- SSL_DH_anon_WITH_RC4_128_MD5

Deprecated ciphers compatible with TLS 1.0 and TLS 1.1:

- TLS ECDHE RSA WITH 3DES EDE CBC SHA
- TLS ECDHE ECDSA WITH 3DES EDE CBC SHA
- TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA
- TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA
- SSL_RSA_WITH_3DES_EDE_CBC_SHA
- SSL DH anon WITH 3DES EDE CBC SHA

Note:

The DH_anon cipher suites do not provide authentication of the communicating parties, and can be vulnerable to man-in-the-middle attacks. Oracle recommends that you do not use these cipher suites to protect sensitive data.



Examples

```
SSL_CIPHER_SUITES=(TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384)

SSL_CIPHER_SUITES=(TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256)
```

Related Topics

- Set the TLS Cipher Suites on the Server
- Set the TLS Cipher Suites on the Client

5.2.84 SSL_CLIENT_AUTHENTICATION

Use the SSL_CLIENT_AUTHENTICATION parameter to specify whether a client is authenticated using Transport Layer Security (TLS).

Purpose

To specify whether a client is authenticated using Transport Layer Security (TLS).

Usage Notes

The database server authenticates the client. Therefore, this value should be set to false. If this parameter is set to true, then the listener attempts to authenticate the client, which can result in a failure.

Default

true

Values

true | false

Example

SSL_CLIENT_AUTHENTICATION=false



Oracle Database Security Guide



5.2.85 SSL_DISABLE_WEAK_EC_CURVES

Use the SSL_DISABLE_WEAK_EC_CURVES parameter to disable the use of weak Elliptic Curve Cryptography (ECC) curves.

Purpose

To disable the use of weak ECC curves with key length less than 256 bits. You can set this parameter in the database server (sqlnet.ora), client (sqlnet.ora or tnsnames.ora connect string), or the listener (listener.ora).

Usage Notes

By default, this parameter is set to FALSE to enable the use of all ECC curves. If you want to enable the use of only Oracle approved curves with ECC curve key size of 256 bits or higher, then set this parameter to TRUE.

When set to TRUE, you can use only the following ECC curves:

- secp256r1
- secp384r1
- secp521r1

Values

- TRUE | ON | YES | 1: To enable only the Oracle approved ECC curves with minimum ECC curve key length of 256 bits
- FALSE | OFF | NO | 0: To enable all ECC curves

Default

FALSE

Examples

• In the tnsnames.ora file:

```
net_service_name=
  (DESCRIPTION=
       (ADDRESS=(PROTOCOL=tcps) (HOST=sales-svr) (PORT=1521))
       (SECURITY=(SSL_DISABLE_WEAK_EC_CURVES=TRUE))
)
```

• In the sqlnet.ora file or the listener.ora file:

```
SSL DISABLE WEAK EC CURVES=TRUE
```

5.2.86 SSL_EXTENDED_KEY_USAGE

Purpose

To specify the purpose of the key in the certificate.

Usage Notes

When this parameter is specified, the certificate with the matching extended key is used.

Values

client authentication

Example

SSL EXTENDED KEY USAGE="client authentication"

5.2.87 SSL_SERVER_DN_MATCH

Use the SSL_SERVER_DN_MATCH parameter to enforce server-side certificate validation through distinguished name (DN) matching.

Purpose

To enforce server-side certification validation through distinguished name (DN) matching.

Usage Notes

If you enforce the DN matching, in addition to verifying the server's certificate chain, the client performs another check through DN matching.

You can configure either partial DN matching or full DN matching. Partial DN matching occurs if the server's CN contains its host name. Full DN matching occurs against the server's complete DN. Not enforcing the match allows the server to potentially fake its identity.

In addition to the sqlnet.ora file, configure the tnsnames.ora parameter SSL SERVER CERT DN to enable full DN matching.

Default

NO

Values

YES | ON | TRUE | 1:

To enforce partial or full DN matching. If the DN matches the service name, then the connection succeeds. If the DN does not match the service name, then the connection fails.

• NO | OFF | FALSE | 0:

To not enforce DN matching. If the DN does not match the service name, then the connection is successful, but an error is logged to the sqlnet.log file.

Example

SSL SERVER DN MATCH=YES

Related Topics

Oracle Database Security Guide



5.2.88 SSL_VERSION

Use the SSL_VERSION parameter to define valid Transport Layer Security (TLS) versions to be used for connections.

Purpose

To define the version of TLS that must run on the systems with which the database server communicates. By default, the database server and client negotiate the strongest security protocol. Oracle does not recommend modifying this parameter, unless your security requirements mandate the usage of certain protocol versions.

Usage Notes

Clients, listeners, and database servers must use compatible versions. Modify this parameter only when necessary to enforce the use of the more secure TLS protocol and not allow clients that only work with the older TLS protocols. If you need to specify TLS 1.0 or TLS 1.1, then also include TLS 1.2 to allow more secure connections. The current default uses TLS 1.2, which is the version required for multiple security compliance requirements.

If you set SSL_VERSION to undetermined, then the most secure TLS protocol version is used. You can use the SSL_VERSION=undetermined setting in the connect string for a specific connection to override the SSL_VERSION value configured in the sqlnet.ora file.

If you do not set SSL_VERSION to any value, then all the supported TLS protocol versions are tried starting with the most secure version. This is typically the most common configuration, ensuring that the strongest protocol is chosen during TLS negotiation.

Default

undetermined

Values

```
undetermined | 1.0 | 1.1 | 1.2
```

The version numbers correspond to the TLS versions, such as TLSv1.0, TLSv1.1, and TLSv1.2.



The sqlnet.ora parameter ADD_SSLV3_TO_DEFAULT has no impact on this parameter.

Syntax and Examples

To specify a single TLS version:

```
SSL VERSION=TLS protocol version
```

For example:

SSL VERSION=1.2



To specify multiple TLS versions, use the or operator as follows:

```
SSL VERSION=TLS protocol version1 or TLS protocol version2
```

For example:

```
SSL_VERSION=1.1 or 1.2

SSL VERSION=1.0 or 1.1 or 1.2
```

Related Topics

- Set the Required TLS Version on the Server
- Set the Required TLS Version on the Client

5.2.89 TCP.CONNECT_TIMEOUT

Purpose

To specify the time, in ms, sec, or min, for a client to establish a TCP connection (PROTOCOL=tcp in the TNS connect address) to the database server.

Usage Notes

If a TCP connection to the database host is not established in the specified time, then the connection attempt is terminated. The client receives an ORA-12170: TNS:Connect timeout occurred error.

The timeout applies to each IP address that resolves to a host name. It accepts different timeouts with or without space between the value and the unit. For example, if a host name resolves to an IPv6 and an IPv4 address, and if the host is not reachable through the network, then the connection request times out twice because there are two IP addresses. In this example, the default timeout setting of 60 causes a timeout in 120 seconds. In case, no unit is mentioned, the default unit is sec.

Default

60

Example

TCP.CONNECT TIMEOUT=10 ms

5.2.90 TCP.EXCLUDED NODES

Purpose

To specify which clients are denied access to the database.

Usage Notes

This parameter is only valid when the TCP.VALIDNODE_CHECKING parameter is set to yes.

This parameter can use wildcards for IPv4 addresses and CIDR notation for IPv4 and IPv6 addresses.

Syntax

```
TCP.EXCLUDED_NODES=(hostname | ip_address, hostname | ip_address, ...)
```

Example

TCP.EXCLUDED_NODES=(finance.us.example.com, mktg.us.example.com, 192.0.2.25, 172.30.*, 2001:DB8:200C:417A/32)

5.2.91 TCP.INVITED_NODES

Purpose

To specify which clients are allowed access to the database. This list takes precedence over the ${\tt TCP.EXCLUDED_NODES}$ parameter if both lists are present.

Syntax

```
TCP.INVITED_NODES=(hostname | ip_address, hostname | ip_address, ...)
```

Usage Notes

- This parameter is only valid when the TCP.VALIDNODE_CHECKING parameter is set to yes.
- This parameter can use wildcards for IPv4 addresses and CIDR notation for IPv4 and IPv6 addresses.

Example

```
TCP.INVITED_NODES=(sales.us.example.com, hr.us.example.com, 192.0.*, 2001:DB8:200C:433B/32)
```

5.2.92 TCP.NODELAY

Purpose

To preempt delays in buffer flushing within the TCP/IP protocol stack.

Default

yes

Values

yes | no

Example

TCP.NODELAY=yes

5.2.93 TCP.QUEUESIZE

Purpose

To configure the maximum length of the queue for pending connections on a TCP listening socket.

Default

System-defined maximum value. The defined maximum value for Linux is 128.

Values

Any integer value up to the system-defined maximum.

Examples

TCP.QUEUESIZE=100

5.2.94 TCP.VALIDNODE_CHECKING

Purpose

To enable and disable valid node checking for incoming connections.

Usage Notes

If this parameter is set to yes, then incoming connections are allowed only if they originate from a node that conforms to list specified by TCP.INVITED_NODES or TCP.EXCLUDED_NODES parameters.

The TCP.INVITED_NODES and TCP.EXCLUDED_NODES parameters are valid only when the TCP.VALIDNODE_CHECKING parameter is set to yes.

This parameter and the depending parameters, TCP.INVITED_NODES and TCP.EXCLUDED_NODES must be set in the sqlnet.ora file of the listener. This is important in an Oracle RAC environment where the listener runs out of the Oracle Grid Infrastructure home. Setting the parameter in the database home does not have any effect in Oracle RAC environments. In such environments, the address of all Single Client Access Name (SCANs), Virtual IPs (VIPs), local IP must be included in the TCP.INVITED_NODES list.

In VLAN environments, the sqlnet.ora file present in the Oracle Grid Infrastructure home should include all the addresses of all the VLANs. The VLANs perform the network segregation, whereas the INVITED_NODES allows or restricts access to databases within the VLANs.

If multiple databases within the same VLAN require different INVITED_NODE lists, then separate listeners are required.

Default

no

Values

yes | no

Example

TCP.VALIDNODE CHECKING=yes



5.2.95 TNSPING.TRACE_DIRECTORY

Purpose

To specify the destination directory for the TNSPING utility trace file, tnsping.trc.

Default

The ORACLE HOME/network/trace directory.

Example

TNSPING.TRACE DIRECTORY=/oracle/traces

5.2.96 TNSPING.TRACE_LEVEL

Purpose

To turn TNSPING utility tracing on at a specified level or to turn it off.

Default

off

Values

- off for no trace output
- user for user trace information
- admin for administration trace information
- support for Oracle Support Services trace information

Example

TNSPING.TRACE LEVEL=admin

5.2.97 TOKEN AUTH

Use the TOKEN_AUTH parameter to configure token-based authentication for Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM) or Microsoft Azure users of Microsoft Entra ID (previously called Microsoft Azure Active Directory).

Purpose

Token-based access enforces strong authentication, which enables a more secure access to the database. IAM users can connect to OCI Database as a Service (DBaaS) databases, and Azure users can connect to Oracle Databases (cloud or on-premises).

With this setting, the database client looks for a token file when a / (slash) login is used.

Use this parameter under the SECURITY section of the tnsnames.ora file, sqlnet.ora file, or directly as part of the command-line connect string. The parameter value specified in the connect string takes precedence over the other specified values.



Usage Notes for IAM

OCI IAM token:

An OCI IAM token (db-token), which is obtained from IAM using Oracle Cloud Infrastructure (OCI) Command Line Interface (CLI) or programmatically from the OCI Software Development Kit (SDK), is a proof-of-possession (PoP) token with an expiration time and scope.

You can use one of the IAM user credentials, such as API-key, security token, resource principal, instance principal, or delegation token to retrieve the db-token and private key from IAM.

These tokens are transmitted over secure channels. You must use only the TCP/IP with Transport Layer Security (TLS) protocol, otherwise an error message appears indicating that non-TLS connections are disallowed.

Required setting for token-based authentication:

You must configure the TCPS protocol (PROTOCOL=tcps) and set the SSL_SERVER_DN_MATCH parameter to TRUE for token-based authentication.

Use a file location to send the token to Oracle Database:

When an IAM user logs in using <code>/@connect_identifier</code> (and <code>TOKEN_AUTH</code> is set to <code>OCI_TOKEN</code>), the <code>TOKEN_AUTH=OCI_TOKEN</code> setting along with <code>/@connect_identifier</code> instructs the database client to get the <code>db-token</code> and private key from either the default directory or the location specified by <code>TOKEN_LOCATION</code> (using IAM token-based authentication).

Use the client API to send the token to Oracle Database:

If your client application is updated to retrieve tokens from IAM, then you can override the <code>TOKEN_AUTH=OCI_TOKEN</code> setting. The client application gets the <code>db-token</code> and private key from IAM and sends as attributes to the database client using the client API. In this case, you do not need to specify the <code>TOKEN_AUTH</code> and <code>TOKEN_LOCATION</code> parameters.

General IAM token-based authentication process:

 An IAM user or application in OCI first requests the db-token from IAM by using APIkey, security token, resource principal, service principal, instance principal, or delegation token (delegation token is available only in the Cloud Shell).

To use a security token, you need to generate it by completing the browser authentication process and then request the <code>db-token</code> using that security token. If the IAM policy that authorizes you to be issued the <code>db-token</code> exists, then the <code>db-token</code> is returned.

You request the db-token using OCI CLI (or OCI SDK for applications). For example, run the following OCI CLI command to request the db-token by using an API-key (apikey):

```
$ oci iam db-token get --profile scott
```

The profile option specifies the profile for which you want to access the IAM user credentials and retrieve the db-token.

For more information on using OCI CLI, see the get command details in *Oracle Cloud Infrastructure CLI Command Reference*.

- 2. OCI CLI references the config file (that stores your IAM user credentials as part of the profile) and makes a call to IAM to get the db-token. The db-token and private key files are written at the default or specified token location.
- 3. You can specify the TOKEN_LOCATION parameter to override the default directory where the db-token and private key files are stored.

The database client gets the <code>db-token</code> and private key from the default token location or the location specified by <code>TOKEN_LOCATION</code>, signs the <code>db-token</code> with the private key and sends it to the database server. The database server verifies the <code>db-token</code> and gets the group membership information for the user. If the IAM user is mapped to a database schema (exclusively or shared), then the login is completed.

Note:

You can also use another IAM credential, IAM database password, to request the dbtoken from IAM. This db-token is a bearer token and does not come with a private key. You can configure the database client to request this token using your IAM user name and IAM database password. An application cannot pass this type of db-token to the client. In this case, you use a different parameter setting (PASSWORD AUTH=OCI TOKEN).

Unlike the API-key, security token, resource principal, service principal, instance principal, and delegation token that require an application or tool to get a token, the IAM database password can only be used by the database client to retrieve the token. See PASSWORD AUTH.



Table 5-2 Values and Examples for IAM

Default	Value	Example
None	TOKEN_AUTH=OCI_TOKEN	In the tnsnames.ora file:
		<pre>net_service_name= (DESCRIPTION = (ADDRESS=(PROTOCOL=tcps)(HOST=sales-svr)(PORT=1521)) (SECURITY= (SSL_SERVER_DN_MATCH=TRUE)</pre>
		<pre>(SSL_SERVER_CERT_DN="C=US,O=example,CN=Oracle Context")</pre>
		<pre>(CONNECT_DATA=(SERVICE_NAME=sales.us.example. com)))</pre>
		In the sqlnet.ora file:
		SSL_SERVER_DN_MATCH=TRUE TOKEN_AUTH=OCI_TOKEN
		In these examples, the optional TOKEN_LOCATION parameter is not specified. Thus, the client automatically gets the dbtoken and private key from the default token location.

Usage Notes for Entra ID

Entra ID access token:

An Entra ID OAuth2 access token is a bearer token with an expiration time and scope. This token follows the OAuth2.0 standard with Entra ID extensions. You can request these tokens from tools and scripts run on Linux, Microsoft PowerShell, or other environments. You can also request these tokens programmatically using the Microsoft SDKs.

These tokens are transmitted over secure channels. You must use only the TCP/IP with Transport Layer Security (TLS) protocol, otherwise an error message appears indicating that non-TLS connections are disallowed.

Required setting for token-based authentication:

You must configure the TCPS protocol (PROTOCOL=tcps) and set the SSL_SERVER_DN_MATCH parameter to TRUE for token-based authentication.

Use a file location to send the token to Oracle Database:

When an Azure user logs in using $/@connect_identifier$, then the <code>TOKEN_AUTH=OAUTH</code> setting instructs the database client to get the access token from the directory location specified by <code>TOKEN_LOCATION</code> if the token file is named <code>token</code>. If the token file name is different from <code>token</code>, then you must use the file name along with the directory location while specifying the <code>TOKEN_LOCATION</code> parameter.

The TOKEN_LOCATION parameter is mandatory for Azure token-based authentication. The database client gets the token from this location and sends it to the database server.

Use the client API to send the token to Oracle Database:

If your client application is updated to retrieve tokens from Entra ID, then you can override the <code>TOKEN_AUTH=OAUTH</code> setting. Entra ID directly passes the <code>db-token</code> as an attribute to the database client using the client API. When the client receives this request, the client sends it to the database server.

In this case, you do no need to specify the TOKEN AUTH and TOKEN LOCATION parameters.

General Azure token-based authentication process:

1. An Azure user or application first requests the access token from Entra ID using one of the supported authentication flows (resource owner password credentials, authorization code, on-behalf-of (OBO) flow, or client credentials).

An Azure user can connect using any supported utility to retrieve the token and store it in a local file directory.

You can request the token from tools and scripts run on Linux, Microsoft PowerShell, or other environments. You can also request programmatically using the Microsoft SDKs.

For detailed examples on how to retrieve an Entra ID OAuth2 access token, see *Oracle Database Security Guide*.

2. The database client then sends the token to the database server. The database server verifies the token (using the Entra ID public key) and extracts various claims from the token, including user name, app roles, and audience. If the Entra ID principal is mapped to a database schema (exclusively or shared), then the login is completed.



Table 5-3 Values and Examples for Entra ID

Default	Value	Example
None	If the token file is named token:	In the tnsnames.ora file:
	TOKEN_AUTH=OAUTH TOKEN_LOCATION="token_f ile_directory"	<pre>net_service_name= (DESCRIPTION= (ADDRESS=(PROTOCOL=tcps) (HOST=salesserver1) (PORT=1522)) (SECURITY= (SSL_SERVER_DN_MATCH=TRUE)</pre>
		<pre>(SSL_SERVER_CERT_DN="C=US,O=example,CN=Oracle Context")</pre>
		In the sqlnet.ora file:
		SSL_SERVER_DN_MATCH=TRUE TOKEN_AUTH=OAUTH TOKEN_LOCATION="/home/dbuser1/access-token"
		In these examples, the token file name is token. Thus, only the directory path (/home/dbuser1/access-token) is specified. The client automatically looks for the token file in the specified path and gets the access token.



Table 5-3 (Cont.) Values and Examples for Entra ID

Default	Value	Example
	If the token file name is different from token:	In the tnsnames.ora file:
	TOKEN_AUTH=OAUTH TOKEN_LOCATION="token_f ile_directory/ token_filename"	<pre>net_service_name= (DESCRIPTION= (ADDRESS=(PROTOCOL=tcps) (HOST=salesserver1) (PORT=1522)) (SECURITY= (SSL_SERVER_DN_MATCH=TRUE)</pre>
		<pre>(SSL_SERVER_CERT_DN="C=US,O=example,CN=Oracle Context")</pre>
		In the sqlnet.ora file:
		SSL_SERVER_DN_MATCH=TRUE TOKEN_AUTH=OAUTH TOKEN_LOCATION="/home/dbuser1/access-token/ mytoken"
		In these examples, the token file name is mytoken. Thus, both the file name and directory path (/home/dbuser1/access-token) are specified. The client gets the access token from the mytoken file in the specified path.

Related Topics

- Authenticating and Authorizing IAM Users for Oracle DBaaS Databases
- Authenticating and Authorizing Microsoft Azure Users for Oracle Databases
- TOKEN_LOCATION
 Use the TOKEN_LOCATION parameter to specify the directory location where token file is stored for token-based authentication.

5.2.98 TOKEN_LOCATION

Use the ${\tt TOKEN_LOCATION}$ parameter to specify the directory location where token file is stored for token-based authentication.

Purpose

To specify the token file directory location. You use this parameter while configuring tokenbased authentication for Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM) or Microsoft Azure users of Microsoft Entra ID. The database client gets the token from this location and sends it to the database server. For Entra ID, you can also specify the token file name along with the directory location.

Use this parameter along with the <code>TOKEN_AUTH</code> parameter in the <code>tnsnames.ora</code> file, <code>sqlnet.ora</code> file, or directly as part of the command-line connect string. The parameter values specified in the connect string take precedence over the other specified values.

Usage Notes for IAM

The TOKEN_LOCATION parameter is optional for IAM token-based authentication. You can use this parameter along with the TOKEN_AUTH parameter to override the default directory where the db-token and private key are stored. This location is used by the database client to retrieve the db-token and private key.

When an IAM user initiates a connection using <code>/@connect_identifier</code> (and <code>TOKEN_AUTH</code> is set to <code>OCI_TOKEN</code>), the database client retrieves the <code>db-token</code> and private key from either the default directory or the location specified by <code>TOKEN_LOCATION</code>. The client then signs the <code>db-token</code> using the private key and sends the <code>db-token</code> to the database server.

Default Setting for IAM

On Linux:

/home/username/.oci/db-token

On Windows:

The database client searches for the default directory in this order:

If the USERPROFILE environment variable is set, then the client searches in the USERPROFILE directory (for example, C:\Users\username).

If USERPROFILE is not set, then the client searches in HOMEDRIVE directory (for example, C:) with HOMEPATH (for example, \Users\username).

For example, the default token location directory on Windows is:

C:\Users\username\.oci\db-token



Values and Examples for IAM

Value	Example
TOKEN_LOCATION="token_file_directory"	In the tnsnames.ora file:
	<pre>net_service_name= (DESCRIPTION = (ADDRESS=(PROTOCOL=tcps) (HOST=sales-svr) (PORT=1521)) (SECURITY= (SSL_SERVER_DN_MATCH=TRUE)</pre>
	<pre>(SSL_SERVER_CERT_DN="C=US,O=example,CN=OracleContext")</pre>
	<pre>(CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)))</pre>
	In the sqlnet.ora file:
	SSL_SERVER_DN_MATCH=TRUE TOKEN_AUTH=OCI_TOKEN TOKEN_LOCATION="/home/oracle/.oci/db-token"

Usage Notes for Entra ID

The TOKEN_LOCATION parameter is mandatory for Azure token-based authentication. You must use this parameter along with the TOKEN_AUTH parameter to specify the directory location where the Entra ID OAuth2 access token is stored. This location is used by the database client to get the access token.

If your token file is named token, then specify only the directory path. If the token file name is different from token, then you must use the file name along with the directory path.

When an Azure user initiates a connection using $/@connect_identifier$, the database client retrieves the access token from the location specified by <code>TOKEN_LOCATION</code> and sends the token to the database server.

Default Setting for Entra ID

None

Values and Examples for Entra ID

Value	Example
If the token file is named token:	In the tnsnames.ora file:
TOKEN LOCATION="token file directory"	
	<pre>net_service_name=</pre>
	(DESCRIPTION=
	(ADDRESS=(PROTOCOL=tcps)(HOST=salesserver1)
	(PORT=1522))
	(SECURITY=
	(SSL_SERVER_DN_MATCH=TRUE)
	(SSL_SERVER_CERT_DN="C=US,O=example,CN=OracleContext") (TOKEN AUTH=OAUTH) (TOKEN LOCATION="/home/
	dbuser1/access-token"))
	<pre>(CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)))</pre>
	In the sqlnet.ora file:
	SSL_SERVER_DN_MATCH=TRUE
	TOKEN_AUTH=OAUTH
	TOKEN_LOCATION="/home/dbuser1/access-token"
	In these examples, the token file name is token. Thus, only the directory path (/home/dbuser1/access-token) is specified. The client automatically looks for the token file in the specified path and gets the access token.



Value	Example
If the token file name is different from token:	In the tnsnames.ora file:
TOKEN_LOCATION="token_file_directory/token_filename"	<pre>net_service_name= (DESCRIPTION=</pre>
	(SECURITY= (SSL_SERVER_DN_MATCH=ON)
	<pre>(SSL_SERVER_CERT_DN="C=US,O=example,CN=OracleContext")</pre>
	In the sqlnet.ora file:
	SSL_SERVER_DN_MATCH=TRUE TOKEN_AUTH=OAUTH TOKEN_LOCATION="/home/dbuser1/access-token/mytoken"
	In these examples, the token file name is mytoken. Thus, both the file name and directory path (/home/dbuser1/access-token) are specified. The client gets the access token from the mytoken file in the specified path.

Related Topics

- Authenticating and Authorizing IAM Users for Oracle DBaaS Databases
- Authenticating and Authorizing Microsoft Azure Users for Oracle Databases
- TOKEN AUTH

Use the <code>TOKEN_AUTH</code> parameter to configure token-based authentication for Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM) or Microsoft Azure users of Microsoft Entra ID (previously called Microsoft Azure Active Directory).

5.2.99 USE_CMAN

Purpose

To specify client routing to Oracle Connection Manager.

Usage Notes

If set to true, then the parameter routes the client to a protocol address for Oracle Connection Manager.

If set to false, then the client picks one of the address lists at random and fails over to the other address list if the chosen ADDRESS_LIST fails. With USE_CMAN=true, the client always uses the first address list.

If no Oracle Connection Manager addresses are available, then connections are routed through any available listener address.

Default

false

Values

true | false

Example

USE_CMAN=true

5.2.100 USE_DEDICATED_SERVER

Purpose

To append (SERVER=dedicated) to the CONNECT_DATA section of the connect descriptor used by the client.

Usage Notes

It overrides the current value of the SERVER parameter in the tnsnames.ora file.

If set to on, then the parameter <code>USE_DEDICATED_SERVER</code> automatically appends (<code>SERVER=dedicated</code>) to the connect data for a connect descriptor. This way connections from this client use a dedicated server process, even if shared server is configured.

Default

off

Values

- on to append (SERVER=dedicated)
- off to send requests to existing server processes

Example

USE DEDICATED SERVER=on



Oracle Database Net Services Administrator's Guide for complete configuration information



5.2.101 WALLET_LOCATION

Use the sqlnet.ora parameter WALLET LOCATION to specify the location of Oracle wallets.

Purpose

To specify the directory path where you want to create and store an Oracle wallet. Wallets securely contain certificates, secrets, private keys, and trust points used by Oracle Database.

Usage Notes

• The password-protected wallet is stored in an ewallet.p12 file. The auto-login and local auto-login wallets are stored in a cwallet.sso file.

For example, if an Oracle wallet is stored in the Microsoft Windows registry and the wallet's key (KEY) is SALESAPP, then the storage location of the password-protected wallet is HKEY_CURRENT_USER\SOFTWARE\ORACLE\WALLETS\SALESAPP\EWALLET.P12. The storage location of the auto-login and local auto-login wallets is HKEY_CURRENT_USER\SOFTWARE\ORACLE\WALLETS\SALESAPP\CWALLET.SSO.

• The WALLET_LOCATION parameter is optional for TLS connections that do not use a client wallet. If WALLET_LOCATION is not included in sqlnet.ora or connect string, then the driver automatically picks up common root certificates from the client system's default certificate store (if the system is Linux or Windows). In this case, the database server certificate needs to be signed by a trusted root certificate that is already installed in the default certificate store. The default certificate store is located in /etc/pki/tls/cert.pem on Linux and Microsoft Certificate Store (MCS) on Windows.

If WALLET_LOCATION is set in sqlnet.ora for all connections, then you can override this setting for a particular connection that does not need a client wallet by using WALLET_LOCATION=SYSTEM in the connect string (in tnsnames.ora or directly to the command line). The database client then uses common root certificates from the default certificate store (instead of the client wallet) to validate the database server certificate.

Additional Parameters

WALLET LOCATION supports the following parameters:

- SOURCE: Type of storage for wallets (METHOD) and storage location (METHOD DATA)
- METHOD: Type of storage
- METHOD DATA: Storage location
- DIRECTORY: Location of wallet on the file system
- KEY: Wallet type and location in the Microsoft Windows registry
- PROFILE: Entrust profile file (.epf)
- INIFILE: Entrust initialization file (.ini)



Syntax and Examples

The syntax depends on the wallet as follows:

Wallet on the file system:

Microsoft certificate store:

```
WALLET_LOCATION=
(SOURCE=
(METHOD=mcs))
```

The key-value pair for MCS omits the METHOD_DATA parameter because MCS does not use wallets. Instead, Oracle PKI (public key infrastructure) applications obtain certificates, trust points and private keys directly from a user's profile.

Wallet in the Microsoft Windows registry:

```
WALLET_LOCATION=
(SOURCE=
(METHOD=reg)
(METHOD_DATA=
(KEY=registry_key)))
```

For example:

```
WALLET_LOCATION=
(SOURCE=
(METHOD=REG)
(METHOD_DATA=
(KEY=SALESAPP)))
```

Entrust wallets:

For example:



Default

None

Related Topics

- Oracle Database Enterprise User Security Administrator's Guide
- Oracle Database Security Guide

5.3 ADR Diagnostic Parameters in sqlnet.ora

Diagnostic data for critical errors is stored in the sqlnet.ora Automatic Diagnostic Repository (ADR).

About ADR Diagnostic Parameters

You can use Automatic Diagnostic Repository (ADR) diagnostic parameters when ADR is enabled, which is the default. Oracle ignores non-ADR parameters in the sqlnet.ora file when you enable ADR.

ADR BASE

It is a diagnostic parameter in the sqlnet.ora file and it specifies the base location of the ADR files.

DIAG ADR ENABLED

DIAG_ADR_ENABLED diagnostic parameter of the sqlnet.ora file specifies whether ADR tracing is enabled.

TRACE LEVEL CLIENT

The TRACE_LEVEL_CLIENT diagnostic parameter of the sqlnet.ora file turns client tracing on or off at a specified level.

TRACE_LEVEL_SERVER

The TRACE_LEVEL_SERVER diagnostic parameter of the sqlnet.ora file turns server tracing on or off at a specified level.

TRACE_TIMESTAMP_CLIENT

The TRACE_TIMESTAMP_CLIENT diagnostic parameter of the sqlnet.ora file adds a time stamp to every trace event in the client trace file.

TRACE TIMESTAMP SERVER

The TRACE_TIMESTAMP_CLIENT diagnostic parameter of the sqlnet.ora file adds a time stamp to every trace event in the database server trace file.

5.3.1 About ADR Diagnostic Parameters

You can use Automatic Diagnostic Repository (ADR) diagnostic parameters when ADR is enabled, which is the default. Oracle ignores non-ADR parameters in the sqlnet.ora file when you enable ADR.

Since Oracle Database 11g, Oracle Database includes an advanced fault diagnostic infrastructure to prevent, detect, diagnose, and resolve problems. The problems might be critical errors such as those that are caused by database code bugs, metadata corruption, or customer data corruption.

When critical errors occur, they are assigned incident numbers. Diagnostic data for the errors, such as traces and dumps, are captured and tagged with the incident number. The data is then stored in ADR, which is a file-based repository outside the database.

The following sqlnet.ora parameters are used when you enable ADR (when DIAG ADR ENABLED is set to on):

5.3.2 ADR_BASE

It is a diagnostic parameter in the sqlnet.ora file and it specifies the base location of the ADR files.

Purpose

To specify the base directory into which tracing and logging incidents are stored when ADR is enabled.

Usage Notes

This parameter is applicable only to clients. On the server side, the ADR base location is defined by the <code>DIAGNOSTIC_DEST</code> initialization parameter in the <code>init.ora</code> file. See <code>DIAGNOSTIC_DEST</code> in *Oracle Database Reference*.

Default

ORACLE BASE or ORACLE HOME/log (if ORACLE BASE is not defined)

Values

Any valid directory path to a directory with write permission.

Example

ADR BASE=/oracle/network/trace

5.3.3 DIAG_ADR_ENABLED

DIAG_ADR_ENABLED diagnostic parameter of the sqlnet.ora file specifies whether ADR tracing is enabled.

Purpose

To specify whether ADR tracing is enabled.

Usage Notes

If the DIAG ADR ENABLED parameter is set to OFF, then non-ADR file tracing is used.

Default

on

Values

on | off

Example 5-7 Example

DIAG ADR ENABLED=on



5.3.4 TRACE_LEVEL_CLIENT

The TRACE_LEVEL_CLIENT diagnostic parameter of the sqlnet.ora file turns client tracing on or off at a specified level.

Purpose

To turn client tracing on at a specified level or to turn it off.

Usage Notes

This parameter is also applicable when non-ADR tracing is used.

Default

off or 0

Values

- off or 0 for no trace output
- user or 4 for user trace information
- admin or 10 for administration trace information
- support or 16 for Oracle Support Services trace information

Example

TRACE_LEVEL_CLIENT=user

5.3.5 TRACE_LEVEL_SERVER

The TRACE_LEVEL_SERVER diagnostic parameter of the sqlnet.ora file turns server tracing on or off at a specified level.

Purpose

To turn server tracing on at a specified level or to turn it off.

Usage Notes

This parameter is also applicable when non-ADR tracing is used.

Default

off or 0

Values

- off or 0 for no trace output
- user or 4 for user trace information
- admin or 10 for administration trace information
- support or 16 for Oracle Support Services trace information



Example

TRACE_LEVEL_SERVER=admin

5.3.6 TRACE_TIMESTAMP_CLIENT

The TRACE_TIMESTAMP_CLIENT diagnostic parameter of the sqlnet.ora file adds a time stamp to every trace event in the client trace file.

Purpose

To add a time stamp in the form of dd-mmm-yyyy hh:mm:ss:mil to every trace event in the client trace file, which has a default name of sqlnet.trc.

Usage Notes

This parameter is also applicable when non-ADR tracing is used.

Default

on

Values

on **Or** true | off **Or** false

Example

TRACE TIMESTAMP CLIENT=true

5.3.7 TRACE_TIMESTAMP_SERVER

The TRACE_TIMESTAMP_CLIENT diagnostic parameter of the sqlnet.ora file adds a time stamp to every trace event in the database server trace file.

Purpose

To add a time stamp in the form of dd-mmm-yyyy hh:mm:ss:mil to every trace event in the database server trace file, which has a default name of svr pid.trc.

Usage Notes

This parameter is also applicable when non-ADR tracing is used.

Default

on

Values

on **or** true | off **or** false

Example

TRACE_TIMESTAMP_SERVER=true



5.4 Non-ADR Diagnostic Parameters in sqlnet.ora Files

Learn about sqlnet.ora parameters that you use when you disable ADR.

This section lists the sqlnet.ora parameters that are used when you disable ADR.

Note:

The default value of <code>DIAG_ADR_ENABLED</code> is on. Therefore, the <code>DIAG_ADR_ENABLED</code> parameter must explicitly be set to <code>off</code> to use non-ADR tracing.

LOG DIRECTORY CLIENT

The $LOG_DIRECTORY_CLIENT$ non-ADR diagnostic parameter of the sqlnet.ora file specifies the destination directory for the client log file.

- LOG_DIRECTORY_SERVER
- LOG_FILE_CLIENT
- LOG_FILE_SERVER
- TRACE_DIRECTORY_CLIENT
- TRACE_DIRECTORY_SERVER
- TRACE_FILE_CLIENT
- TRACE FILE SERVER
- TRACE_FILEAGE_CLIENT
- TRACE_FILEAGE_SERVER
- TRACE_FILELEN_CLIENT
- TRACE FILELEN SERVER
- TRACE_FILENO_CLIENT
- TRACE_FILENO_SERVER
- TRACE_UNIQUE_CLIENT

5.4.1 LOG_DIRECTORY_CLIENT

The LOG_DIRECTORY_CLIENT non-ADR diagnostic parameter of the sqlnet.ora file specifies the destination directory for the client log file.

Purpose

To specify the destination directory for the client log file.

Usage Notes

Use this parameter when ADR is not enabled.



Default

ORACLE HOME/network/log

Values

Any valid directory path.

Example

LOG DIRECTORY CLIENT=/oracle/network/log

5.4.2 LOG_DIRECTORY_SERVER

Purpose

To specify the destination directory for the database server log file.

Usage Notes

Use this parameter when ADR is not enabled.

Default

ORACLE_HOME/network/trace

Values

Any valid directory path to a directory with write permission.

Example

LOG_DIRECTORY_SERVER=/oracle/network/trace

5.4.3 LOG_FILE_CLIENT

Purpose

To specify the name of the log file for the client.

Usage Notes

Use this parameter when ADR is not enabled.

Default

ORACLE_HOME/network/log/sqlnet.log

Values

The default value cannot be changed.



5.4.4 LOG_FILE_SERVER

Purpose

To specify the name of the log file for the database server.

Usage Notes

Use this parameter when ADR is not enabled.

Default

sqlnet.log

Values

Any valid directory path to a directory with write permission.

Example

LOG FILE SERVER=svr.log

5.4.5 TRACE_DIRECTORY_CLIENT

Purpose

To specify the destination directory for the client trace file.

Usage Notes

Use this parameter when ADR is not enabled.

Default

ORACLE HOME/network/trace

Values

Any valid directory path to a directory with write permission.

Example

TRACE DIRECTORY CLIENT=/oracle/traces

5.4.6 TRACE_DIRECTORY_SERVER

Purpose

To specify the destination directory for the database server trace file. Use this parameter when ADR is not enabled.

Default

ORACLE HOME/network/trace

Values

Any valid directory path to a directory with write permission.



Example

TRACE_DIRECTORY_SERVER=/oracle/traces

5.4.7 TRACE_FILE_CLIENT

Purpose

To specify the name of the client trace file.

Usage Notes

Use this parameter when ADR is not enabled.

Default

ORACLE_HOME/network/trace/cli.trc

Values

Any valid file name.

Example

TRACE FILE CLIENT=clientsqlnet.trc

5.4.8 TRACE_FILE_SERVER

Purpose

To specify the destination directory for the database server trace output.

Usage Notes

Use this parameter when ADR is not enabled.

Default

ORACLE_HOME/network/trace/svr_pid.trc

Values

Any valid file name. The process identifier (pid) is appended to the name automatically.

Example

TRACE_FILE_SERVER=svrsqlnet.trc

5.4.9 TRACE_FILEAGE_CLIENT

Purpose

To specify the maximum age of client trace files in minutes.



Usage Notes

When the age limit is reached, the trace information is written to the next file. The number of files is specified with the TRACE_FILENO_CLIENT parameter. Use this parameter when ADR is not enabled.

Default

Unlimited

This is the same as setting the parameter to 0.

Example 5-8 Example

TRACE FILEAGE CLIENT=60

5.4.10 TRACE FILEAGE SERVER

Purpose

To specify the maximum age of database server trace files in minutes.

Usage Notes

When the age limit is reached, the trace information is written to the next file. The number of files is specified with the TRACE_FILENO_SERVER parameter. Use this parameter when ADR is not enabled.

Default

Unlimited

This is the same as setting the parameter to0.

Example 5-9 Example

TRACE FILEAGE SERVER=60

5.4.11 TRACE_FILELEN_CLIENT

Purpose

To specify the size of the client trace files in kilobytes (KB).

Usage Notes

When the size is met, the trace information is written to the next file. The number of files is specified with the TRACE_FILENO_CLIENT parameter. Use this parameter when ADR is not enabled.

Example

TRACE FILELEN CLIENT=100



5.4.12 TRACE_FILELEN_SERVER

Purpose

To specify the size of the database server trace files in kilobytes (KB).

Usage Notes

When the size is met, the trace information is written to the next file. The number of files is specified with the TRACE_FILENO_SERVER parameter. Use this parameter when ADR is not enabled.

Example

TRACE FILELEN SERVER=100

5.4.13 TRACE_FILENO_CLIENT

Purpose

To specify the number of trace files for client tracing.

Usage Notes

When this parameter is set with the TRACE_FILELEN_CLIENT parameter, trace files are used in a cyclical fashion. The first file is filled first, then the second file, and so on. When the last file has been filled, then the first file is re-used, and so on.

When this parameter is set with the TRACE_FILEAGE_CLIENT parameter, trace files are cycled based on their age. The first file is used until the age limit is reached, then the second file is used, and so on. When the last file's age limit is reached, the first file is re-used, and so on

When this parameter is set with both the TRACE_FILELEN_CLIENT and TRACE_FILEAGE_CLIENT parameters, trace files are cycled when either the size limit or the age limit is reached.

The trace file names are distinguished from one another by their sequence number. For example, if the default trace file of sqlnet.trc is used, and this parameter is set to 3, then the trace files would be named sqlnet1.trc, sqlnet2.trc and sqlnet3.trc.

In addition, trace events in the trace files are preceded by the sequence number of the file. Use this parameter when ADR is not enabled.

Default

None

Example

TRACE FILENO CLIENT=3

5.4.14 TRACE_FILENO_SERVER

Purpose

To specify the number of trace files for database server tracing.



Usage Notes

When this parameter is set with the TRACE_FILELEN_SERVER parameter, trace files are used in a cyclical fashion. The first file is filled first, then the second file, and so on. When the last file has been filled, then the first file is re-used, and so on.

When this parameter is set with the TRACE_FILEAGE_SERVER parameter, trace files are cycled based on the age of the trace file. The first file is used until the age limit is reached, then the second file is used, and so on. When the last file's age limit is reached, the first file is reused, and so on.

When this parameter is set with both the TRACE_FILELEN_SERVER and TRACE_FILEAGE_SERVER parameters, trace files are cycled when either the size limit or the age limit is reached.

The trace file names are distinguished from one another by their sequence number. For example, if the default trace file of svr_pid.trc is used, and this parameter is set to 3, then the trace files would be named svr1 pid.trc, svr2 pid.trc and svr3 pid.trc.

In addition, trace events in the trace files are preceded by the sequence number of the file. Use this parameter when ADR is not enabled.

Default

None

Example

TRACE FILENO SERVER=3

5.4.15 TRACE_UNIQUE_CLIENT

Purpose

To specify whether a unique trace file is created for each client trace session.

Usage Notes

When the value is set to on, a process identifier is appended to the name of each trace file, enabling several files to coexist. For example, trace files named sqlnetpid.trc are created if default trace file name sqlnet.trc is used. When the value is set to off, data from a new client trace session overwrites the existing file. Use this parameter when ADR is not enabled.

Default

on

Values

on or off

Example

TRACE_UNIQUE_CLIENT=on



Local Naming Parameters in the tnsnames.ora File

Learn about the Oracle Net service names local naming parameters that you specify in the tnsnames.ora configuration file.

Overview of Local Naming Parameters

The tnsnames.ora file is a configuration file that contains network service names that are mapped to connect descriptors for the Local Naming method.

General Syntax of thsnames.ora

In this example, you can see the general tnsnames.ora file syntax.

Using Multiple Descriptions in tnsnames.ora Files

In this example, you can see two connect descriptors with multiple addresses.

Multiple Address Lists in tnsnames.ora Files

The tnsnames.ora file supports connect descriptors with multiple lists of addresses, each with its own characteristics. Learn how to configure multiple address lists in tnsnames.ora files.

Connect-Time Failover and Client Load Balancing with Oracle Connection Managers

When tnsnames.ora connect descriptors have at least two protocol addresses for Oracle Connection Manager, you can also include parameters for connect-time failover and load balancing in the file.

Connect Descriptor Descriptions

Specify connect descriptors using the DESCRIPTION parameter. Identify multiple connect descriptors with the DESCRIPTION LIST parameter.

Protocol Addresses

The protocol address section of a tnsnames.ora file specifies listener protocol addresses.

Optional Parameters for Description

Use these parameters to configure connect descriptor containers under the DESCRIPTION parameter.

Connection Data Section

Learn how to configure network connections with protocol addresses.

Security Section

The security section of the tnsnames.ora file specifies these security-related parameters for use with Oracle security features.

Timeout Parameters

The timeout section of the tnsnames.ora file provides the ability to specify timeout and retry configuration through the TNS connect string.

Compression Parameters

The compression section of the tnsnames.ora file provides the ability to enable compression and specify compression levels. These parameters can be set at the DESCRIPTION level of a connect string.

6.1 Overview of Local Naming Parameters

The tnsnames.ora file is a configuration file that contains network service names that are mapped to connect descriptors for the Local Naming method.

A net service name is an alias that is mapped to a database network address that is contained in a connect descriptor. A connect descriptor contains the location of the listener that is accessed through a protocol address and the service name of the database to which to connect. Clients and database servers that are clients of other database servers use the net service name when connecting with applications.

Typically, tools such as Oracle Database Configuration Assistant (DBCA) and Oracle Net Configuration Assistant (NETCA) create the thinnames.ora file in the <code>ORACLE_HOME/network/admin</code> directory for Oracle Database installations, the <code>GRID_HOME/network/admin</code> directory for Oracle Grid Infrastructure installations, or the corresponding <code>ORACLE_BASE_HOME/network/admin</code> directory for a read-only Oracle home. Note that if you have installed multiple databases, then the file is created in the Oracle home or Grid home where DBCA or NETCA is run (or the Oracle base home for read-only instances).

The order for checking the tnsnames.ora file is:

- 1. The directory specified by the TNS ADMIN environment variable
- 2. If the TNS_ADMIN environment variable is not set or the file is not found in the TNS_ADMIN directory:
 - On Linux and UNIX: The ORACLE_HOME/network/admin directory (or its ORACLE BASE HOME/network/admin directory for a read-only Oracle home)
 - On Windows: The ORACLE_HOME\network\admin directory (or its
 ORACLE BASE HOME\network\admin directory for a read-only Oracle home)
- 3. For a read-only Oracle home, if the file is not found in the Oracle base home:
 - On Linux and UNIX: The ORACLE HOME/network/admin directory
 - On Windows: The ORACLE HOME\network\admin directory

Note:

- On Windows, the <code>ORACLE_HOME</code> location is determined by the <code>ORACLE_HOME\bin\oracle.key</code> file (which contains the name of the Windows Registry key where <code>ORACLE_HOME</code> is defined). Also, the <code>TNS_ADMIN</code> environment variable is used if it is set in the environment of the process. If you do not define the <code>TNS_ADMIN</code> environment variable in the environment or if the process is a service that does not have an environment, then Windows scans the registry for a <code>TNS_ADMIN</code> parameter.
- With Oracle Instant Client, tnsnames.ora is located in the subdirectory of the Oracle Instant Client software. For example, in the /opt/oracle/ instantclient release number/network/admin directory.

Related Topics

Oracle Database Net Services Administrator's Guide



6.2 General Syntax of tnsnames.ora

In this example, you can see the general tnsnames.ora file syntax.

Here, DESCRIPTION contains the connect descriptor, ADDRESS contains the protocol address, and CONNECT DATA contains database service identification information.

Example 6-1 Basic Format of thsnames.ora File

```
net_service_name=
  (DESCRIPTION=
        (ADDRESS=(protocol_address_information))
        (CONNECT_DATA=
                (SERVICE_NAME=service_name)
        )
        )
}
```

6.3 Using Multiple Descriptions in tnsnames.ora Files

In this example, you can see two connect descriptors with multiple addresses.

A tnsnames.ora file can contain net service names with one or more connect descriptors. Each connect descriptor can contain one or more protocol addresses.

Use the tnsnames.ora parameter DESCRIPTION LIST to define the list of connect descriptors.

Example 6-2 Net Service Name with Multiple Connect Descriptors in tnsnames.ora

Note:

Oracle Net Manager does not support multiple connect descriptors for a net service name if you use Oracle Connection Manager.

6.4 Multiple Address Lists in tnsnames.ora Files

The tnsnames.ora file supports connect descriptors with multiple lists of addresses, each with its own characteristics. Learn how to configure multiple address lists in tnsnames.ora files.

The following example shows two address lists. The first address list features client load balancing and no connect-time failover. These setting apply only to protocol addresses that are within its <code>ADDRESS_LIST</code>. The second protocol address list does not enable client load loading balancing, but the list does enable connect-time failover. These settings affect only protocol addresses that are included in its <code>ADDRESS_LIST</code>. The client first tries the first or second protocol address at random, then it tries protocol addresses number three and four, in that order, and so on.

Example 6-3 Multiple Address Lists in thsnames.ora Files

Note:

- Oracle Net Manager supports only the creation of one protocol address list for a connect descriptor.
- Oracle Net Services supports the IFILE parameter in the tnsnames.ora file, with up to three levels of nesting. You must add the parameter manually to the file. The following is an example of the syntax:

```
IFILE=/tmp/listener_em.ora
IFILE=/tmp/listener_cust1.ora
IFILE=/tmp/listener_cust2.ora
```

6.5 Connect-Time Failover and Client Load Balancing with Oracle Connection Managers

When tnsnames.ora connect descriptors have at least two protocol addresses for Oracle Connection Manager, you can also include parameters for connect-time failover and load balancing in the file.

Example 6-4 Multiple Oracle Connection Manager Addresses in tnsnames.ora

This example illustrates the failover of multiple Oracle Connection Manager protocol addresses.

```
sample1=
  (DESCRIPTION=
      (SOURCE_ROUTE=yes)
      (ADDRESS_LIST=
```



```
(ADDRESS=(PROTOCOL=tcp) (HOST=host1) (PORT=1630)) # 1
(ADDRESS_LIST=
    (FAILOVER=on)
    (LOAD_BALANCE=off) # 2
    (ADDRESS=(PROTOCOL=tcp) (HOST=host2a) (PORT=1630))
    (ADDRESS=(PROTOCOL=tcp) (HOST=host2b) (PORT=1630)))
    (ADDRESS=(PROTOCOL=tcp) (HOST=host3) (PORT=1521))) # 3
(CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)))
```

The previous syntax does the following:

1. The client connects to the protocol address of the first Oracle Connection Manager as indicated by:

```
(ADDRESS=(PROTOCOL=tcp) (HOST=host1) (PORT=1630))
```

2. Oracle Connection Manager connects to the first protocol address of another Oracle Connection Manager. If the first protocol address fails, then it tries to connect to the second protocol address. This sequence is specified with the following configuration:

```
(ADDRESS_LIST=
  (FAILOVER=on)
  (LOAD_BALANCE=off)
  (ADDRESS=(PROTOCOL=tcp) (HOST=host2a) (PORT=1630))
  (ADDRESS=(PROTOCOL=tcp) (HOST=host2b) (PORT=1630)))
```

Oracle Connection Manager connects to the database service using the following protocol address:

```
(ADDRESS=(PROTOCOL=tcp) (HOST=host3) (PORT=1521))
```

Example 6-5 Client Load Balancing in the three controls of the control of the con

This example illustrates client load balancing among two Oracle Connection Managers and two protocol addresses:

```
sample2=
(DESCRIPTION=
  (LOAD_BALANCE=on) # 1
  (FAILOVER=on)
  (ADDRESS_LIST=
      (SOURCE_ROUTE=yes)
      (ADDRESS=(PROTOCOL=tcp) (HOST=host1) (PORT=1630)) # 2
      (ADDRESS=(PROTOCOL=tcp) (HOST=host2) (PORT=1521)))
(ADDRESS_LIST=
      (SOURCE_ROUTE=yes)
      (ADDRESS=(PROTOCOL=tcp) (HOST=host3) (port=1630))
      (ADDRESS=(PROTOCOL=tcp) (HOST=host4) (port=1521)))
      (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com))) # 3
```

The previous syntax does the following:

1. The client selects an ADDRESS_LIST at random and fails over to the other address if the chosen ADDRESS_LIST fails. This is indicated if you set the LOAD_BALANCE and FAILOVER parameters to on.



- When an ADDRESS_LIST is chosen, the client first connects to Oracle Connection Manager using the Oracle Connection Manager protocol address that uses port 1630 as is indicated for the ADDRESS_LIST.
- 3. Oracle Connection Manager then connects to the database service using the protocol address that is indicated for the ADDRESS LIST.

6.6 Connect Descriptor Descriptions

Specify connect descriptors using the DESCRIPTION parameter. Identify multiple connect descriptors with the DESCRIPTION LIST parameter.

DESCRIPTION LIST

DESCRIPTION_LIST networking parameter of the tnsnames.ora file defines a list of connect descriptors for a particular net service name.

DESCRIPTION

DESCRIPTION networking parameter of the tnsnames.ora file specifies a container for a connect descriptor.

6.6.1 DESCRIPTION LIST

DESCRIPTION_LIST networking parameter of the tnsnames.ora file defines a list of connect descriptors for a particular net service name.

Purpose

To define a list of connect descriptors for a particular net service name.

Example 6-6 Example

```
net_service_name=
(DESCRIPTION_LIST=
  (DESCRIPTION=
    (ADDRESS=...)
    (CONNECT_DATA=(SERVICE_NAME=sales.example.com)))
  (DESCRIPTION=
```

6.6.2 DESCRIPTION

DESCRIPTION networking parameter of the tnsnames.ora file specifies a container for a connect descriptor.

Purpose

To specify a container for a connect descriptor.

Usage Notes

When using more than one DESCRIPTION parameter, put the parameters under the DESCRIPTION LIST parameter.

Example 6-7 Example

```
net_service_name=
(DESCRIPTION=
```



```
(ADDRESS=...)
(CONNECT DATA=(SERVICE NAME=sales.us.example.com)))
```

6.7 Protocol Addresses

The protocol address section of a tnsnames.ora file specifies listener protocol addresses.

If there is only one listener protocol address, then use the ADDRESS parameter. If there is more than one address, then use the ADDRESS LIST parameter.

ADDRESS

The ADDRESS networking parameter is in the tnsnames.ora file and it specifies the protocol address under the ADDRESS LIST or the DESCRIPTION parameter for one listener.

HTTPS_PROXY

Learn to use the tnsnames.ora parameter HTTPS_PROXY to specify HTTP proxy host names to tunnel Transport Layer Security (TLS) client connections.

HTTPS_PROXY_PORT

Learn how to use the tnsnames.ora parameter HTTPS_PROXY_PORT to specify forward HTTP proxy host ports for tunneling Transport Layer Security (TLS) client connections.

ADDRESS LIST

The ADDRESS LIST networking parameter specifies the number of protocol addresses.

6.7.1 ADDRESS

The ADDRESS networking parameter is in the tnsnames.ora file and it specifies the protocol address under the ADDRESS LIST or the DESCRIPTION parameter for one listener.

Purpose

To specify a single listener protocol address.

Usage Notes

Put this parameter under either the ADDRESS LIST parameter or the DESCRIPTION parameter.

Example

```
net_service_name=
(DESCRIPTION=
    (ADDRESS=(PROTOCOL=tcp) (HOST=sales-svr) (PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=sales.us.example.com))
```

6.7.2 HTTPS_PROXY

Learn to use the tnsnames.ora parameter HTTPS_PROXY to specify HTTP proxy host names to tunnel Transport Layer Security (TLS) client connections.

Purpose

To specify HTTP proxy host name for tunneling TLS client connections.



Usage Notes

The clients can tunnel secure connections over forward HTTP proxy using HTTP CONNECT method. This helps in accessing the public cloud database service as it eliminates the requirement to open an outbound port on a client side firewall. This parameter is applicable only to the connect descriptors where PROTOCOL=TCPS. This is similar to the web browser setting for intranet users who want to connect to internet hosts. Increase the forward web proxy read timeout for requests to a higher value depending on client queries. Otherwise, the forward web proxy closes the connection assuming that no requests are made from the client.

Successful connection depends on specific proxy configurations. The performance of data transfers depends on proxy capacity. Oracle recommends not to use this feature in production environments where performance is critical.

Configuring thisnames.ora for the HTTP proxy may not be enough depending your organization's network configuration and security policies. For example, some networks require a user name and password for the HTTP proxy.

Oracle Client versions earlier than 18c does not support connections through HTTP proxy.

Contact your network administrator to open outbound connections to hosts in the oraclecloud.com domain using the relevant port, without going through an HTTP proxy. For example, port 1522.

Default

None

Values

HTTP proxy host name that can make an outbound connection to the internet hosts.

Example

HTTPS PROXY=www-proxy.example.com

6.7.3 HTTPS_PROXY_PORT

Learn how to use the tnsnames.ora parameter HTTPS_PROXY_PORT to specify forward HTTP proxy host ports for tunneling Transport Layer Security (TLS) client connections.

Purpose

To specify forward HTTP proxy host port for tunneling TLS client connections.

Usage Notes

It forwards the HTTP proxy host port that receives HTTP CONNECT method. This parameter should be used along with HTTPS_PROXY_PORT. This value takes effect only when SQLNET.USE HTTPS PROXY=1 is set in sqlnet.ora.

Default

none

Values

port number



Example

HTTPS_PROXY_PORT=80

6.7.4 ADDRESS LIST

The ADDRESS LIST networking parameter specifies the number of protocol addresses.

Purpose

To define a list of protocol addresses.

Usage Notes

If there is only one listener protocol address, then ADDRESS LIST is not necessary.

Put this parameter either under the DESCRIPTION parameter or the DESCRIPTION_LIST parameter.

Example

```
net_service_name=
(DESCRIPTION=
  (ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=tcp) (HOST=sales1-svr) (PORT=1521))
      (ADDRESS=(PROTOCOL=tcp) (HOST=sales2-svr) (PORT=1521)))
  (CONNECT DATA=(SERVICE NAME=sales.us.example.com)))
```

6.8 Optional Parameters for Description

Use these parameters to configure connect descriptor containers under the DESCRIPTION parameter.

ENABLE

Use the ${\tt ENABLE}$ parameter to enable the keepalive feature on the supported TCP transports.

FAILOVER

Use the ${\tt FAILOVER}$ parameter to enable or disable connect-time failover for multiple protocol addresses.

LOAD BALANCE

Use the LOAD_BALANCE parameter to enable or disable client load balancing for multiple protocol addresses.

RECV BUF SIZE

Use the RECV BUF SIZE parameter to specify buffer space for session receive operations.

SDL

Use the SDU parameter to configure the session data unit (SDU) size.

SEND BUF SIZE

Use the ${\tt SEND}\ {\tt BUF}\ {\tt SIZE}$ parameter to specify buffer space for session send operations.

SOURCE ROUTE

Use the <code>SOURCE_ROUTE</code> parameter to enable routing through multiple protocol addresses.

TYPE OF SERVICE

Use the ${\tt TYPE_OF_SERVICE}$ parameter to specify the type of service to use for an Oracle Rdb database.

6.8.1 ENABLE

Use the ENABLE parameter to enable the keepalive feature on the supported TCP transports.

Purpose

To allow the caller to detect a terminated remote server; typically it takes 2 hours or more to notice.

Usage Notes

The keepalive feature on the supported TCP transports can be enabled for a net service client by putting (ENABLE=broken) under the DESCRIPTION parameter in the connect string. On the client side, the default for tcp_keepalive is off. Operating system TCP configurables, which vary by platform, define the actual keepalive timing details.

Values

broken

Example

```
net_service_name=
(DESCRIPTION=
  (ENABLE=broken)
  (ADDRESS=(PROTOCOL=tcp) (HOST=sales1-svr) (PORT=1521))
  (ADDRESS=(PROTOCOL=tcp) (HOST=sales2-svr) (PORT=1521)))
  (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com))
```

Although the preceding example has multiple addresses, the <code>ADDRESS_LIST</code> parameter was not used. This is because the <code>ADDRESS_LIST</code> parameter is not mandatory.

6.8.2 FAILOVER

Use the Failover parameter to enable or disable connect-time failover for multiple protocol addresses.

Purpose

To enable or disable connect-time failover for multiple protocol addresses.

Usage Notes

When you set the parameter to on, yes, or true, Oracle Net fails over at connect time to a different address if the first protocol address fails. When you set the parameter to off, no, or false, Oracle Net tries one protocol address.

Put this parameter under the <code>DESCRIPTION_LIST</code> parameter, the <code>DESCRIPTION</code> parameter, or the <code>ADDRESS LIST</code> parameter.



Do not set the GLOBAL_DBNAME parameter in the SID_LIST_listener_name section of the listener.ora. A statically configured global database name disables connect-time failover.

Default

on for the DESCRIPTION LIST, DESCRIPTION, and ADDRESS LIST parameters

Values

- yes on true
- no | off | false

Example

```
net_service_name=
(DESCRIPTION=
    (FAILOVER=on)
    (ADDRESS_LIST=
          (ADDRESS=(PROTOCOL=tcp) (HOST=sales1-svr) (PORT=1521))
          (ADDRESS=(PROTOCOL=tcp) (HOST=sales2-svr) (PORT=1521)))
(CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)))
```

6.8.3 LOAD BALANCE

Use the $LOAD_BALANCE$ parameter to enable or disable client load balancing for multiple protocol addresses.

Purpose

To enable or disable client load balancing for multiple protocol addresses.

Usage Notes

When you set the parameter to on, yes, or true, Oracle Net goes through the list of addresses in a random sequence, balancing the load on the various listener or Oracle Connection Manager protocol addresses. When you set the parameter to off, no, or false, Oracle Net tries the protocol addresses sequentially until one succeeds.

Put this parameter under the <code>DESCRIPTION_LIST</code> parameter, the <code>DESCRIPTION</code> parameter, or the <code>ADDRESS_LIST</code> parameter.

Default

on for DESCRIPTION LIST

Values

- yes | on | true
- no | off | false



Example

6.8.4 RECV BUF SIZE

Use the RECV BUF SIZE parameter to specify buffer space for session receive operations.

Purpose

To specify, in bytes, the buffer space for receive operations of sessions.

Usage Notes

This parameter is supported by the TCP/IP, TCP/IP with TLS, and SDP protocols.

Put this parameter under the DESCRIPTION parameter or at the end of the protocol address.

Setting this parameter in the connect descriptor for a client overrides the RECV_BUF_SIZE parameter at the client-side sqlnet.ora file.



Additional protocols might support this parameter on certain operating systems. Refer to the operating system-specific documentation for additional information about additional protocols.

Default

The default value for this parameter is specific to the operating system. The default for the Linux 2.6 operating system is 87380 bytes.

Example



```
(CONNECT_DATA=
  (SERVICE NAME=hr.us.example.com)))
```

Related Topics

Oracle Database Net Services Administrator's Guide

6.8.5 SDU

Use the SDU parameter to configure the session data unit (SDU) size.

Purpose

To instruct Oracle Net to optimize the transfer rate of data packets being sent across the network with a specified SDU size.

Usage Notes

Put this parameter under the DESCRIPTION parameter.

Setting this parameter in the connect descriptor for a client overrides the <code>DEFAULT_SDU_SIZE</code> parameter at client-side <code>sqlnet.ora</code> file.

Default

```
8192 bytes (8 KB)
```

Values

512 to 2097152 bytes

Example

```
net_service_name=
(DESCRIPTION=
  (SDU=8192)
  (ADDRESS_LIST=
        (ADDRESS=(PROTOCOL=tcp) (HOST=sales1-server) (PORT=1521))
        (ADDRESS=(PROTOCOL=tcp) (HOST=sales2-server) (PORT=1521)))
(CONNECT_DATA=
        (SERVICE NAME=sales.us.example.com))
```

Related Topics

Oracle Database Net Services Administrator's Guide

6.8.6 SEND_BUF_SIZE

Use the SEND BUF SIZE parameter to specify buffer space for session send operations.

Purpose

To specify, in bytes, the buffer space for send operations of sessions.

Usage Notes

This parameter is supported by the TCP/IP, TCP/IP with TLS, and SDP protocols.

Put this parameter under the DESCRIPTION parameter or at the end of the protocol address.

Setting this parameter in the connect descriptor for a client overrides the SEND_BUF_SIZE parameter at the client-side sqlnet.ora file.



Additional protocols might support this parameter on certain operating systems. Refer to the operating system-specific documentation for information about additional protocols.

Default

The default value for this parameter is operating system specific. The default for the Linux 2.6 operating system is 16 KB.

Example

```
net service name=
 (DESCRIPTION=
   (ADDRESS LIST=
     (ADDRESS=(PROTOCOL=tcp) (HOST=sales1-server) (PORT=1521)
        (SEND BUF SIZE=11784))
     (ADDRESS=(PROTOCOL=tcp) (HOST=sales2-server) (PORT=1521)
        (SEND BUF SIZE=11784)))
   (CONNECT DATA=
     (SERVICE NAME=sales.us.example.com)))
net service name=
 (DESCRIPTION=
   (SEND BUF SIZE=11784)
   (ADDRESS LIST=
     (ADDRESS=(PROTOCOL=tcp) (HOST=hr1-server) (PORT=1521)
     (ADDRESS=(PROTOCOL=tcp)(HOST=hr2-server)(PORT=1521)))
   (CONNECT DATA=
     (SERVICE NAME=hr.us.example.com)))
```

Related Topics

Oracle Database Net Services Administrator's Guide

6.8.7 SOURCE ROUTE

Use the SOURCE ROUTE parameter to enable routing through multiple protocol addresses.

Purpose

To enable routing through multiple protocol addresses.

Usage Notes

When you set this parameter to on or yes, Oracle Net uses each address in order until the destination is reached.

To use Oracle Connection Manager, an initial connection from the client to Oracle Connection Manager is required, and a second connection from Oracle Connection Manager to the listener is required.

Put this parameter under either the DESCRIPTION_LIST parameter, the DESCRIPTION parameter, or the ADDRESS LIST parameter.

Default

off

Values

- yes on
- no off

Example

```
net_service_name=
  (DESCRIPTION=
        (SOURCE_ROUTE=on)
        (ADDRESS=(PROTOCOL=tcp) (HOST=cman-pc) (PORT=1630))
        (ADDRESS=(PROTOCOL=tcp) (HOST=sales1-svr) (PORT=1521))
        (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com))
```

Related Topics

Oracle Database Net Services Administrator's Guide

6.8.8 TYPE_OF_SERVICE

Use the TYPE_OF_SERVICE parameter to specify the type of service to use for an Oracle Rdb database.

Purpose

To specify the type of service to use for an Oracle Rdb database.

Usage Notes

This parameter should only be used if the application supports both an Oracle Rdb and Oracle database service, and you want the application to load balance between the two.

Put this parameter under the DESCRIPTION parameter.

Example

Related Topics

Oracle Database Net Services Administrator's Guide

6.9 Connection Data Section

Learn how to configure network connections with protocol addresses.

A network object is identified by a protocol address. When a connection is made, the client and the receiver of the request (listener or Oracle Connection Manager) are configured with identical protocol addresses. The client uses this address to send the connection request to a particular network object location, and the recipient "listens" for requests on this address, and grants a connection based on its address information matching the client information.

- COLOCATION_TAG
- CONNECT DATA

Use the CONNECT DATA parameter to define the connection service.

- FAILOVER MODE
- GLOBAL NAME
- HS
- INSTANCE NAME
- KERBEROS5 PRINCIPAL

Use the KERBEROS5_PRINCIPAL parameter to set the Kerberos principal name associated with the Kerberos credentials cache (CC) file.

- RDB DATABASE
- SHARDING KEY

Use the SHARDING_KEY parameter to route the database connection request to an appropriate shard.

SUPER_SHARDING_KEY

Use the <code>SUPER_SHARDING_KEY</code> parameter in the case of composite sharding to route the database request to a collection of shards (shardspace).

- SERVER
- SERVICE NAME

6.9.1 COLOCATION TAG

Purpose

To direct the listener to route all connections with the same <code>colocation_tag</code> to the same database instance.

Usage Notes

Use this parameter with the CONNECT DATA parameter.

The parameter value must be an alphanumeric string.

Example

net_service_name=
 (DESCRIPTION=
 (ADDRESS_LIST=



```
(ADDRESS=...)
  (ADDRESS=...))
(CONNECT_DATA=
   (SERVICE_NAME=sales.us.example.com)
  (COLOCATION TAG=abc)))
```

Note:

Under certain conditions, such as, when maximum load of an instance is reached or when new instances are added or deleted for a service, the colocation of client connections that have the same <code>colocation_tag</code> to the same database instance may not be consistent.

6.9.2 CONNECT_DATA

Use the CONNECT DATA parameter to define the connection service.

Purpose

To define the service to which you want to connect, such as SERVICE NAME.

Usage Notes

Put this parameter under the DESCRIPTION parameter. CONNECT_DATA permits additional parameters as listed in Connection Data Section.

Example

6.9.3 FAILOVER_MODE

Purpose

To instruct Oracle Net to fail over to a different listener if the first listener fails during run time.

Usage Notes

Depending upon the configuration, the session or any SELECT statements which were in progress are automatically failed over.

This type of failover is called Transparent Application Failover (TAF) and should not be confused with the connect-time failover FAILOVER parameter.

Put this parameter under the CONNECT DATA parameter.

Additional Parameters

FAILOVER MODE supports the following parameters:

- BACKUP: Specifies the failover node by its net service name. A separate net service name must be created for the failover node.
- TYPE: Specifies the type of failover. Three types of Oracle Net failover functionality are available by default to Oracle Call Interface (OCI) applications:
 - SESSION: Fails over the session. For example, if a user's connection is lost, then a new session is automatically created for the user on the backup. This type of failover does not attempt to recover selects.
 - SELECT: Allows users with open cursors to continue fetching them after failure.
 However, this mode involves overhead on the client side in normal select operations.
 - NONE: This is the default, in which no failover functionality is used. This can also be explicitly specified to prevent failover from happening.
- METHOD: Specifies how fast failover is to occur from the primary node to the backup node:
 - BASIC: Establishes connections at failover time. This option requires almost no work on the backup database server until failover time.
 - PRECONNECT: Pre-establishes connections. This provides faster failover but requires that the backup instance be able to support all connections from every supported instance.
- TRANSACTION: Allows the database to complete the current database transaction following a recoverable error. This parameter is used with the COMMIT OUTCOME=TRUE parameter.
- RETRIES: Specifies the number of times to attempt to connect after a failover. If DELAY is specified, then RETRIES defaults to five retry attempts.
- DELAY: Specifies the amount of time in seconds to wait between connect attempts. If RETRIES is specified, then DELAY defaults to one second.



If a callback function is registered, then RETRIES and DELAY parameters are ignored.

See Also:

Oracle Database Net Services Administrator's Guide for additional configuration information

6.9.4 GLOBAL_NAME

Purpose

To identify the Oracle Rdb database.

Usage Notes

Put this parameter under the CONNECT DATA parameter.



Example

```
net_service_name=
(DESCRIPTION=
    (ADDRESS_LIST=
        (ADDRESS=...)
        (ADDRESS=...))
(CONNECT_DATA=
        (SERVICE_NAME=generic)
        (RDB_DATABASE=[.mf]mf_personal.rdb)
        (GLOBAL_NAME=alpha5)))
```

6.9.5 HS

Purpose

To direct Oracle Net to connect to a non-Oracle system through Heterogeneous Services.

Usage Notes

Put this parameter under the ${\tt CONNECT_DATA}$ parameter.

Default

None

Values

οk

Example

```
net_service_name=
(DESCRIPTION=
    (ADDRESS_LIST=
         (ADDRESS=...)
         (ADDRESS=...))
(CONNECT_DATA=
         (SID=sales6)
    )
(HS=ok))
```

See Also:

Oracle Database Net Services Administrator's Guide for complete configuration information

6.9.6 INSTANCE_NAME

Purpose

To identify the database instance to access.

Usage Notes

Set the value to the value specified by the INSTANCE_NAME parameter in the initialization parameter file.

Put this parameter under the CONNECT DATA parameter.

Example

```
net_service_name=
  (DESCRIPTION=
          (ADDRESS_LIST=
                (ADDRESS=...)
                 (ADDRESS=...))
          (CONNECT_DATA=
                 (SERVICE_NAME=sales.us.example.com)
                 (INSTANCE_NAME=sales1)))
```

See Also:

Oracle Database Net Services Administrator's Guide for additional information about the use of ${\tt INSTANCE_NAME}$

6.9.7 KERBEROS5_PRINCIPAL

Use the ${\tt KERBEROS5_PRINCIPAL}$ parameter to set the Kerberos principal name associated with the Kerberos credentials cache (CC) file.

Purpose

When you configure Kerberos authentication for an Oracle Database client, you can specify multiple Kerberos principals with a single Oracle Database client.

This is an optional parameter. When specified, it is used to verify if the principal name in the credential cache (specified using SQLNET.KERBEROS5 CC NAME) matches the parameter value.

Usage Notes

Use this parameter with the CONNECT_DATA parameter in the tnsnames.ora file. Alternatively, you can specify KERBEROS5_CC_NAME in the connect string along with the optional KERBEROS5_PRINCIPAL parameter to connect as a different Kerberos principal.

Each Kerberos principal must have a valid credential cache. Oracle Database checks KERBEROS5_PRINCIPAL against the value that is retrieved from the credential cache. If the two values do not match, then the user is not authenticated.

Example

For a user krbuser1, who is externally authenticated using the Kerberos principal krbprinc1.example.com and the credential cache for this principal is located at /tmp/krbuser1/krb.cc, the connect string is:

```
net service_name=
```



Note:

The connection fails if the principal in the /tmp/krbuser1/krb.cc file does not contain the krbprincl@example.com value.

Similarly, for a user krbuser2, who is externally authenticated using the Kerberos principal krbprinc2.example.com and the credential cache for this principal is located at /tmp/krbuser2/krb.cc, the connect string is:

Related Topics

- SQLNET.KERBEROS5_CC_NAME
 Use the sqlnet.ora parameter SQLNET.KERBEROS5_CC_NAME to specify the complete path
 name to the Kerberos credentials cache file.
- Oracle Database Security Guide

6.9.8 RDB DATABASE

Purpose

To specify the file name of an Oracle Rdb database.

Usage Notes

Put this parameter under the CONNECT DATA parameter.

Example

```
net_service_name=
(DESCRIPTION=
    (ADDRESS_LIST=
         (ADDRESS=...)
         (ADDRESS=...))
(CONNECT_DATA=
         (SERVICE_NAME=sales.us.example.com)
         (RDB_DATABASE= [.mf]mf_personal.rdb)))
```



6.9.9 SHARDING_KEY

Use the SHARDING_KEY parameter to route the database connection request to an appropriate shard.

Purpose

To specify the value of a sharding key. Based on the value specified during a database connection request, the request is directly routed to the appropriate shard.

Usage Notes

You specify this parameter under the CONNECT_DATA section of a connect string or tnsnames.ora file.

Use the SHARDING_KEY parameter to specify a sharding key in simplified text format. This parameter supports only ASCII character set and not special characters. The following data types are supported for a sharding key:

- NUMBER
- INTEGER
- SMALLINT
- RAW
- NVARCHAR
- NVARCHAR2
- NCHAR
- DATE
- TIMESTAMP

Use the <code>SHARDING_KEY_B64</code> parameter to specify the base64-encoded binary representation of a sharding key. This parameter supports these special characters: " quotation mark , comma () close parenthesis + plus sign)

Values

The fields for base64-encoded values ($*_B64$) start with a header, which is a sequence of space-separated integer values:

```
(CONNECT_DATA=
  (SHARDING_KEY_B64=
     [version][type][key column 1 type identifier][key column 2 type
identifier] ..., [base64 string], [base64 string], [base64 string], ...))...
```

In the above syntax:

- Parts of the compound key are separated with a comma.
- *version* specifies the version number of base64 representation. Currently, only version 1 is supported, and thus the supported *version* value is 1.
- type specifies the character set string and its encoding information. The supported type values are:

Value	Character Set String	Encoding Scheme
0	String contains hash value.	Character values are encoded in - AL32UTF8 (for VARCHAR) and AL16UTF16 (for NVARCHAR).
1	String does not contain hash value.	
2	String does not contain hash value.	Character values are encoded in database encoding, which may be specific for each column.
3	String contains hash value.	
4	String contains only hash value.	

 key column type identifier specifies the data types. The supported key column type identifier values are:

Value	Data Type
1	VARCHAR, NVARCHAR, CHAR, NCHAR
2	NUMBER
6	NUMBER with length in first byte
12	DATE
23	RAW
180	TIMESTAMP

 The header is terminated by a comma and is followed by base64 string. base64 string is a comma-separated list of the base64-encoded value string. The hash value, if available, is the last value in the list.

Example 6-8

In the following sample connect string, the ${\tt SHARDING_KEY}$ parameter value is specified in simplified text format:

Example 6-9

In the following sample connect string, the <code>SHARDING_KEY_B64</code> parameter value is encoded to base64 binary representation:

```
net_service_name=
  (DESCRIPTION=
    (ADDRESS_LIST=
        (ADDRESS=(host=sales-east1) (port=1522))
        (ADDRESS=(host=sales-east2) (port=1522))
)
```

```
(CONNECT_DATA=
    (SERVICE_NAME=sales.us.example.com)
    (SHARDING_KEY_B64=1 1 2, VVM=, OTQwMDI=)
)
```

Related Topics

SUPER SHARDING KEY

Use the SUPER_SHARDING_KEY parameter in the case of composite sharding to route the database request to a collection of shards (shardspace).

Oracle Database Net Services Administrator's Guide

6.9.10 SUPER_SHARDING_KEY

Use the SUPER_SHARDING_KEY parameter in the case of composite sharding to route the database request to a collection of shards (shardspace).

Purpose

To specify a shardspace key for a collection of shards. A shardspace is set of shards that store data that corresponds to a range or list of key values. Based on the value specified during a database connection request, the request is directly routed to an appropriate shardspace.

Usage Notes

You specify this parameter under the CONNECT_DATA section of a connect string or tnsnames.ora file.

Use the <code>SUPER_SHARDING_KEY</code> parameter to specify a shardspace key for a collection of shards in simplified text format. This parameter supports only ASCII character set and not special characters. The supported data types for a super sharding key are the same as those for a sharding key.

Use the SUPER_SHARDING_KEY_B64 parameter to specify the base64-encoded binary representation of a shardspace key. This parameter supports special characters (such as "quotation mark, comma () close parenthesis + plus sign).

Values

The fields for base64-encoded values (*_B64) start with a header, which is a sequence of space-separated integer values:

```
(CONNECT_DATA=(SUPER_SHARDING_KEY_B64=[version] [type] [integer literal] [integer literal] ..., [base64 binary], [base64 binary], [base64 binary], ...))...
```

For details on each of the base64-encoded header fields, see SHARDING_KEY.

Example 6-10

In the following sample connect string, the <code>SHARDING_KEY</code> and <code>SUPER_SHARDING_KEY</code> parameter values are specified in simplified text format:

```
net_service_name=
  (DESCRIPTION=
```



```
(ADDRESS_LIST=
  (ADDRESS=(host=sales-east1) (port=1522))
  (ADDRESS=(host=sales-east2) (port=1522))
)
(CONNECT_DATA=
  (SERVICE_NAME=sales.us.example.com)
  ((SHARDING_KEY=40598230) (SUPER_SHARDING_KEY=gold))
)
```

Example 6-11

In the following sample connect string, the <code>SHARDING_KEY_B64</code> and <code>SUPER_SHARDING_KEY_B64</code> parameter values are encoded to base64 binary representation:

Related Topics

SHARDING KEY

Use the ${\tt SHARDING_KEY}$ parameter to route the database connection request to an appropriate shard.

Oracle Database Net Services Administrator's Guide

6.9.11 SERVER

Purpose

To direct the listener to connect the client to a specific type of service handler.

Usage Notes

Put this parameter under the CONNECT DATA parameter.

Values

- dedicated to specify whether client requests be served by dedicated server.
- shared to specify whether client requests be served by a dispatcher or shared server.
- pooled to get a connection from the connection pool if database resident connection pooling is enabled on the server.

Note:

- Shared server must be configured in the database initialization file in order for the client to connect to the database with a shared server process.
- The USE_DEDICATED_SERVER parameter in the sqlnet.ora file overrides this parameter.

Example

```
net_service_name=
(DESCRIPTION=
    (ADDRESS_LIST=
         (ADDRESS=...)
         (ADDRESS=...))
(CONNECT_DATA=
    (SERVICE_NAME=sales.us.example.com)
    (SERVER=dedicated)))
```

6.9.12 SERVICE_NAME

Purpose

To identify the Oracle Database database service to access.

Usage Notes

Set the value to a value specified by the SERVICE_NAMES parameter in the initialization parameter file.

Put this parameter under the CONNECT DATA parameter.

Example

```
net_service_name=
(DESCRIPTION=
    (ADDRESS_LIST=
        (ADDRESS=...)
        (ADDRESS=...))
(CONNECT_DATA=
        (SERVICE_NAME=sales.us.example.com)))
```

Related Topics

Oracle Database Net Services Administrator's Guide

6.10 Security Section

The security section of the tnsnames.ora file specifies these security-related parameters for use with Oracle security features.

AUTHENTICATION_SERVICE

Use the tnsnames.ora parameter AUTHENTICATION_SERVICE to enable an authentication service.

IGNORE ANO ENCRYPTION FOR TCPS

The IGNORE_ANO_ENCRYPTION_FOR_TCPS parameter specifies if the SQLNET.ENCRYPTION CLIENT parameter should be ignored for this specific TNS alias.

KERBEROS5 CC NAME

Use the tnsnames.ora parameter KERBEROS5_CC_NAME to specify the complete path name to the Kerberos credentials cache (CC) file.

OCI COMPARTMENT

Use the <code>OCI_COMPARTMENT</code> parameter to specify Oracle Cloud Identifier (OCID) of the compartment that holds database instances for client connections.

OCI DATABASE

Use the <code>OCI_DATABASE</code> parameter to specify Oracle Cloud Identifier (OCID) of the database that you want to access for the client connection.

OCI IAM URL

Use the OCI_IAM_URL parameter to specify an endpoint URL that the database client must connect with to get the database token for authenticating Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM) users on OCI Database as a Service (DBaaS).

OCI TENANCY

Use the OCI_TENANCY parameter to specify Oracle Cloud Identifier (OCID) of the user's tenancy.

PASSWORD AUTH

SECURITY

Use the SECURITY parameter to change the security properties of a connection.

SSL DISABLE WEAK EC CURVES

Use the SSL_DISABLE_WEAK_EC_CURVES parameter to disable the use of weak Elliptic Curve Cryptography (ECC) curves.

SSL_SERVER_CERT_DN

Use the ${\tt SSL_SERVER_CERT_DN}$ parameter to specify the distinguished name (DN) of the database server.

SSL SERVER DN MATCH

Use the $SSL_SERVER_DN_MATCH$ parameter to enforce server-side certificate validation through distinguished name (DN) matching.

SSL VERSION

Use the SSL_VERSION parameter to define valid Transport Layer Security (TLS) versions to be used for connections.

TOKEN AUTH

Use the TOKEN_AUTH parameter to configure token-based authentication for Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM) or Microsoft Azure users of Microsoft Entra ID (previously called Microsoft Azure Active Directory).

TOKEN LOCATION

Use the TOKEN_LOCATION parameter to specify the directory location where token file is stored for token-based authentication.

WALLET_LOCATION

Use the ${\tt WALLET_LOCATION}$ parameter in the ${\tt tnsnames.ora}$ file to specify different locations where Oracle wallets are stored.



6.10.1 AUTHENTICATION_SERVICE

Use the thsnames.ora parameter AUTHENTICATION_SERVICE to enable an authentication service.

Purpose

To enable an authentication service. If you have installed authentication, then Oracle recommends that you set AUTHENTICATION_SERVICE to either NONE or to one of the listed authentication methods.

Usage Notes

• Use this parameter in the SECURITY section of the this names or a file or directly as part of the connect string.

You can also set this value in the sqlnet.ora file. The AUTHENTICATION_SERVICE parameter is equivalent to the sqlnet.ora parameter SQLNET.AUTHENTICATION_SERVICES. The parameter value specified in the connect string takes precedence.

Note that the AUTHENTICATION_SERVICE parameter can take only a single authentication service, unlike the SQLNET.AUTHENTICATION_SERVICES parameter that can take multiple values.

- When using the AUTHENTICATION_SERVICE=ALL, the server attempts to authenticate using each of the following methods:
 - 1. Authentication based on a service external to the database, such as a service on the network layer, Kerberos, or RADIUS.
 - 2. Authentication based on the operating system user's membership in an administrative operating system group. Group names are platform-specific. This authentication applies to administrative connections only.
 - 3. Authentication performed by the database.
 - 4. Authentication based on credentials stored in a directory server.

The server falls back to the authentication methods that appear further down on the list if attempts to use the authentication methods appearing higher on the list were unsuccessful.

- When using local database password authentication (no external authentication), set AUTHENTICATION_SERVICE=NONE for better client performance.
- Operating system authentication enables access to the database using any user name and any password when an administrative connection is attempted to the CDB root, such as using the AS SYSDBA clause when connecting using SQL*Plus.

An example of a connection to the CDB root is as follows.

```
sqlplus ignored username/ignored password AS SYSDBA
```

When the operating-system user who issued the preceding command is already a member of the appropriate administrative operating system group, then the connection is successful. This is because Oracle checks the group membership first, and thus the user name and password are ignored by the server.

Default

ALL





When installing Oracle Database with Database Configuration Assistant (DBCA), you can set this parameter to \mathtt{NTS} in the $\mathtt{sqlnet.ora}$ file.

Values

Authentication methods that are available with Oracle Net Services:

- NONE for no authentication methods, including Microsoft Windows native operating system authentication. When you set AUTHENTICATION_SERVICE to NONE, then the user can use a valid user name and password to access the database.
- ALL for all authentication methods.
- BEQ for native operating system authentication for operating systems other than Microsoft Windows.
- KERBEROS5 for Kerberos authentication.
- RADIUS for Remote Authentication Dial-In User Service (RADIUS) authentication.
- TCPS for TLS authentication.
- NTS for Microsoft Windows native operating system authentication. In this case, the user
 must authenticate to the database (CDB root) with OS credentials using Windows native
 authentication. No external password is needed. NTS checks the group membership for an
 OS user. For example, if an OS user is a member of the ORA_DBA group, then the user can
 log in to the database as SYSDBA.

Note:

With the AUTHENTICATION_SERVICE=NTS setting, if you try to connect through SQL*Plus using NTS authentication and specify an external password (for example, SQL*Plus SYSTEM/password), then the connection fails with an ORA-12638: credential retrieval failed error. For regular user name and password based authentication, set the value to NONE.

Example

Related Topics

- Oracle Database Security Guide
- SQLNET.AUTHENTICATION_SERVICES

 Use the sqlnet.ora parameter SQLNET.AUTHENTICATION_SERVICES to enable one or more authentication services.



6.10.2 IGNORE ANO ENCRYPTION FOR TCPS

The IGNORE_ANO_ENCRYPTION_FOR_TCPS parameter specifies if the SQLNET.ENCRYPTION_CLIENT parameter should be ignored for this specific TNS alias.

Purpose

To specify if the <code>SQLNET.ENCRYPTION_CLIENT</code> parameter should be ignored for this specific TNS alias.

Usage Notes

If your requirements are that <code>SQLNET.ENCRYPTION_SERVER</code> be set to <code>required</code>, then you can set the <code>IGNORE_ANO_ENCRYPTION_FOR_TCPS</code> parameter in both <code>SQLNET.ENCRYPTION_CLIENT</code> and <code>SQLNET.ENCRYPTION_SERVER</code> to <code>TRUE</code>. This forces the client to ignore the value that is set for the <code>SQLNET.ENCRYPTION</code> <code>CLIENT</code> parameter for all outgoing TCPS connections.

Default

FALSE

Example 6-12 Example

6.10.3 KERBEROS5_CC_NAME

Use the tnsnames.ora parameter KERBEROS5_CC_NAME to specify the complete path name to the Kerberos credentials cache (CC) file.

Purpose

To specify the complete path name to the Kerberos CC file.

Usage Notes

Use this parameter in the SECURITY section of the tnsnames.ora file or directly as part of the connect string.

You can also set this value in the sqlnet.ora file. The KERBEROS5_CC_NAME parameter is equivalent to the sqlnet.ora parameter SQLNET.KERBEROS5_CC_NAME. The parameter value specified in the connect string takes precedence.

Values and Examples

You can use the following formats to specify a value for KERBEROS5 CC NAME:

- If the Oracle database is using a directory cache:
 - KERBEROS5_CC_NAME=complete_path_to_cc_file

For example:

```
KERBEROS5_CC_NAME=/tmp/kcache
KERBEROS5_CC_NAME=D:\tmp\kcache
```

KERBEROS5_CC_NAME=FILE: complete_path_to_cc_ file

For example:

```
KERBEROS5 CC NAME=FILE:/tmp/kcache
```

- If the Oracle database is using the native Windows cache:
 - KERBEROS5 CC NAME=OSMSFT://
 - KERBEROS5 CC NAME=MSLSA:

The OSMSFT and MSLSA options specify that the file is on Microsoft Windows and is running Microsoft Kerberos Key Distribution Center (KDC).

Note:

If you want to authenticate multiple Kerberos principals, then you can specify additional Kerberos principals either through the connect string directly or in the tnsnames.ora file.

Default

The default value is operating system-dependent, as follows:

- On Linux and UNIX operating systems: /tmp/krb5cc userid
- On Microsoft Windows operating systems: c:\tmp\krbcache

Related Topics

KERBEROS5 PRINCIPAL

Use the KERBEROS5_PRINCIPAL parameter to set the Kerberos principal name associated with the Kerberos credentials cache (CC) file.

SQLNET.KERBEROS5 CC NAME

Use the sqlnet.ora parameter SQLNET.KERBEROS5_CC_NAME to specify the complete path name to the Kerberos credentials cache file.

Oracle Database Security Guide

6.10.4 OCI COMPARTMENT

Use the OCI_COMPARTMENT parameter to specify Oracle Cloud Identifier (OCID) of the compartment that holds database instances for client connections.

Purpose

To define the scope of your database token request. This value instructs the database client to initiate a token request to databases within the specified compartment only.

Usage Notes

You can use this parameter along with the PASSWORD_AUTH, OCI_IAM_URL, and OCI_TENANCY parameters while configuring token-based authentication for Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM) users on OCI Database as a Service (DBaaS). You



can also use the optional <code>OCI_DATABASE</code> parameter to specify a database instance within the compartment for your connection.

With this configuration, the database client can only request an IAM database token using the IAM user name and IAM database password. The client cannot request an IAM database token for an API-key, delegation token, security token, resource principal, service principal, or instance principal.

The OCI_COMPARTMENT parameter is optional if OCI_DATABASE is not set. If you choose to set OCI_DATABASE, then you must also set OCI_COMPARTMENT so that your token request is for the specified database in that compartment.

If you do not set both <code>OCI_COMPARTMENT</code> and <code>OCI_DATABASE</code>, then the entire tenancy is the scope of your token request.

Use this parameter under the SECURITY section of the tnsnames.ora file, sqlnet.ora file, or directly as part of the command-line connect string. The parameter value specified in the connect string takes precedence over the other specified values.

Default

None

Value

OCID for the IAM compartment to allow access for the database token. You can get the OCID value for your compartment from the Compartments information page in the OCI console.

The compartment OCID uses this syntax:

```
OCI COMPARTMENT=compartment OCID
```

For details on the syntax options, see Oracle Cloud IDs (OCIDs).

Examples

```
In the tnsnames.ora file:
```

```
net service name=
  (DESCRIPTION=
     (ADDRESS=(PROTOCOL=tcps)(HOST=salesserver1)(PORT=1522))
     (SECURITY=
        (SSL SERVER DN MATCH=TRUE)
        (SSL SERVER CERT DN="C=US, O=example, CN=OracleContext")
        (PASSWORD AUTH=OCI TOKEN)
        (OCI IAM URL=https://auth.us-region-1.example.com/v1/actions/
generateScopedAccessBearerToken)
        (OCI TENANCY=ocid1.tenancy..12345)
        (OCI COMPARTMENT=ocid1.compartment..12345)
        (OCI DATABASE=ocid1.autonomousdatabase.oc1.12345))
     (CONNECT DATA=(SERVICE NAME=sales.us.example.com))
In the sqlnet.ora file:
SSL SERVER DN MATCH=TRUE
PASSWORD AUTH=OCI TOKEN
OCI IAM URL=https://auth.us-region-1.example.com/v1/actions/
```



```
generateScopedAccessBearerToken
OCI_TENANCY=ocid1.tenancy..12345
OCI_COMPARTMENT=ocid1.compartment..12345
OCI_DATABASE=ocid1.autonomousdatabase.oc1.12345
```

Related Topics

- Oracle Database Security Guide
- PASSWORD_AUTH

6.10.5 OCI DATABASE

Use the OCI_DATABASE parameter to specify Oracle Cloud Identifier (OCID) of the database that you want to access for the client connection.

Purpose

To define the scope of your database token request. The database OCID value instructs the database client to initiate a token request to the specified database within your compartment.

Usage Notes

This parameter is optional. You can use this parameter along with the PASSWORD_AUTH, OCI_IAM_URL, OCI_TENANCY, and OCI_COMPARTMENT parameters while configuring token-based authentication for Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM) users on OCI Database as a Service (DBaaS).

With this configuration, the database client can only request an IAM database token using the IAM user name and IAM database password. The client cannot request an IAM database token for an API-key, delegation token, security token, resource principal, service principal, or instance principal.

The <code>OCI_DATABASE</code> value limits your token request to the specified database only. If you set <code>OCI_DATABASE</code>, then you must also set <code>OCI_COMPARTMENT</code> so that your token request is for the specified database in that compartment.

Use this parameter under the SECURITY section of the tnsnames.ora file, sqlnet.ora file, or directly as part of the command-line connect string. The parameter value specified in the connect string takes precedence over the other specified values.

Default

None

Value

OCID of the database that you want to access for the client connection. You can get the OCID value for your database from the Database details page in the OCI console.

The database OCID uses this syntax:

OCI DATABASE=database OCID

For details on the syntax options, see Oracle Cloud IDs (OCIDs).



Examples

```
In the tnsnames.ora file:
net service name=
  (DESCRIPTION=
     (ADDRESS=(PROTOCOL=tcps)(HOST=salesserver1)(PORT=1522))
     (SECURITY=
        (SSL SERVER DN MATCH=TRUE)
        (SSL SERVER CERT DN="C=US,O=example,CN=OracleContext")
        (PASSWORD AUTH=OCI TOKEN)
        (OCI IAM URL=https://auth.us-region-1.example.com/v1/actions/
generateScopedAccessBearerToken)
        (OCI TENANCY=ocid1.tenancy..12345)
        (OCI COMPARTMENT=ocid1.compartment..12345)
        (OCI DATABASE=ocid1.autonomousdatabase.oc1.12345))
     (CONNECT DATA=(SERVICE NAME=sales.us.example.com))
  )
In the sqlnet.ora file:
SSL SERVER DN MATCH=TRUE
PASSWORD AUTH=OCI TOKEN
OCI IAM URL=https://auth.us-region-1.example.com/v1/actions/
generateScopedAccessBearerToken
OCI TENANCY=ocid1.tenancy..12345
OCI COMPARTMENT=ocid1.compartment..12345
OCI DATABASE=ocid1.autonomousdatabase.oc1.12345
```

Related Topics

- Oracle Database Security Guide
- PASSWORD_AUTH

6.10.6 OCI_IAM_URL

Use the <code>OCI_IAM_URL</code> parameter to specify an endpoint URL that the database client must connect with to get the database token for authenticating Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM) users on OCI Database as a Service (DBaaS).

Purpose

To specify the IAM URL for your REST API requests. The database client connects to this URL to retrieve the database token from IAM.

Usage Notes

You set the OCI_IAM_URL parameter along with the PASSWORD_AUTH and OCI_TENANCY parameters while configuring IAM token-based authentication (using the IAM user name and IAM database password to retrieve the database token). These parameters are mandatory.

With this configuration, the database client can only request an IAM database token using the IAM user name and IAM database password. The client cannot request an IAM database

token for an API-key, delegation token, security token, resource principal, service principal, or instance principal.

You can also set the optional OCI_COMPARTMENT and OCI_DATABASE parameters to specify the scope of your token request.

Use this parameter under the SECURITY section of the tnsnames.ora file, sqlnet.ora file, or directly as part of the command-line connect string. The parameter value specified in the connect string takes precedence over the other specified values.

Default

None

Value

OCI IAM endpoint URL that the database client must connect with to get the database token. This URL is specific to your region and uses this syntax:

```
<authentication regional endpoint>/v1/actions/generateScopedAccessBearerToken
```

You can derive this value by replacing <authentication_regional_endpoint> with the API endpoint URL for your region. To obtain the appropriate API endpoint URL, see Identity and Access Management Data Plane API.

For example, if you want to use the URL as https://auth.us-region-1.example.com, then your OCI_IAM_URL value is:

```
https://auth.us-region-1.example.com/v1/actions/generateScopedAccessBearerToken
```

Examples

```
In the tnsnames.ora file:
```

```
net service name=
  (DESCRIPTION=
     (ADDRESS=(PROTOCOL=tcps) (HOST=salesserver1) (PORT=1522))
     (SECURITY=
        (SSL SERVER DN MATCH=TRUE)
        (SSL SERVER CERT DN="C=US, O=example, CN=OracleContext")
        (PASSWORD AUTH=OCI TOKEN)
        (OCI IAM URL=https://auth.us-region-1.example.com/v1/actions/
generateScopedAccessBearerToken)
        (OCI TENANCY=ocid1.tenancy..12345))
     (CONNECT DATA=(SERVICE NAME=sales.us.example.com))
In the sqlnet.ora file:
SSL SERVER DN MATCH=TRUE
PASSWORD AUTH=OCI TOKEN
OCI IAM URL=https://auth.us-region-1.example.com/v1/actions/
generateScopedAccessBearerToken
OCI TENANCY=ocid1.tenancy..12345
```



In these examples, the optional <code>OCI_COMPARTMENT</code> and <code>OCI_DATABASE</code> parameters are not specified and thus the entire tenancy is set as the scope of the token request.

Related Topics

- Oracle Database Security Guide
- PASSWORD_AUTH

6.10.7 OCI_TENANCY

Use the OCI_TENANCY parameter to specify Oracle Cloud Identifier (OCID) of the user's tenancy.

Purpose

To specify OCID of the user's tenancy (root compartment).

Usage Notes

You set this parameter along with the mandatory PASSWORD_AUTH and OCI_IAM_URL parameters while configuring token-based authentication for Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM) users on OCI Database as a Service (DBaaS).

With this configuration, the database client can only request an IAM database token using the IAM user name and IAM database password. The client cannot request an IAM database token for an API-key, delegation token, security token, resource principal, service principal, or instance principal.

You can also set the optional OCI_COMPARTMENT and OCI_DATABASE parameters to specify the scope of your token request. If you do not set the OCI_COMPARTMENT and OCI_DATABASE parameter values, then the entire tenancy is the scope of your token request.

Use this parameter under the SECURITY section of the tnsnames.ora file, sqlnet.ora file, or directly as part of the command-line connect string. The parameter value specified in the connect string takes precedence over the other specified values.

Default

None

Value

OCID of the user's tenancy. You can get the OCID value for your tenancy from the Tenancy information page in the OCI console.

The tenancy OCID uses this syntax:

```
OCI TENANCY=tenancy OCID
```

For details on the syntax options, see Oracle Cloud IDs (OCIDs).

Examples

In the tnsnames.ora file:

```
net_service_name=
  (DESCRIPTION=
          (ADDRESS=(PROTOCOL=tcps) (HOST=salesserver1) (PORT=1522))
          (SECURITY=
```



```
(SSL_SERVER_DN_MATCH=TRUE)
    (SSL_SERVER_CERT_DN="C=US,O=example,CN=OracleContext")
    (PASSWORD_AUTH=OCI_TOKEN)
    (OCI_IAM_URL=https://auth.us-region-1.example.com/v1/actions/
generateScopedAccessBearerToken)
    (OCI_TENANCY=ocid1.tenancy..12345))
    (CONNECT_DATA=(SERVICE_NAME=sales.us.example.com))
)
In the sqlnet.ora file:

SSL_SERVER_DN_MATCH=TRUE
PASSWORD_AUTH=OCI_TOKEN
OCI_IAM_URL=https://auth.us-region-1.example.com/v1/actions/
generateScopedAccessBearerToken
OCI_TENANCY=ocid1.tenancy..12345
```

In these examples, the optional <code>OCI_COMPARTMENT</code> and <code>OCI_DATABASE</code> parameters are not specified and thus the entire tenancy is set as the scope of the token request.

Related Topics

- Oracle Database Security Guide
- PASSWORD AUTH

6.10.8 PASSWORD_AUTH

Use the PASSWORD_AUTH parameter to configure an authentication method for Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM) users on OCI Database as a Service (DBaaS). With this setting, client connections use the IAM user name and IAM database password for logging in users to the database.

Purpose

To configure either IAM database password verifier authentication or IAM token-based authentication, using the IAM user name and IAM database password for the access.

For password verifier authentication, the database server retrieves an IAM database password verifier from IAM. For token-based authentication, the database client requests a database token (db-token) from IAM.

Usage Notes

- Use this parameter under the SECURITY section of the tnsnames.ora file, sqlnet.ora file, or directly as part of the command-line connect string. The parameter value specified in the connect string takes precedence over the other specified values.
- This setting instructs the database client to either use the existing password login process
 with the database server (password verifier authentication) or to get a token with the IAM
 user name and IAM database password (token-based authentication). This IAM database
 password is different from the OCI console password. An IAM user can set this password
 from the OCI console.

See Create an OCI IAM password to use for Autonomous Databases User Authentication and Authorization.

By default, this parameter is set to PASSWORD_VERIFIER. The
 PASSWORD_AUTH=PASSWORD_VERIFIER setting configures IAM database password verifier
 authentication. The database server retrieves an IAM database password verifier (an
 encrypted hash of password) from IAM to authenticate users.

When an IAM user logs in with the IAM user name and IAM database password using <code>@connect_identifier</code>, the <code>PASSWORD_AUTH=PASSWORD_VERIFIER</code> setting along with <code>@connect_identifier</code> instructs the database client to follow the existing user name and password login process with the database server.

You can use the PASSWORD_AUTH parameter to override the tnsnames.ora or sqlnet.ora setting by specifying a different value in the connect string.

• To configure IAM token-based authentication with the IAM user name and IAM database password, set PASSWORD_AUTH=OCI_TOKEN. The database client requests a database token (db-token) from IAM for the user to access the database.

This db-token obtained by the client is a bearer token with an expiration time and scope, and does not come with a private key. These tokens are transmitted over secure channels. You must use only the TCP/IP with Transport Layer Security (TLS) protocol, otherwise an error message appears indicating that non-TLS connections are disallowed.

When an IAM user logs in with the IAM user name and IAM database password using / @connect_identifier, the PASSWORD_AUTH=OCI_TOKEN setting along with / @connect_identifier instructs the database client to get the token directly from an OCI IAM endpoint using a REST API request. If the IAM user is mapped to a database schema (exclusively or shared), then the login is completed.

For the database client to retrieve the token from IAM, you must set additional parameters so that the database client can find the IAM endpoint along with additional meta-data. The additional parameters are OCI_IAM_URL and OCI_TENANCY along with the optional OCI_COMPARTMENT and OCI_DATABASE. These values enable the database client to make appropriate calls to the specified endpoint.

The <code>OCI_IAM_URL</code> parameter specifies the API endpoint URL that the database client must connect with. The <code>OCI_TENANCY</code> parameter specifies the OCID (Oracle Cloud Identifier) of the user's tenancy. The optional <code>OCI_COMPARTMENT</code> and <code>OCI_DATABASE</code> parameters limit the scope of your request.

This authentication method is more secure than using a password verifier because a password verifier is considered sensitive. Also, only the database client can retrieve the database token. Applications or tools cannot pass these types of tokens through the database client API.

Note:

You can also use other IAM user credentials (such as API-key, security token, resource principal, service principal, instance principal, or delegation token) to get the db-token. This db-token is a proof-of-possession (PoP) token. In this case, you use a different parameter setting (TOKEN_AUTH=OCI_TOKEN).

Unlike the IAM database password that can only be used by the database client to retrieve the token, these credentials require an application or tool to retrieve the token. See TOKEN AUTH.



Default

PASSWORD VERIFIER

Values and Examples

Value

For IAM database password verifier authentication: In the tnsnames.ora file:

PASSWORD AUTH=PASSWORD_VERIFIER

Note: Use of IAM user name and IAM database password with the IAM database password verifier is the default configuration, and you do not need to set any additional parameters for the client.

However, if PASSWORD AUTH is set to OCI TOKEN in the client-side sqlnet.ora file, then the client tries to connect with OCI IAM to retrieve a database token using the IAM user name and IAM database password. In this case, you can override this setting for a particular connection using PASSWORD_AUTH=PASSWORD_VERIFIER.

Example

```
net service name=
  (DESCRIPTION =
     (ADDRESS=(PROTOCOL=tcps)
(HOST=sales-svr) (PORT=1521))
     (SECURITY=
        (SSL SERVER DN MATCH=TRUE)
(SSL SERVER CERT DN="C=US,O=example,C
N=OracleContext")
(PASSWORD AUTH=PASSWORD VERIFIER))
(CONNECT DATA=(SERVICE NAME=sales.us.
example.com))
In the sqlnet.ora file:
PASSWORD_AUTH=PASSWORD_VERIFIER
```



Value

For IAM token-based authentication with the IAM user name and IAM database password:

PASSWORD AUTH=OCI TOKEN

Note: You must configure the TCPS protocol (PROTOCOL=tcps) and set the SSL_SERVER_DN_MATCH parameter to TRUE for token-based authentication.

Example

In the tnsnames.ora file:

net service name=

```
(DESCRIPTION=
     (ADDRESS=(PROTOCOL=tcps)
(HOST=salesserver1) (PORT=1522))
     (SECURITY=
        (SSL_SERVER_DN_MATCH=TRUE)
(SSL SERVER CERT DN="C=US,O=example,C
N=OracleContext")
        (PASSWORD AUTH=OCI_TOKEN)
        (OCI IAM URL=https://auth.us-
region-1.example.com/v1/actions/
generateScopedAccessBearerToken)
(OCI TENANCY=ocid1.tenancy..12345))
(CONNECT DATA=(SERVICE NAME=sales.us.
example.com))
 )
In the sqlnet.ora file:
```

SSL_SERVER_DN_MATCH=TRUE
PASSWORD_AUTH=OCI_TOKEN
OCI_IAM_URL=https://auth.usregion-1.example.com/v1/actions/
generateScopedAccessBearerToken
OCI_TENANCY=ocid1.tenancy..12345

In these examples, the optional <code>OCI_COMPARTMENT</code> and <code>OCI_DATABASE</code> parameters are not specified and thus the entire tenancy is set as the scope of the token request.

Related Topics

- Oracle Database Security Guide
- OCI IAM URL

Use the <code>OCI_IAM_URL</code> parameter to specify an endpoint URL that the database client must connect with to get the database token for authenticating Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM) users on OCI Database as a Service (DBaaS).

OCI TENANCY

Use the $\texttt{OCI_TENANCY}$ parameter to specify Oracle Cloud Identifier (OCID) of the user's tenancy.

OCI COMPARTMENT

Use the OCI_COMPARTMENT parameter to specify Oracle Cloud Identifier (OCID) of the compartment that holds database instances for client connections.

OCI DATABASE

Use the <code>OCI_DATABASE</code> parameter to specify Oracle Cloud Identifier (OCID) of the database that you want to access for the client connection.

6.10.9 SECURITY

Use the SECURITY parameter to change the security properties of a connection.

Purpose

To change the security properties of a connection.

Usage Notes

Put this parameter under the DESCRIPTION parameter. SECURITY permits additional parameters as listed in Security Section.

Example

6.10.10 SSL DISABLE WEAK EC CURVES

Use the SSL_DISABLE_WEAK_EC_CURVES parameter to disable the use of weak Elliptic Curve Cryptography (ECC) curves.

Purpose

To disable the use of weak ECC curves with key length less than 256 bits. You can set this parameter in the database server (sqlnet.ora), client (sqlnet.ora or tnsnames.ora connect string), or the listener (listener.ora).

Usage Notes

By default, this parameter is set to FALSE to enable the use of all ECC curves. If you want to enable the use of only Oracle approved curves with ECC curve key size of 256 bits or higher, then set this parameter to TRUE.

When set to TRUE, you can use only the following ECC curves:

- secp256r1
- secp384r1
- secp521r1

Values

- TRUE | ON | YES | 1: To enable only the Oracle approved ECC curves with minimum ECC curve key length of 256 bits
- FALSE | OFF | NO | 0: To enable all ECC curves



Default

FALSE

Examples

• In the thshames.ora file:

```
net_service_name=
  (DESCRIPTION=
      (ADDRESS=(PROTOCOL=tcps) (HOST=sales-svr) (PORT=1521))
      (SECURITY=(SSL_DISABLE_WEAK_EC_CURVES=TRUE))
)
```

• In the sqlnet.ora file or the listener.ora file:

```
SSL DISABLE WEAK EC CURVES=TRUE
```

6.10.11 SSL_SERVER_CERT_DN

Use the SSL_SERVER_CERT_DN parameter to specify the distinguished name (DN) of the database server.

Purpose

To specify the distinguished name (DN) of the database server.

Usage Notes

The server DN must be known by the client ahead of time. Otherwise, the client cannot specify the server's DN in <code>SSL_SERVER_CERT_DN</code>. The client uses this information to obtain the list of DNs it expects for each of the servers, enforcing the database server DN to match its service name. This parameter must be set to the server DN (for example,

SSL_SERVER_CERT_DN="finance, cn=OracleContext,c=us,o=example") to use full DN matching. For partial DN matching, do not include this parameter.

Use this parameter with the sqlnet.ora parameter SSL_SERVER_DN_MATCH to enable full DN matching.

Example



See Also:

Oracle Database Security Guide

6.10.12 SSL_SERVER_DN_MATCH

Use the SSL_SERVER_DN_MATCH parameter to enforce server-side certificate validation through distinguished name (DN) matching.

Purpose

To enforce server-side certification validation through distinguished name (DN) matching.

Usage Notes

If you enforce the DN matching, in addition to verifying the server's certificate chain, the client performs another check through DN matching.

You can configure either partial DN matching or full DN matching. Partial DN matching occurs if the server's CN contains its host name. Full DN matching occurs against the server's complete DN. Not enforcing the match allows the server to potentially fake its identity.

In addition to the sqlnet.ora file, configure the tnsnames.ora parameter SSL_SERVER_CERT_DN to enable full DN matching.

Default

NO

Values

YES | ON | TRUE | 1:

To enforce partial or full DN matching. If the DN matches the service name, then the connection succeeds. If the DN does not match the service name, then the connection fails.

NO | OFF | FALSE | 0:

To not enforce DN matching. If the DN does not match the service name, then the connection is successful, but an error is logged to the sqlnet.log file.

Example

SSL_SERVER_DN_MATCH=YES

Related Topics

Oracle Database Security Guide



6.10.13 SSL_VERSION

Use the SSL_VERSION parameter to define valid Transport Layer Security (TLS) versions to be used for connections.

Purpose

To define the version of TLS that must run on the systems with which the database server communicates. By default, the database server and client negotiate the strongest security protocol. Oracle does not recommend modifying this parameter, unless your security requirements mandate the usage of certain protocol versions.

Usage Notes

Clients, listeners, and database servers must use compatible versions. Modify this parameter only when necessary to enforce the use of the more secure TLS protocol and not allow clients that only work with the older TLS protocols. If you need to specify TLS 1.0 or TLS 1.1, then also include TLS 1.2 to allow more secure connections. The current default uses TLS 1.2, which is the version required for multiple security compliance requirements.

If you set SSL_VERSION to undetermined, then the most secure TLS protocol version is used. You can use the SSL_VERSION=undetermined setting in the connect string for a specific connection to override the SSL_VERSION value configured in the sqlnet.ora file.

If you do not set SSL_VERSION to any value, then all the supported TLS protocol versions are tried starting with the most secure version. This is typically the most common configuration, ensuring that the strongest protocol is chosen during TLS negotiation.

Default

undetermined

Values

```
undetermined | 1.0 | 1.1 | 1.2
```

The version numbers correspond to the TLS versions, such as TLSv1.0, TLSv1.1, and TLSv1.2.



The sqlnet.ora parameter ADD_SSLV3_TO_DEFAULT has no impact on this parameter.

Syntax and Examples

To specify a single TLS version:

```
SSL VERSION=TLS protocol version
```

For example:

SSL VERSION=1.2



To specify multiple TLS versions, use the or operator as follows:

```
SSL VERSION=TLS protocol version1 or TLS protocol version2
```

For example:

```
SSL_VERSION=1.1 or 1.2

SSL_VERSION=1.0 or 1.1 or 1.2
```

Related Topics

- · Set the Required TLS Version on the Server
- Set the Required TLS Version on the Client

6.10.14 TOKEN_AUTH

Use the TOKEN_AUTH parameter to configure token-based authentication for Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM) or Microsoft Azure users of Microsoft Entra ID (previously called Microsoft Azure Active Directory).

Purpose

Token-based access enforces strong authentication, which enables a more secure access to the database. IAM users can connect to OCI Database as a Service (DBaaS) databases, and Azure users can connect to Oracle Databases (cloud or on-premises).

With this setting, the database client looks for a token file when a / (slash) login is used.

Use this parameter under the SECURITY section of the tnsnames.ora file, sqlnet.ora file, or directly as part of the command-line connect string. The parameter value specified in the connect string takes precedence over the other specified values.

Usage Notes for IAM

OCI IAM token:

An OCI IAM token (db-token), which is obtained from IAM using Oracle Cloud Infrastructure (OCI) Command Line Interface (CLI) or programmatically from the OCI Software Development Kit (SDK), is a proof-of-possession (PoP) token with an expiration time and scope.

You can use one of the IAM user credentials, such as API-key, security token, resource principal, instance principal, or delegation token to retrieve the db-token and private key from IAM.

These tokens are transmitted over secure channels. You must use only the TCP/IP with Transport Layer Security (TLS) protocol, otherwise an error message appears indicating that non-TLS connections are disallowed.

Required setting for token-based authentication:

You must configure the TCPS protocol (PROTOCOL=tcps) and set the SSL_SERVER_DN_MATCH parameter to TRUE for token-based authentication.

Use a file location to send the token to Oracle Database:



When an IAM user logs in using <code>/@connect_identifier</code> (and <code>TOKEN_AUTH</code> is set to <code>OCI_TOKEN</code>), the <code>TOKEN_AUTH=OCI_TOKEN</code> setting along with <code>/@connect_identifier</code> instructs the database client to get the <code>db-token</code> and private key from either the default directory or the location specified by <code>TOKEN_LOCATION</code> (using IAM token-based authentication).

Use the client API to send the token to Oracle Database:

If your client application is updated to retrieve tokens from IAM, then you can override the <code>TOKEN_AUTH=OCI_TOKEN</code> setting. The client application gets the <code>db-token</code> and private key from IAM and sends as attributes to the database client using the client API. In this case, you do not need to specify the <code>TOKEN_AUTH</code> and <code>TOKEN_LOCATION</code> parameters.

General IAM token-based authentication process:

 An IAM user or application in OCI first requests the db-token from IAM by using APIkey, security token, resource principal, service principal, instance principal, or delegation token (delegation token is available only in the Cloud Shell).

To use a security token, you need to generate it by completing the browser authentication process and then request the <code>db-token</code> using that security token. If the IAM policy that authorizes you to be issued the <code>db-token</code> exists, then the <code>db-token</code> is returned.

You request the db-token using OCI CLI (or OCI SDK for applications). For example, run the following OCI CLI command to request the db-token by using an API-key (apikey):

```
$ oci iam db-token get --profile scott
```

The profile option specifies the profile for which you want to access the IAM user credentials and retrieve the db-token.

For more information on using OCI CLI, see the get command details in *Oracle Cloud Infrastructure CLI Command Reference*.

- 2. OCI CLI references the config file (that stores your IAM user credentials as part of the profile) and makes a call to IAM to get the db-token. The db-token and private key files are written at the default or specified token location.
- 3. You can specify the TOKEN_LOCATION parameter to override the default directory where the db-token and private key files are stored.

The database client gets the <code>db-token</code> and private key from the default token location or the location specified by <code>TOKEN_LOCATION</code>, signs the <code>db-token</code> with the private key and sends it to the database server. The database server verifies the <code>db-token</code> and gets the group membership information for the user. If the IAM user is mapped to a database schema (exclusively or shared), then the login is completed.





You can also use another IAM credential, IAM database password, to request the <code>dbtoken</code> from IAM. This <code>dbtoken</code> is a bearer token and does not come with a private key. You can configure the database client to request this token using your IAM user name and IAM database password. An application cannot pass this type of <code>db-token</code> to the client. In this case, you use a different parameter setting (<code>PASSWORD AUTH=OCI TOKEN</code>).

Unlike the API-key, security token, resource principal, service principal, instance principal, and delegation token that require an application or tool to get a token, the IAM database password can only be used by the database client to retrieve the token. See PASSWORD_AUTH.

Table 6-1 Values and Examples for IAM

Default	Value	Example
None	TOKEN_AUTH=OCI_TOKEN	In the tnsnames.ora file:
		<pre>net_service_name= (DESCRIPTION = (ADDRESS=(PROTOCOL=tcps) (HOST=sales-svr) (PORT=1521)) (SECURITY=</pre>
		<pre>(SSL_SERVER_CERT_DN="C=US,O=example,CN=Oracle Context")</pre>
		<pre>(CONNECT_DATA=(SERVICE_NAME=sales.us.example. com)))</pre>
		In the sqlnet.ora file:
		SSL_SERVER_DN_MATCH=TRUE TOKEN_AUTH=OCI_TOKEN
		In these examples, the optional <code>TOKEN_LOCATION</code> parameter is not specified. Thus, the client automatically gets the <code>db-token</code> and private key from the default token location.

Usage Notes for Entra ID

Entra ID access token:

An Entra ID OAuth2 access token is a bearer token with an expiration time and scope. This token follows the OAuth2.0 standard with Entra ID extensions. You can request these tokens from tools and scripts run on Linux, Microsoft PowerShell, or other environments. You can also request these tokens programmatically using the Microsoft SDKs.

These tokens are transmitted over secure channels. You must use only the TCP/IP with Transport Layer Security (TLS) protocol, otherwise an error message appears indicating that non-TLS connections are disallowed.

· Required setting for token-based authentication:

You must configure the TCPS protocol (PROTOCOL=tcps) and set the SSL_SERVER_DN_MATCH parameter to TRUE for token-based authentication.

Use a file location to send the token to Oracle Database:

When an Azure user logs in using <code>/@connect_identifier</code>, then the <code>TOKEN_AUTH=OAUTH</code> setting instructs the database client to get the access token from the directory location specified by <code>TOKEN_LOCATION</code> if the token file is named <code>token</code>. If the token file name is different from <code>token</code>, then you must use the file name along with the directory location while specifying the <code>TOKEN_LOCATION</code> parameter.

The TOKEN_LOCATION parameter is mandatory for Azure token-based authentication. The database client gets the token from this location and sends it to the database server.

Use the client API to send the token to Oracle Database:

If your client application is updated to retrieve tokens from Entra ID, then you can override the <code>TOKEN_AUTH=OAUTH</code> setting. Entra ID directly passes the <code>db-token</code> as an attribute to the database client using the client API. When the client receives this request, the client sends it to the database server.

In this case, you do no need to specify the TOKEN AUTH and TOKEN LOCATION parameters.

General Azure token-based authentication process:

 An Azure user or application first requests the access token from Entra ID using one of the supported authentication flows (resource owner password credentials, authorization code, on-behalf-of (OBO) flow, or client credentials).

An Azure user can connect using any supported utility to retrieve the token and store it in a local file directory.

You can request the token from tools and scripts run on Linux, Microsoft PowerShell, or other environments. You can also request programmatically using the Microsoft SDKs.

For detailed examples on how to retrieve an Entra ID OAuth2 access token, see *Oracle Database Security Guide*.

2. The database client then sends the token to the database server. The database server verifies the token (using the Entra ID public key) and extracts various claims from the token, including user name, app roles, and audience. If the Entra ID principal is mapped to a database schema (exclusively or shared), then the login is completed.



Table 6-2 Values and Examples for Entra ID

Default	Value	Example
None	If the token file is named token:	In the tnsnames.ora file:
	TOKEN_AUTH=OAUTH TOKEN_LOCATION="token_f ile_directory"	<pre>net_service_name= (DESCRIPTION= (ADDRESS=(PROTOCOL=tcps) (HOST=salesserver1) (PORT=1522)) (SECURITY= (SSL_SERVER_DN_MATCH=TRUE)</pre>
		<pre>(SSL_SERVER_CERT_DN="C=US,O=example,CN=Oracle Context")</pre>
		In the sqlnet.ora file:
		SSL_SERVER_DN_MATCH=TRUE TOKEN_AUTH=OAUTH TOKEN_LOCATION="/home/dbuser1/access-token"
		In these examples, the token file name is token. Thus, only the directory path (/home/dbuser1/access-token) is specified. The client automatically looks for the token file in the specified path and gets the access token.



Table 6-2 (Cont.) Values and Examples for Entra ID

Default	Value	Example
	If the token file name is different from token:	In the tnsnames.ora file:
	TOKEN_AUTH=OAUTH TOKEN_LOCATION="token_f ile_directory/ token_filename"	<pre>net_service_name= (DESCRIPTION= (ADDRESS=(PROTOCOL=tcps) (HOST=salesserver1) (PORT=1522)) (SECURITY= (SSL_SERVER_DN_MATCH=TRUE)</pre>
		<pre>(SSL_SERVER_CERT_DN="C=US,O=example,CN=Oracle Context")</pre>
		com))
		In the sqlnet.ora file:
		SSL_SERVER_DN_MATCH=TRUE TOKEN_AUTH=OAUTH TOKEN_LOCATION="/home/dbuser1/access-token/ mytoken"
		In these examples, the token file name is mytoken. Thus, both the file name and directory path (/home/dbuser1/access-token) are specified. The client gets the access token from the mytoken file in the specified path.

Related Topics

- Authenticating and Authorizing IAM Users for Oracle DBaaS Databases
- Authenticating and Authorizing Microsoft Azure Users for Oracle Databases
- TOKEN_LOCATION
 Use the TOKEN_LOCATION parameter to specify the directory location where token file is stored for token-based authentication.

6.10.15 TOKEN_LOCATION

Use the ${\tt TOKEN_LOCATION}$ parameter to specify the directory location where token file is stored for token-based authentication.

Purpose

To specify the token file directory location. You use this parameter while configuring tokenbased authentication for Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM) or Microsoft Azure users of Microsoft Entra ID. The database client gets the token from this location and sends it to the database server. For Entra ID, you can also specify the token file name along with the directory location.

Use this parameter along with the <code>TOKEN_AUTH</code> parameter in the <code>tnsnames.ora</code> file, <code>sqlnet.ora</code> file, or directly as part of the command-line connect string. The parameter values specified in the connect string take precedence over the other specified values.

Usage Notes for IAM

The <code>TOKEN_LOCATION</code> parameter is optional for IAM token-based authentication. You can use this parameter along with the <code>TOKEN_AUTH</code> parameter to override the default directory where the <code>db-token</code> and private key are stored. This location is used by the database client to retrieve the <code>db-token</code> and private key.

When an IAM user initiates a connection using <code>/@connect_identifier</code> (and <code>TOKEN_AUTH</code> is set to <code>OCI_TOKEN</code>), the database client retrieves the <code>db-token</code> and private key from either the default directory or the location specified by <code>TOKEN_LOCATION</code>. The client then signs the <code>db-token</code> using the private key and sends the <code>db-token</code> to the database server.

Default Setting for IAM

On Linux:

/home/username/.oci/db-token

On Windows:

The database client searches for the default directory in this order:

If the USERPROFILE environment variable is set, then the client searches in the USERPROFILE directory (for example, C:\Users\username).

If USERPROFILE is not set, then the client searches in HOMEDRIVE directory (for example, C:) with HOMEPATH (for example, \Users\username).

For example, the default token location directory on Windows is:

C:\Users\username\.oci\db-token



Values and Examples for IAM

Value	Example
TOKEN_LOCATION="token_file_directory"	In the tnsnames.ora file:
	<pre>net_service_name= (DESCRIPTION = (ADDRESS=(PROTOCOL=tcps) (HOST=sales-svr) (PORT=1521)) (SECURITY= (SSL_SERVER_DN_MATCH=TRUE)</pre>
	<pre>(SSL_SERVER_CERT_DN="C=US,O=example,CN=OracleContext")</pre>
	<pre>(CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)))</pre>
	In the sqlnet.ora file:
	SSL_SERVER_DN_MATCH=TRUE TOKEN_AUTH=OCI_TOKEN TOKEN_LOCATION="/home/oracle/.oci/db-token"

Usage Notes for Entra ID

The TOKEN_LOCATION parameter is mandatory for Azure token-based authentication. You must use this parameter along with the TOKEN_AUTH parameter to specify the directory location where the Entra ID OAuth2 access token is stored. This location is used by the database client to get the access token.

If your token file is named token, then specify only the directory path. If the token file name is different from token, then you must use the file name along with the directory path.

When an Azure user initiates a connection using $/@connect_identifier$, the database client retrieves the access token from the location specified by <code>TOKEN_LOCATION</code> and sends the token to the database server.

Default Setting for Entra ID

None



Values and Examples for Entra ID

Value	Example
If the token file is named token:	In the tnsnames.ora file:
TOKEN LOCATION="token file directory"	
	net_service_name=
	(DESCRIPTION=
	(ADDRESS=(PROTOCOL=tcps) (HOST=salesserver1)
	(PORT=1522))
	(SECURITY=
	(SSL_SERVER_DN_MATCH=TRUE)
	(SSL_SERVER_CERT_DN="C=US,O=example,CN=OracleContext")
	(TOKEN AUTH=OAUTH) (TOKEN LOCATION="/home/
	dbuser1/access-token"))
	<pre>(CONNECT_DATA=(SERVICE_NAME=sales.us.example.com)))</pre>
	In the sqlnet.ora file:
	SSL SERVER DN MATCH=TRUE
	TOKEN AUTH=OAUTH
	TOKEN_LOCATION="/home/dbuser1/access-token"
	In these examples, the token file name is token. Thus, only the directory path (/home/dbuser1/access-token) is specified. The client automatically looks for the token file in the specified path and gets the access token.



Value	Example
If the token file name is different from token:	In the tnsnames.ora file:
TOKEN_LOCATION="token_file_directory/token_filename"	<pre>net_service_name= (DESCRIPTION=</pre>
	(SECURITY= (SSL_SERVER_DN_MATCH=ON)
	<pre>(SSL_SERVER_CERT_DN="C=US,O=example,CN=OracleContext")</pre>
	In the sqlnet.ora file:
	SSL_SERVER_DN_MATCH=TRUE TOKEN_AUTH=OAUTH TOKEN_LOCATION="/home/dbuser1/access-token/mytoken"
	In these examples, the token file name is mytoken. Thus, both the file name and directory path (/home/dbuser1/access-token) are specified. The client gets the access token from the mytoken file in the specified path.

Related Topics

- Authenticating and Authorizing IAM Users for Oracle DBaaS Databases
- Authenticating and Authorizing Microsoft Azure Users for Oracle Databases
- TOKEN AUTH

Use the <code>TOKEN_AUTH</code> parameter to configure token-based authentication for Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM) or Microsoft Azure users of Microsoft Entra ID (previously called Microsoft Azure Active Directory).

6.10.16 WALLET_LOCATION

Use the ${\tt WALLET_LOCATION}$ parameter in the ${\tt tnsnames.ora}$ file to specify different locations where Oracle wallets are stored.

Purpose

This parameter denotes a connection specific wallet. You can use this parameter when different connections need to use different wallets on the client side.



Usage Notes

You can use WALLET_LOCATION in both the sqlnet.ora file and tnsnames.ora file. Use of WALLET_LOCATION in tnsnames.ora overrides the WALLET_LOCATION in sqlnet.ora for the specific tnsnames.ora service.

WALLET_LOCATION enables a client connection to have distinct TLS connections that use certificates. This means that the clients will initiate multiple TLS connections using different TLS certificates, in the same client process.

Use this parameter if you have a single client that must rely on more than one TLS session. An example would be for a client that requires access to multiple pluggable databases (PDBs), each with its own identity (certificate). This feature enables you to configure the client to connect to the correct identity for each PDB. After the configuration is complete, multi-threaded clients are able to access more than one wallet with different certificates in simultaneous TLS sessions.

Examples

```
net service name=
 (DESCRIPTION=
   (ADDRESS LIST=
     (ADDRESS=(PROTOCOL=tcp) (HOST=sales1-svr) (PORT=1521))
     (ADDRESS=(PROTOCOL=tcp) (HOST=sales2-svr) (PORT=1521)))
   (CONNECT DATA=
     (SERVICE NAME=sales.us.example.com))
      (SECURITY=(wallet location=/home/oracle/wallets/databases)))
ssl certs1 =
    (DESCRIPTION =
       (ADDRESS=(PROTOCOL=tcps)(HOST=shobeen.us.example.com) (PORT=1750))
       (CONNECT DATA=(SID=sales pdb))
       (SECURITY=(WALLET LOCATION=/oracle/wallets/certificates/sales cert))
ssl certs2 =
    (DESCRIPTION =
       (ADDRESS=(PROTOCOL=tcps)(HOST=shobeen.us.example.com) (PORT=1750))
       (CONNECT DATA=(SID=marketing pdb))
       (SECURITY=(WALLET LOCATION=/oracle/wallets/certificates/
marketing cert))
     )
```

Related Topics

- Oracle Database Security Guide
- Oracle Database Net Services Reference

6.11 Timeout Parameters

The timeout section of the tnsnames.ora file provides the ability to specify timeout and retry configuration through the TNS connect string.

The following parameters can be set at the DESCRIPTION level of a connect string:

CONNECT_TIMEOUT

Use the tnsnames.ora parameter CONNECT_TIMEOUT to specify the amount of time, in milliseconds, seconds, or minutes, in which clients must establish Oracle Net connections to database instances.

RETRY_COUNT

Use the tnsnames.ora parameter RETRY_COUNT to specify the number of times an ADDRESS list is traversed before terminating the connection attempt.

RETRY_DELAY

Use the tnsnames.ora parameter RETRY_DELAY to specify the delay in seconds between connection retries.

TRANSPORT_CONNECT_TIMEOUT

Use the tnsnames.ora parameter TRANSPORT_CONNECT_TIMEOUT to specify the transport connect timeout duration, in milliseconds, seconds, or minutes.

RECV TIMEOUT

Use the tnsnames.ora parameter RECV_TIMEOUT to specify the duration of time that a database client or server should wait for data from a peer after establishing a connection.

6.11.1 CONNECT_TIMEOUT

Use the thinnames.ora parameter CONNECT_TIMEOUT to specify the amount of time, in milliseconds, seconds, or minutes, in which clients must establish Oracle Net connections to database instances.

Purpose

To specify the timeout duration in ms or msec (milliseconds), sec (seconds), or min (minutes) for a client to establish an Oracle Net connection to Oracle Database.

Usage Notes

- Put this parameter under the DESCRIPTION parameter.
- In case, no unit is mentioned, the default unit is sec.

It accepts different timeouts with or without space between the value and the unit. For example:

```
CONNECT_TIMEOUT=10 ms
```

or

CONNECT TIMEOUT=10ms

The timeout interval specified by CONNECT_TIMEOUT is a superset of the TCP connect
timeout interval. It includes the time to be connected to the database instance providing the
requested service, not just the duration of the TCP connection.

The timeout interval is applicable to each ADDRESS in an ADDRESS_LIST, and each IP address to which a host name is mapped.

• The CONNECT_TIMEOUT parameter is equivalent to the sqlnet.ora parameter SQLNET.OUTBOUND CONNECT TIMEOUT, and overrides it.

Examples

```
net_service_name=
  (DESCRIPTION=
        (CONNECT_TIMEOUT=10 ms) (RETRY_COUNT=3)
        (ADDRESS_LIST=
              (ADDRESS=(PROTOCOL=tcp) (HOST=sales1-svr) (PORT=1521))
              (ADDRESS=(PROTOCOL=tcp) (HOST=sales2-svr) (PORT=1521)))
        (CONNECT_DATA=
              (SERVICE NAME=sales.us.example.com)))
```

Related Topics

SQLNET.OUTBOUND_CONNECT_TIMEOUT

Use the sqlnet.ora parameter SQLNET.OUTBOUND_CONNECT_TIMEOUT to specify the amount of time, in milliseconds, seconds, or minutes, in which clients must establish Oracle Net connections to database instances.

6.11.2 RETRY_COUNT

Use the tnsnames.ora parameter RETRY_COUNT to specify the number of times an ADDRESS list is traversed before terminating the connection attempt.

Purpose

To specify the number of times an ADDRESS list is traversed before the connection attempt is terminated.

Usage Notes

Put this parameter under the DESCRIPTION parameter.

When a DESCRIPTION_LIST is specified, each DESCRIPTION is traversed multiple times based on the specified number of retries.

Example



6.11.3 RETRY_DELAY

Use the tnsnames.ora parameter RETRY_DELAY to specify the delay in seconds between connection retries.

Purpose

To specify the delay in seconds between subsequent retries for a connection. This parameter works in conjunction with RETRY COUNT parameter.

Usage Notes

Put this parameter under the DESCRIPTION parameter.

When a DESCRIPTION_LIST is specified, each DESCRIPTION is traversed multiple times based on the specified number of retries, and the specific delay for the description.

Example

6.11.4 TRANSPORT_CONNECT_TIMEOUT

Use the tnsnames.ora parameter TRANSPORT_CONNECT_TIMEOUT to specify the transport connect timeout duration, in milliseconds, seconds, or minutes.

Purpose

To specify the transport connect timeout duration in ms or msec (milliseconds), sec (seconds), or min (minutes) for a client to establish an Oracle Net connection to Oracle Database.

Usage Notes

- This parameter is put under the DESCRIPTION parameter.
- The default value is 60 sec. In case, no unit is mentioned, the default unit is sec.

It accepts different timeouts with or without space between the value and the unit.

```
TRANSPORT_CONNECT_TIMEOUT=10 ms

Or

TRANSPORT CONNECT TIMEOUT=10ms
```



The timeout interval is applicable for each ADDRESS in an ADDRESS_LIST description, and
each IP address that a host name is mapped. The TRANSPORT_CONNECT_TIMEOUT parameter
is equivalent to the sqlnet.ora parameter TCP.CONNECT_TIMEOUT, and overrides it.

Example

```
net_service_name =
  (DESCRIPTION=
    (TRANSPORT_CONNECT_TIMEOUT=10 ms)
    (ADDRESS_LIST=
         (ADDRESS=(PROTOCOL=tcp) (HOST=sales1-svr) (PORT=1521))
         (ADDRESS=(PROTOCOL=tcp) (HOST=sales2-svr) (PORT=1521)))
    (CONNECT_DATA=
         (SERVICE NAME=sales.us.example.com)))
```

Related Topics

TCP.CONNECT_TIMEOUT

6.11.5 RECV_TIMEOUT

Use the tnsnames.ora parameter RECV_TIMEOUT to specify the duration of time that a database client or server should wait for data from a peer after establishing a connection.

Purpose

To specify the time duration in ms or msec (milliseconds), sec (seconds), min (minutes), or hr (hours) for a database client or server to wait for data from the peer after establishing a connection. The peer must send data within the time interval that you specify.

Usage Notes

- This parameter is put under the DESCRIPTION parameter.
- If you do not specify a unit of measurement, then the default unit is sec.

It accepts time duration with or without space between the value and the unit. For example:

```
Or

RECV_TIMEOUT=10 ms
```

Setting this parameter for clients ensures that receive operations are not left in a wait state
indefinitely or for a long period due to server host being down, server busy state, or
network connectivity issues.

If a client does not receive response data in the time specified, then the client logs ORA-12535: TNS:operation timed out and ORA-12609: TNS: Receive timeout occurred messages to the sqlnet.log file.

Default Value

None

Minimum Value

1 ms



Allowed Range

Any number greater than the minimum value of 1 ms up to 4294967295 ms.

Examples

```
net_service_name=
  (DESCRIPTION=
    (CONNECT_TIMEOUT=10ms) (RETRY_COUNT=3) (RECV_TIMEOUT=10ms)
    (ADDRESS_LIST=
        (ADDRESS=(PROTOCOL=tcp) (HOST=sales1-svr) (PORT=1521))
        (ADDRESS=(PROTOCOL=tcp) (HOST=sales2-svr) (PORT=1521)))
        (CONNECT_DATA=
        (SERVICE_NAME=sales.us.example.com)))
```

Related Topics

Oracle Database Net Services Administrator's Guide

6.12 Compression Parameters

The compression section of the tnsnames.ora file provides the ability to enable compression and specify compression levels. These parameters can be set at the DESCRIPTION level of a connect string.

COMPRESSION

The tnsnames.ora file's compression parameter enables or disables the data compression.

COMPRESSION_LEVELS

The COMPRESSION_LEVELS parameter of the tnsnames.ora file specifies the compression level.

6.12.1 COMPRESSION

The tnsnames or a file's compression parameter enables or disables the data compression.

Purpose

To enable or disable data compression.

Usage Notes

Put this parameter under the DESCRIPTION parameter.

Setting this parameter in the connect descriptor for a client overrides the SQLNET.COMPRESSION parameter in the client-side sqlnet.ora file.

Default

off

Values

- on to enable data compression.
- off to disable data compression.



Example

Related Topics

SQLNET.COMPRESSION

6.12.2 COMPRESSION LEVELS

The COMPRESSION LEVELS parameter of the tnsnames.ora file specifies the compression level.

Purpose

To specify the compression level.

Usage Notes

The compression levels are used at the time of negotiation to verify which levels are used at both ends, and select one level. Put this parameter under the DESCRIPTION parameter.

This parameter is used with the COMPRESSION parameter. Setting this parameter in the connect descriptor for a client overrides the SQLNET.COMPRESSION_LEVELS parameter in the client-side sqlnet.ora file.

Default

low

Values

- low for low CPU usage and a low compression ratio.
- high for high CPU usage and a high compression ratio.

Example

```
net_service_name=
  (DESCRIPTION=
    (COMPRESSION=on)
    (COMPRESSION_LEVELS=(LEVEL=low) (LEVEL=high))
    (ADDRESS_LIST=
          (ADDRESS=(PROTOCOL=tcp) (HOST=sales1-server) (PORT=1521))
          (ADDRESS=(PROTOCOL=tcp) (HOST=sales2-server) (PORT=1521)))
    (CONNECT_DATA=
          (SERVICE_NAME=sales.us.example.com)))
```



Related Topics

SQLNET.COMPRESSION_LEVELS



7

Oracle Net Listener Parameters in the listener.ora File

This chapter provides a complete listing of the listener.ora file configuration parameters.

- Overview of Oracle Net Listener Configuration File
 Oracle Net Listener configuration, stored in the listener.ora file, consists of these elements.
- Protocol Address Parameters
- Connection Rate Limiter Parameters

The connection rate limiter feature in Oracle Net Listener enables a database administrator to limit the number of new connections handled by the listener. When this feature is enabled, Oracle Net Listener imposes a user-specified maximum limit on the number of new connections handled by the listener every second. Depending on the configuration, the rate can be applied to a collection of endpoints, or to a specific endpoint.

Control Parameters

This section describes the following parameters that control the behavior of the listener:

- ADR Diagnostic Parameters for Oracle Net Listener
 The diagnostic data for the critical errors is quickly captured and stored in the ADR for Oracle Net listener.
- Non-ADR Diagnostic Parameters for Oracle Net Listener

 This section lists the parameters used when ADR is disabled. The default value of

 DIAG_ADR_ENABLED_listener_name is on. Therefore, the

 DIAG_ADR_ENABLED_listener_name parameter must explicitly be set to off to use non-ADR tracing.
- Class of Secure Transports Parameters
 The class of secure transports (COST) parameters specify a list of transports that are considered secure for administration and registration of a particular listener.

7.1 Overview of Oracle Net Listener Configuration File

Oracle Net Listener configuration, stored in the listener.ora file, consists of these elements.

- Name of the listener
- Protocol addresses that the listener is accepting connection requests on
- Valid nodes that the listener allows to register with the database
- Database services
- Control parameters

Dynamic service registration, eliminates the need for static configuration of supported services. However, static service configuration is required if you plan to use Oracle Enterprise Manager Cloud Control. For information about static service configuration, see *Oracle Database Net Services Administrator's Guide*.

By default, the listener.ora file is located in the <code>ORACLE_HOME/network/admin</code> directory. The listener.ora file can also be stored the following locations:

- The directory specified by the TNS ADMIN environment variable or registry value.
- On Linux and UNIX operating systems, it is the global configuration directory. For example, on the Oracle Solaris operating system, the directory is /var/opt/oracle. See Oracle Database Global Data Services Concepts and Administration Guide for information about management of global services. Also refer to Oracle operating system-specific documentation.
- In the read-only Oracle home mode, the listener.ora file default location is ORACLE BASE HOME/network/admin.
- In the read-only Oracle home mode, the parameters that default to ORACLE_HOME location change to default to ORACLE_BASE_HOME location.

It is possible to configure multiple listeners, each with a unique name, in one listener.ora file. Multiple listener configurations are possible because each of the top-level configuration parameters has a suffix of the listener name or is the listener name itself.

Note:

- It is often useful to configure multiple listeners in one listener.ora file. However, Oracle recommends running only one listener for each node in most customer environments.
- Oracle Net Services supports the IFILE parameter in the listener.ora file, with up to three levels of nesting. The parameter is added manually to the file. The following is an example of the syntax:

```
IFILE=/tmp/listener_em.ora
IFILE=/tmp/listener_cust1.ora
IFILE=/tmp/listener_cust2.ora
```

Refer to *Oracle Database Reference* for additional information.

Example 7-1 shows a listener.ora file for a listener named LISTENER, which is the default name of the listener.

Example 7-1 listener.ora File

7.2 Protocol Address Parameters

The protocol address section of the listener.ora file defines the protocol addresses on which the listener is accepting connection requests. This section describes the most common parameters used in protocol addresses. The ADDRESS_LIST parameter is also supported. This section lists and describes the following parameters:

ADDRESS

The protocol ADDRESS parameter's networking parameter is in the listener.ora file. It specifies the protocol address under the DESCRIPTION parameter for one listener.

DESCRIPTION

DESCRIPTION networking parameter of the listener.ora file contains listener protocol addresses.

- Firewall
- IP

The protocol address parameter ${\tt IP}$ determine which IP address the listener listens on when a host name is specified

- QUEUESIZE
- RECV_BUF_SIZE

Use the RECV BUF SIZE parameter to specify buffer space for session receive operations.

SEND_BUF_SIZE

Use the <code>SEND BUF SIZE</code> parameter to specify buffer space for session send operations.

7.2.1 ADDRESS

The protocol ADDRESS parameter's networking parameter is in the listener.ora file. It specifies the protocol address under the DESCRIPTION parameter for one listener.

Purpose

Specifies a single listener protocol address in the DESCRIPTION parameter

Usage Notes

Use this parameter to define the protocol, the host, and the port number for the listener.

Example

```
listener_name=
(DESCRIPTION=
  (ADDRESS_LIST=
          (ADDRESS=(PROTOCOL=tcp) (HOST=hr-server) (PORT=1521))
          (ADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=1521))))
```

7.2.2 DESCRIPTION

DESCRIPTION networking parameter of the listener.ora file contains listener protocol addresses.

Purpose

To contain listener protocol addresses.

Example 7-2 Example

listener_name

7.2.3 Firewall

Purpose

It can be set in endpoint to enable firewall functionality.

Related Topics

Oracle Database Net Services Administrator's Guide

7.2.4 IP

The protocol address parameter IP determine which IP address the listener listens on when a host name is specified

Purpose

To determine which IP address the listener listens on when a host name is specified.

Usage Notes

This parameter is only applicable when the HOST parameter specifies a host name.

Values

first

Listen on the first IP address returned by the DNS resolution of the host name. If the user wants the listener to listen on the first IP to which the specified host name resolves, then the address must be qualified with (IP=first).

v4_only

Listen only on IPv4 addresses.

v6_only

Listen only on IPv6 addresses.

Default

This feature is disabled by default.

Example

```
listener_name=
(DESCRIPTION=
   (ADDRESS=(PROTOCOL=tcp) (HOST=rancode1-vip) (PORT=1522) (IP=v6_only))
```

7.2.5 QUEUESIZE

Purpose

To specify the number of concurrent connection requests that the listener can accept on a TCP/IP or IPC listening endpoint (protocol address).



Usage Notes

The number of concurrent connection requests is dependent on the platform and listener usage scenarios. If the listener is heavily-loaded, then set the parameter to a higher number.

Put this parameter at the end of the protocol address with its value set to the expected number of concurrent connection requests.

Default

The default number of concurrent connection requests is operating system specific.

Example

```
listener_name=
(DESCRIPTION=
    (ADDRESS=(PROTOCOL=tcp)(HOST=hr-server)(PORT=1521)(QUEUESIZE=20)))
```



Oracle Database Net Services Administrator's Guide for additional information about configuring this parameter

7.2.6 RECV_BUF_SIZE

Use the RECV BUF SIZE parameter to specify buffer space for session receive operations.

Purpose

To specify, in bytes, the buffer space for receive operations of sessions.

Usage Notes

Put this parameter under the DESCRIPTION parameter or at the end of the protocol address with its value set to the expected number of bytes.

This parameter is supported by the TCP/IP, TCP/IP with TLS, and SDP protocols.



Additional protocols might support this parameter on certain operating systems. Refer to the operating system-specific documentation for information about additional protocols that support this parameter.

Default

The default value for this parameter is operating system specific. The default for the Linux operating system is 87380 bytes.



Example

Related Topics

Oracle Database Net Services Administrator's Guide

7.2.7 SEND_BUF_SIZE

Use the SEND BUF SIZE parameter to specify buffer space for session send operations.

Purpose

To specify, in bytes, the buffer space for send operations of sessions.

Usage Notes

Put this parameter under the DESCRIPTION parameter or at the end of the protocol address.

This parameter is supported by the TCP/IP, TCP/IP with TLS, and SDP protocols.



Additional protocols might support this parameter on certain operating systems. Refer to operating system-specific documentation for additional information about additional protocols that support this parameter.

Default

The default value for this parameter is operating system specific. The default for the Linux operating system is 16 KB.

Example



```
(ADDRESS_LIST=
  (ADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=1521)
  (ADDRESS=(PROTOCOL=ipc) (KEY=extproc))))
```

Related Topics

Oracle Database Net Services Administrator's Guide

7.3 Connection Rate Limiter Parameters

The connection rate limiter feature in Oracle Net Listener enables a database administrator to limit the number of new connections handled by the listener. When this feature is enabled, Oracle Net Listener imposes a user-specified maximum limit on the number of new connections handled by the listener every second. Depending on the configuration, the rate can be applied to a collection of endpoints, or to a specific endpoint.

This feature is controlled through the following listener.ora configuration parameters:

CONNECTION_RATE_listener_name

The CONNECTION_RATE_listener_name configuration parameter of thelistener.ora file specifies a global rate that is enforced across all listening endpoints that are rate-limited.

RATE LIMIT

The RATE_LIMIT configuration parameter of the listener.ora file indicates that a particular listening endpoint is rate-limited.

7.3.1 CONNECTION_RATE_listener_name

The CONNECTION_RATE_listener_name configuration parameter of the listener.ora file specifies a global rate that is enforced across all listening endpoints that are rate-limited.

Purpose

To specify a global rate that is enforced across all listening endpoints that are rate-limited.

Usage Notes

When this parameter is specified, it overrides any endpoint-level numeric rate values that might be specified.

Syntax

CONNECTION RATE listener name=number of connections per second

7.3.2 RATE_LIMIT

The RATE_LIMIT configuration parameter of the listener.ora file indicates that a particular listening endpoint is rate-limited.

Purpose

To indicate that a particular listening endpoint is rate-limited.

Usage Notes

The parameter is specified in the ADDRESS section of the listener endpoint configuration.

Syntax

```
LISTENER=
(ADDRESS=(PROTOCOL=tcp)(HOST=)(PORT=1521)(RATE LIMIT=yes))
```

- When the RATE_LIMIT parameter is set to yes for an endpoint, that endpoint is included in the enforcement of the global rate configured by the CONNECTION_RATE_listener_name parameter. The global rate limit is enforced individually at each endpoint that has RATE LIMIT set to yes.
- Dynamic endpoints for listeners managed by Oracle Clusterware have the RATE_LIMIT parameter set to yes.
- When the RATE_LIMIT parameter is set to a value greater than 0, then the rate limit is enforced at that endpoint level.

Examples

The following examples use the CONNECTION_RATE_listener name and RATE_LIMIT parameters.

Example 1

```
CONNECTION_RATE_LISTENER=10

LISTENER=

(ADDRESS_LIST=

(ADDRESS=(PROTOCOL=tcp) (HOST=) (PORT=1521) (RATE_LIMIT=yes))

(ADDRESS=(PROTOCOL=tcp) (HOST=) (PORT=1522) (RATE_LIMIT=yes))

(ADDRESS=(PROTOCOL=tcp) (HOST=) (PORT=1523)))
```

In the preceding example, the global rate of new connections is enforced separately for each endpoint. Connections through port 1521 are limited at 10 every second, and the connections through port 1522 are also separately limited at 10 every second. Connections through port 1523 are not limited.

Example 2

```
LISTENER= (ADDRESS_LIST=
    (ADDRESS=(PROTOCOL=tcp) (HOST=) (PORT=1521) (RATE_LIMIT=5))
    (ADDRESS=(PROTOCOL=tcp) (HOST=) (PORT=1522) (RATE_LIMIT=10))
    (ADDRESS=(PROTOCOL=tcp) (HOST=) (PORT=1523))
)
```

In the preceding example, the connection rates are enforced at the endpoint level. A maximum of 5 connections are processed through port 1521 every second. The limit for connections through port 1522 is 10 every second. Connections through port 1523 are not limited.





The global CONNECTON_RATE_listener_name parameter is not specified in the preceding configuration. If it is specified, then the limits on ports 1521 and 1522 are ignored, and the global value is used instead.

7.4 Control Parameters

This section describes the following parameters that control the behavior of the listener:

- ADMIN_RESTRICTIONS_listener_name
 - The listener.ora control parameter ADMIN_RESTRICTIONS_listener_name restricts runtime administration of the listener.
- ALLOW_MULTIPLE_REDIRECTS_listener_name
 The listener.ora control parameter ALLOW_MULTIPLE_REDIRECTS_listener_name enables multiple redirects of the client.
- ENABLE EXADIRECT listener name
- CRS_NOTIFICATION_listener_name
 CRS_NOTIFICATION_listener_name control parameter of the listener.ora file sets
 notification to allow or disallow Cluster Ready Services (CRS) to manage the listener in an Oracle Real Application Clusters environment.
- DEDICATED_THROUGH_BROKER_LISTENER
 DEDICATED_THROUGH_BROKER_LISTENER networking parameter of the listener.ora file
 enables the server to spawn a thread or process when a connection to the database is requested through the listener.
- DEFAULT_SERVICE_listener_name

 DEFAULT_SERVICE_listener_name control parameter of the listener.ora file enables

 users to connect to the database without having to specify a service name from the client side.
- INBOUND_CONNECT_TIMEOUT_listener_name
- LOCAL REGISTRATION ADDRESS listener name
- MAX_ALL_CONNECTIONS_listener_name
- MAX_REG_CONNECTIONS_listener_name
 Use the MAX_REG_CONNECTIONS_listener_name parameter to specify the maximum number of concurrent registration connection sessions.
- REGISTRATION_EXCLUDED_NODES_listener_name
- REGISTRATION_INVITED_NODES_listener_name
- REMOTE_REGISTRATION_ADDRESS_listener_name
- SAVE CONFIG ON STOP listener name
- SERVICE RATE listener name
 - The SERVICE_RATE_listener_name control parameter specifies incoming connection rate that is allowed per service for an instance.



SSL CIPHER SUITES

Use the SSL_CIPHER_SUITES parameter to control the combination of authentication, encryption, and data integrity algorithms used by Transport Layer Security (TLS).

SSL CLIENT AUTHENTICATION

Use the SSL_CLIENT_AUTHENTICATION parameter to specify whether a client is authenticated using Transport Layer Security (TLS).

SSL DISABLE WEAK EC CURVES

Use the SSL_DISABLE_WEAK_EC_CURVES parameter to disable the use of weak Elliptic Curve Cryptography (ECC) curves.

SSL VERSION

Use the SSL_VERSION parameter to define valid Transport Layer Security (TLS) versions to be used for connections.

- SUBSCRIBE_FOR_NODE_DOWN_EVENT_listener_name
- USE_SID_AS_SERVICE_listener_name
- VALID_NODE_CHECKING_REGISTRATION_listener_name
 The listener.ora control parameter VALID_NODE_CHECKING_REGISTRATION_listener_name
 determines if valid node checking registration is performed, or if the subnet is allowed.
- WALLET_LOCATION
 Use the WALLET LOCATION parameter to specify the location of Oracle wallets.

7.4.1 ADMIN_RESTRICTIONS_listener_name

The listener.ora control parameter ADMIN_RESTRICTIONS_listener_name restricts runtime administration of the listener.

Purpose

To restrict runtime administration of the listener.

Usage Notes

Setting ADMIN_RESTRICTIONS_listener_name=on disables the runtime modification of parameters in listener.ora. That is, the listener refuses to accept SET commands that alter its parameters. To change any of the parameters in listener.ora, including ADMIN_RESTRICTIONS_listener_name itself, modify the listener.ora file manually and reload its parameters using the RELOAD command for the new changes to take effect without explicitly stopping and restarting the listener.

Default

off

Example

ADMIN RESTRICTIONS listener=on

Related Topics

SET

The Listener Control utility command SET alters the parameter values for the listener.

RELOAD

The Listener Control utility command RELOAD reloads the listener.ora file, so that you can add or change statically configured services without stopping the listener.

7.4.2 ALLOW_MULTIPLE_REDIRECTS_listener_name

The *listener.ora* control parameter ALLOW_MULTIPLE_REDIRECTS_listener_name enables multiple redirects of the client.

Purpose

To support multiple redirects of the client.

Usage Notes

This parameter should only be set on the SCAN listener on the Oracle Public Cloud. When set to on, multiple redirects of the client are allowed.

Do not set this parameter for a node listener if that is used as a SCAN listener.

Default

off

Values

on | off

Example

ALLOW_MULTIPLE_REDIRECTS_listener=on

7.4.3 ENABLE_EXADIRECT_listener_name

Purpose

To enable Exadirect protocol.

Usage Notes

The parameter enables Exadirect support.

Default

Off

Values

on | off

Example 7-3 Example

ENABLE EXADIRECT listener=on



7.4.4 CRS_NOTIFICATION_listener_name

CRS_NOTIFICATION_listener_name control parameter of the listener.ora file sets notification to allow or disallow Cluster Ready Services (CRS) to manage the listener in an Oracle Real Application Clusters environment.

Purpose

To set notification.

Usage Notes

By default, the Oracle Net listener notifies Cluster Ready Services (CRS) when it is started or stopped. These notifications allow CRS to manage the listener in an Oracle Real Application Clusters environment. This behavior can be prevented by setting the CRS NOTIFICATION listener name parameter to off.

Default

on

Values

on | off

7.4.5 DEDICATED THROUGH BROKER LISTENER

DEDICATED_THROUGH_BROKER_LISTENER networking parameter of the listener.ora file enables the server to spawn a thread or process when a connection to the database is requested through the listener.

Purpose

To enable the server to spawn a thread or process when a connection to the database is requested through the listener.

Default

off

Values

on | off

Example 7-4 Example

(Optional) Enter an example to illustrate your reference here.

7.4.6 DEFAULT_SERVICE_listener_name

DEFAULT_SERVICE_listener_name control parameter of the listener.ora file enables users to connect to the database without having to specify a service name from the client side.

Purpose

To enable users to connect to the database without having to specify a service name from the client side.

Usage Notes

When a client tries to connect to the database, the connection request passes through the listener. The listener may be servicing several different databases. If a service name is configured in this parameter, then users may not necessarily need to specify a service name in the connect syntax. If a user specifies a service name, then the listener connects the user to that specific database, otherwise the listener connects to the service name specified by the DEFAULT_SERVICE_listener_name parameter. For container databases, the client must explicitly specify the service name.

Default

There is no default value for the <code>DEFAULT_SERVICE_listener_name</code> parameter. If this parameter is not configured and a user does not specify a fully-qualified service name in the connect syntax, then the connection attempt fails. This parameter only accepts one value.

Example 7-5 Example

DEFAULT SERVICE listener=sales.us.example.com

7.4.7 INBOUND CONNECT TIMEOUT listener_name

Purpose

To specify the time, in seconds, for the client to complete its connect request to the listener after the network connection had been established.

Usage Notes

If the listener does not receive the client request in the time specified, then it terminates the connection. In addition, the listener logs the IP address of the client and an ORA-12525: TNS: listener has not received client's request in time allowed error message to the listener.log file.

To protect both the listener and the database server, Oracle recommends setting this parameter in combination with the SQLNET.INBOUND_CONNECT_TIMEOUT parameter in the sqlnet.ora file. When specifying values for these parameters, consider the following recommendations:

- Set both parameters to an initial low value.
- Set the value of the INBOUND_CONNECT_TIMEOUT_listener_name parameter to a lower value than the SQLNET.INBOUND CONNECT TIMEOUT parameter.

For example, you can set the <code>INBOUND_CONNECT_TIMEOUT_listener_name</code> parameter to 2 seconds and the <code>INBOUND_CONNECT_TIMEOUT</code> parameter to 3 seconds. If clients are unable to complete connections within the specified time due to system or network delays that are normal for the particular environment, then increment the time as needed.

Default

60 seconds

Example

INBOUND CONNECT TIMEOUT listener=2



7.4.8 LOCAL_REGISTRATION_ADDRESS_listener_name

Purpose

To secure registration requests through dedicated secure registration endpoints for local listeners. Service ACLs are accepted by listener only if <code>LOCAL_REGISTRATION_ADDRESS_Isnr</code> alias is configured. The parameter specifies the group that is allowed to send ACLs.

Usage Notes

The local registration endpoint accepts local registration connections from the specified group. All local registration requests coming on normal listening endpoints are redirected to the local registration endpoint. If the registrar is not a part of the group, then it cannot connect to the endpoint.

Default

OFF

Values

ON, OFF, or IPC endpoint address with group

When set to ON, listener defaults the group to oinstall on UNIX and ORA_INSTALL on Windows.

Example 7-6 Example

Related Topics

- Firewall
- DBMS_SFW_ACL_ADMIN

7.4.9 MAX ALL CONNECTIONS_listener_name

Purpose

To specify the maximum number of concurrent registration and client connection sessions that can be supported by Oracle Net Listener.

Usage Notes

This number includes registration connections from databases, and ongoing client connection establishment requests. After a connection is established, the clients do not maintain a connection to the listener. This limit only applies to client connections that are in the initial connection establishment phase from a listener perspective.

Default

Operating system-specific



Example

MAX ALL CONNECTIONS listener=4096

7.4.10 MAX REG CONNECTIONS_listener_name

Use the ${\tt MAX_REG_CONNECTIONS_listener_name}$ parameter to specify the maximum number of concurrent registration connection sessions.

Purpose

To specify the maximum number of concurrent registration connection sessions that can be supported by Oracle Net Listener.

Default

512

Example

MAX REG CONNECTIONS listener=2048

7.4.11 REGISTRATION_EXCLUDED_NODES_listener_name

Purpose

To specify the list of nodes that cannot register with the listener.

Usage Notes

The list can include host names or CIDR notation for IPv4 and IPv6 addresses. The wildcard format (*) is supported for IPv4 addresses. The presence of a host name in the list results in the inclusion of all IP addresses mapped to the host name. The host name should be consistent with the public network interface.

If the <code>REGISTRATION_INVITED_NODES_listener_name</code> parameter and the <code>REGISTRATION_EXCLUDED_NODES_listener_name</code> parameter are set, then the <code>REGISTRATION_EXCLUDED_NODES_listener_name</code> parameter is ignored.

Values

Valid nodes and subnet IP addresses or names.

Example

```
REGISTRATION_EXCLUDED_NODES_listener = (10.1.26.*, 10.16.40.0/24, \
2001:DB8:3eff:fe38, node2)
```

7.4.12 REGISTRATION INVITED NODES listener name

Purpose

To specify the list of node that can register with the listener.

Usage Notes

- The list can include host names or CIDR notation for IPv4 and IPv6 addresses. The wildcard format (*) is supported for IPv4 addresses. The presence of a host name in the list results in the inclusion of all IP addresses mapped to the host name. The host name should be consistent with the public network interface.
- If the REGISTRATION_INVITED_NODES_listener_name parameter and the REGISTRATION_EXCLUDED_NODES_listener_name parameter are set, then the REGISTRATION_EXCLUDED_NODES_listener_name parameter is ignored.
- Starting with Oracle Grid Infrastructure 12c, for a SCAN listener, if the VALID_NODE_CHECKING_REGISTRATION_listener_name and REGISTRATION_INVITED_NODES_listener_name parameters are set in the listener.ora file, then the listener agent overwrites these parameters.

Values

Valid nodes and subnet IP addresses or names.

Example

```
REGISTRATION_INVITED_NODES_listener = (10.1.35.*, 10.1.34.0/24, \
2001:DB8:fe38:7303, node1)
```

See Also

Oracle Real Application Clusters Administration and Deployment Guide for information about valid node checking for registration

7.4.13 REMOTE_REGISTRATION_ADDRESS_listener_name

Purpose

To secure registration requests through dedicated secure registration endpoints for SCAN listeners.

Usage Notes

The registration endpoint is on a private network within the cluster. All remote registration requests coming in on normal listening endpoints are redirected to the registration endpoint. Any system which is not a part of the cluster cannot connect to the endpoint. This feature is not supported when ADMIN_RESTRICTIONS_listener_name is set to ON as the Cluster Ready Services agent configures the remote registration address dynamically at run time.

Default

This parameter is configured internally in listeners managed by Oracle Clusterware to restrict registrations to the private network. The value of this parameter should not be modified or specified explicitly. The only supported explicit setting is for turning this feature off by setting the value to OFF.



Values

off

Example

REMOTE_REGISTRATION_ADDRESS_listener=off

7.4.14 SAVE CONFIG ON STOP listener_name

Purpose

To specify whether runtime configuration changes are saved to the listener.ora file.

Usage Notes

When you set the parameter to true, any parameters that were modified while the listener was running using the Listener Control utility SET command are saved to the listener.ora file when the STOP command is issued. When you set the parameter to false, the Listener Control utility does not save the runtime configuration changes to the listener.ora file.

Default

false

Values

true | false

Example

SAVE_CONFIG_ON_STOP_listener=true

7.4.15 SERVICE_RATE_listener_name

The SERVICE_RATE_listener_name control parameter specifies incoming connection rate that is allowed per service for an instance.

Purpose

To specify incoming connection rate that is allowed per service for an instance.

Usage Notes

Any user-specified value greater than 0 sets the maximum limit on the number of new connections per service-instance handled by the proxy listener every second. Listener rejects connections after it reaches the maximum limit. Client side connection failure is reported with the "TNS:listener: rate limit reached" error.

Default

0

Example 7-7 Example

SERVICE RATE=10



7.4.16 SSL_CIPHER_SUITES

Use the SSL_CIPHER_SUITES parameter to control the combination of authentication, encryption, and data integrity algorithms used by Transport Layer Security (TLS).

Purpose

To control the combination of authentication, encryption, and data integrity algorithms used by Transport Layer Security (TLS). By default, the strongest protocol and cipher are negotiated between the database client and server. Setting this parameter will override the default behavior. You must use this parameter only if you have internal security controls that dictate the usage of certain protocol versions.

Usage Notes

Enclose the SSL_CIPHER_SUITES parameter value in parentheses. Otherwise, the cipher suite setting does not parse correctly.

Default

None

Values

Approved ciphers compatible with TLS 1.2:

- TLS ECDHE RSA WITH AES 256 GCM SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS ECDHE ECDSA WITH AES 256 GCM SHA384
- TLS ECDHE ECDSA WITH AES 128 GCM SHA256
- TLS DHE RSA WITH AES 256 GCM SHA384
- TLS DHE RSA WITH AES 128_GCM_SHA256

Deprecated ciphers compatible with TLS 1.2:

- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS ECDHE RSA WITH AES 128 CBC SHA256
- TLS ECDHE ECDSA WITH AES 256 CBC SHA384
- TLS ECDHE ECDSA WITH AES 128 CBC SHA256
- TLS RSA WITH AES 256 GCM SHA384
- TLS RSA WITH AES 256 CBC SHA256
- TLS_RSA_WITH_AES_128_GCM_SHA256
- TLS RSA WITH AES 128 CBC SHA256
- TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384
- TLS ECDH RSA WITH AES 128 GCM SHA256
- TLS ECDH RSA WITH AES 256 CBC SHA384
- TLS ECDH RSA WITH AES 128 CBC SHA256
- TLS ECDH ECDSA WITH AES 256 GCM SHA384



- TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384
- TLS ECDH ECDSA WITH AES 128 CBC SHA256
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
- TLS DHE RSA WITH AES 128 CBC SHA256
- TLS_DH_anon_WITH_AES_256_GCM_SHA384
- TLS_DH_anon_WITH_AES_128_GCM_SHA256

Deprecated ciphers compatible with TLS 1.0, TLS 1.1, and TLS 1.2:

- TLS ECDHE RSA WITH AES 256 CBC SHA
- TLS ECDHE RSA WITH AES 128 CBC SHA
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
- TLS ECDHE ECDSA WITH AES 128 CBC SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS RSA WITH AES 128 CBC SHA
- TLS ECDH RSA WITH AES 256 CBC SHA
- TLS_ECDH_RSA_WITH_AES_128_CBC_SHA
- TLS ECDH ECDSA WITH AES 256 CBC SHA
- TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA
- TLS DHE RSA WITH AES 256 CBC SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_RSA_WITH_RC4_128_SHA
- TLS ECDHE ECDSA WITH RC4 128 SHA
- TLS ECDH RSA WITH RC4 128 SHA
- TLS ECDH ECDSA WITH RC4 128 SHA
- SSL RSA WITH RC4 128 SHA
- SSL_RSA_WITH_RC4_128_MD5
- TLS ECDHE ECDSA WITH NULL SHA
- TLS_ECDHE_RSA_WITH_NULL_SHA
- TLS_ECDH_ECDSA_WITH_NULL_SHA
- TLS_ECDH_RSA_WITH_NULL_SHA
- SSL_RSA_WITH_NULL_SHA
- SSL RSA WITH NULL MD5
- SSL_DH_anon_WITH_RC4_128_MD5

Deprecated ciphers compatible with TLS 1.0 and TLS 1.1:

- TLS ECDHE RSA WITH 3DES EDE CBC SHA
- TLS ECDHE ECDSA WITH 3DES EDE CBC SHA



- TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA
- TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA
- SSL RSA WITH 3DES EDE CBC SHA
- SSL_DH_anon_WITH_3DES_EDE_CBC_SHA



The $\mathtt{DH_anon}$ cipher suites do not provide authentication of the communicating parties, and can be vulnerable to man-in-the-middle attacks. Oracle recommends that you do not use these cipher suites to protect sensitive data.

Examples

```
SSL_CIPHER_SUITES=(TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384)

SSL_CIPHER_SUITES=(TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
```

Related Topics

- Set the TLS Cipher Suites on the Server
- Set the TLS Cipher Suites on the Client

7.4.17 SSL_CLIENT_AUTHENTICATION

Use the SSL_CLIENT_AUTHENTICATION parameter to specify whether a client is authenticated using Transport Layer Security (TLS).

Purpose

To specify whether a client is authenticated using Transport Layer Security (TLS).

Usage Notes

The database server authenticates the client. Therefore, this value should be set to false. If this parameter is set to true, then the listener attempts to authenticate the client, which can result in a failure.

Default

true

Values

true | false

Example

 ${\tt SSL_CLIENT_AUTHENTICATION=false}$





Oracle Database Security Guide

7.4.18 SSL_DISABLE_WEAK_EC_CURVES

Use the SSL_DISABLE_WEAK_EC_CURVES parameter to disable the use of weak Elliptic Curve Cryptography (ECC) curves.

Purpose

To disable the use of weak ECC curves with key length less than 256 bits. You can set this parameter in the database server (sqlnet.ora), client (sqlnet.ora or tnsnames.ora connect string), or the listener (listener.ora).

Usage Notes

By default, this parameter is set to FALSE to enable the use of all ECC curves. If you want to enable the use of only Oracle approved curves with ECC curve key size of 256 bits or higher, then set this parameter to TRUE.

When set to TRUE, you can use only the following ECC curves:

- secp256r1
- secp384r1
- secp521r1

Values

- TRUE | ON | YES | 1: To enable only the Oracle approved ECC curves with minimum ECC curve key length of 256 bits
- FALSE | OFF | NO | 0: To enable all ECC curves

Default

FALSE

Examples

In the tnsnames.ora file:

```
net_service_name=
  (DESCRIPTION=
       (ADDRESS=(PROTOCOL=tcps) (HOST=sales-svr) (PORT=1521))
       (SECURITY=(SSL_DISABLE_WEAK_EC_CURVES=TRUE))
)
```

• In the sqlnet.ora file or the listener.ora file:

```
SSL DISABLE WEAK EC CURVES=TRUE
```



7.4.19 SSL_VERSION

Use the SSL_VERSION parameter to define valid Transport Layer Security (TLS) versions to be used for connections.

Purpose

To define the version of TLS that must run on the systems with which the database server communicates. By default, the database server and client negotiate the strongest security protocol. Oracle does not recommend modifying this parameter, unless your security requirements mandate the usage of certain protocol versions.

Usage Notes

Clients, listeners, and database servers must use compatible versions. Modify this parameter only when necessary to enforce the use of the more secure TLS protocol and not allow clients that only work with the older TLS protocols. If you need to specify TLS 1.0 or TLS 1.1, then also include TLS 1.2 to allow more secure connections. The current default uses TLS 1.2, which is the version required for multiple security compliance requirements.

If you set SSL_VERSION to undetermined, then the most secure TLS protocol version is used. You can use the SSL_VERSION=undetermined setting in the connect string for a specific connection to override the SSL_VERSION value configured in the sqlnet.ora file.

If you do not set SSL_VERSION to any value, then all the supported TLS protocol versions are tried starting with the most secure version. This is typically the most common configuration, ensuring that the strongest protocol is chosen during TLS negotiation.

Default

undetermined

Values

```
undetermined | 1.0 | 1.1 | 1.2
```

The version numbers correspond to the TLS versions, such as TLSv1.0, TLSv1.1, and TLSv1.2.



The sqlnet.ora parameter ADD_SSLV3_TO_DEFAULT has no impact on this parameter.

Syntax and Examples

To specify a single TLS version:

```
SSL VERSION=TLS protocol version
```

For example:

SSL VERSION=1.2



To specify multiple TLS versions, use the or operator as follows:

```
SSL VERSION=TLS protocol version1 or TLS protocol version2
```

For example:

```
SSL_VERSION=1.1 or 1.2

SSL_VERSION=1.0 or 1.1 or 1.2
```

Related Topics

- · Set the Required TLS Version on the Server
- Set the Required TLS Version on the Client

7.4.20 SUBSCRIBE_FOR_NODE_DOWN_EVENT_listener_name

Purpose

To subscribe to Oracle Notification Service (ONS) notifications for downed events.

Usage Notes

By default, the listener subscribes to the ONS node down event on startup, if ONS is available. This subscription enables the listener to remove the affected service when it receives node down event notification from ONS. The listener uses asynchronous subscription for the event notification. Alter this behavior by setting

```
{\tt SUBSCRIBE\_FOR\_NODe\_DOWN\_EVENT\_listener\_name} = {\tt off} \ \textbf{in} \ {\tt listener\_ora}.
```

Default

on

Values

on | off

7.4.21 USE_SID_AS_SERVICE_listener_name

Purpose

To enable the system identifier (SID) in the connect descriptor to be interpreted as a service name when a user attempts a database connection.

Usage Notes

Database clients with earlier releases of Oracle Database that have hard-coded connect descriptors can use this parameter to connect to a container or pluggable database.

For an Oracle container database, the client must specify a service name in order to connect to it. Setting this parameter to on instructs the listener to use the SID in the connect descriptor as a service name and connect the client to the specified database.



Default

off

Example

USE SID AS SERVICE listener=on

7.4.22 VALID_NODE_CHECKING_REGISTRATION_listener_name

The *listener.ora* control parameter VALID_NODE_CHECKING_REGISTRATION_*listener_name* determines if valid node checking registration is performed, or if the subnet is allowed.

Purpose

To determine whether valid node checking registration is performed, or the subnet is allowed.

Usage Notes

- When set to on, valid node checking registration is performed at the listener for any incoming registration request, and only local IP addresses are allowed.
- Starting with Oracle Grid Infrastructure 12c, for a SCAN listener, if the VALID_NODE_CHECKING_REGISTRATION_listener_name and REGISTRATION_INVITED_NODES_listener_name parameters are set in the listener.ora file, then the listener agent overwrites these parameters.

Default

on

Values

- off | 0 to specify valid node checking registration is off, and no checking is performed.
- on | 1 | local to specify valid node checking registration is on, and all local IP addresses can register. If a list of invited nodes is set, then all IP addresses, host names, or subnets in the list as well as local IP addresses are allowed.
- subnet | 2 to specify valid node checking registration is on, and all machines in the local subnets are allowed to register. If a list of invited nodes is set, then all nodes in the local subnets as well as all IP addresses, host names and subnets in the list are allowed.

Example

VALID NODE CHECKING REGISTRATION listener=on



Oracle Real Application Clusters Administration and Deployment Guide for information about valid node checking for registration



7.4.23 WALLET_LOCATION

Use the Wallet location parameter to specify the location of Oracle wallets.

Purpose

To specify the directory path where an Oracle wallet is stored. Wallets securely contain certificates, secrets, private keys, and trust points used by Oracle Database.

Usage Notes

Where to set this parameter:

You can set WALLET_LOCATION in the listener.ora file to specify a common wallet location for all listeners. You can also set WALLET_LOCATION as part of address string of the listener in listener.ora to specify a different wallet location for each listener. For example:

Storage of wallet files:

The password-protected wallet is stored in an <code>ewallet.p12</code> file. The auto-login and local auto-login wallets are stored in a <code>cwallet.sso</code> file.

For example, if an Oracle wallet is stored in the Microsoft Windows registry and the wallet's key (KEY) is SALESAPP, then the storage location of the password-protected wallet is HKEY_CURRENT_USER\SOFTWARE\ORACLE\WALLETS\SALESAPP\EWALLET.P12. The storage location of the auto-login and local auto-login wallets is HKEY_CURRENT_USER\SOFTWARE\ORACLE\WALLETS\SALESAPP\CWALLET.SSO.

Additional Parameters

Use Source to specify the type of storage and storage location for wallets, as follows:

- METHOD: Type of storage
- METHOD DATA: Storage location:
 - DIRECTORY: Location of wallet on the file system
 - KEY: Wallet type and location in the Microsoft Windows registry

Syntax and Examples

The syntax depends on the wallet as follows:

Wallet on the file system:

```
WALLET_LOCATION=
  (SOURCE=
    (METHOD=file)
    (METHOD_DATA=
          (DIRECTORY=directory)))
```

For example:

```
WALLET_LOCATION=
(SOURCE=
```

```
(METHOD=file)
(METHOD_DATA=
    (DIRECTORY=/etc/oracle/wallets/databases)))
```

Microsoft certificate store:

```
WALLET_LOCATION=
  (SOURCE=
          (METHOD=mcs))
```

The key-value pair for MCS omits the METHOD_DATA parameter because MCS does not use wallets. Instead, Oracle PKI (public key infrastructure) applications obtain certificates, trust points and private keys directly from a user's profile.

Wallet in the Microsoft Windows registry:

(KEY=SALESAPP)))

Default

None

7.5 ADR Diagnostic Parameters for Oracle Net Listener

The diagnostic data for the critical errors is quickly captured and stored in the ADR for Oracle Net listener.

Since Oracle Database 11g, Oracle Database includes an advanced fault diagnosability infrastructure for preventing, detecting, diagnosing, and resolving problems. The problems are critical errors such as those caused by database code bugs, metadata corruption, and customer data corruption.

When a critical error occurs, it is assigned an incident number, and diagnostic data for the error, such as traces and dumps, are immediately captured and tagged with the incident number. The data is then stored in the Automatic Diagnostic Repository (ADR), a file-based repository outside the database.

This section includes the parameters used when ADR is enabled. ADR is enabled by default. Non-ADR parameters listed in the listener.ora file are ignored when ADR is enabled.

The following listener.ora parameters are used when ADR is enabled (when DIAG ADR ENABLED is set to on):

ADR BASE listener name

The ADR_BASE_listener_name parameter is a diagnostic parameter specifies the base directory that stores tracing and logging incidents when ADR is enabled.

DIAG ADR ENABLED listener name

The DIAG_ADR_ENABLED_listener_name is a diagnostic parameter of the listener.ora file. It indicates whether ADR is enabled.

· LOG FILE NUM listener name

The LOG_FILE_NUM_listener_name is a diagnostic parameter of the listener.ora file that specifies the number of log file segments.

· LOG FILE SIZE listener name

The LOG_FILE_SIZE_listener_name diagnostic parameter of the listener.ora file specifies the size of each log file segment.

LOGGING listener name

The LOGGING_listener_name diagnostic parameter of the listener.ora file turns logging on or off.

TRACE LEVEL listener name

The TRACE_LEVEL_listener_name diagnostic parameter of the listener.ora file turns listener tracing on, at a specific level, or turns it off.

TRACE TIMESTAMP listener name

The TRACE_TIMESTAMP_listener_name diagnostic parameter of the listener.ora file adds a time stamp to every trace event in the trace file for the listener.

7.5.1 ADR_BASE_listener_name

The ADR_BASE_listener_name parameter is a diagnostic parameter specifies the base directory that stores tracing and logging incidents when ADR is enabled.

Purpose

To specify the base directory that stores tracing and logging incidents when ADR is enabled.

Default

The default is ORACLE BASE, or ORACLE HOME/log if ORACLE BASE is not defined.

Values

Any valid directory path to a directory with write permission.

Example

ADR BASE listener=/oracle/network/trace

7.5.2 DIAG ADR ENABLED listener name

The <code>DIAG_ADR_ENABLED_listener_name</code> is a diagnostic parameter of the <code>listener.ora</code> file. It indicates whether ADR is enabled.

Purpose

To indicate whether ADR tracing is enabled.



Usage Notes

When the <code>DIAG_ADR_ENABLED_listener_name</code> parameter is set to <code>on</code>, then ADR file tracing is used. When the <code>DIAG_ADR_ENABLED_listener_name</code> parameter is set to <code>off</code>, then non-ADR file tracing is used.

Default

on

Values

onloff

Example 7-8 Example

DIAG_ADR_ENABLED_listener=on

7.5.3 LOG_FILE_NUM_listener_name

The LOG_FILE_NUM_listener_name is a diagnostic parameter of the listener.ora file that specifies the number of log file segments.

Purpose

To specify the number of log file segments. At any point of time there can be only n log file segments where n is LOG_FILE_NUM_listener_name. If the log grows beyond this number, then the older segments are deleted.

Default

No default. If you don't specify a value, or set the value to zero, then the number of segments grows indefinitely.

Values

Any integer value.

Example 7-9

LOG_FILE_NUM_listener=3

7.5.4 LOG_FILE_SIZE_listener_name

The LOG_FILE_SIZE_listener_name diagnostic parameter of the listener.ora file specifies the size of each log file segment.

Purpose

To specify the size of each log file segment. The size is in MB.

Default

300 MB

Values

Any integer value.



Example 7-10 Example

LOG FILE SIZE listener=10

7.5.5 LOGGING_listener_name

The LOGGING_listener_name diagnostic parameter of the listener.ora file turns logging on or off.

Purpose

To turn logging on or off.

Usage Notes

This parameter is also applicable when non-ADR tracing is used.

Default

on

Values

on off

Example

LOGGING_listener=on

7.5.6 TRACE_LEVEL_listener_name

The TRACE_LEVEL_listener_name diagnostic parameter of the listener.ora file turns listener tracing on, at a specific level, or turns it off.

Purpose

To turn listener tracing on, at a specific level, or to turn it off.

Usage Notes

This parameter is also applicable when non-ADR tracing is used.

Default

off | 0

Values

- off or 0 for no trace output
- user or 4 for user trace information
- admin or 10 for administration trace information
- support or 16 for Oracle Support Services trace information



Example

TRACE LEVEL listener=admin

7.5.7 TRACE_TIMESTAMP_listener_name

The TRACE_TIMESTAMP_listener_name diagnostic parameter of the listener.ora file adds a time stamp to every trace event in the trace file for the listener.

Purpose

To add a time stamp in the form of dd-mmm-yyyy hh:mi:ss:mil to every trace event in the trace file for the listener.

Usage Notes

This parameter is used with the TRACE_LEVEL_listener_name parameter. This parameter is also applicable when non-ADR tracing is used.

Default

on

Values

- on | true
- off | false

Example

TRACE TIMESTAMP listener=true

7.6 Non-ADR Diagnostic Parameters for Oracle Net Listener

This section lists the parameters used when ADR is disabled. The default value of DIAG_ADR_ENABLED_listener_name is on. Therefore, the DIAG_ADR_ENABLED_listener_name parameter must explicitly be set to off to use non-ADR tracing.

- LOG_DIRECTORY_listener_name
- LOG_FILE_listener_name
- TRACE_DIRECTORY_listener_name
- TRACE_FILE_listener_name
- TRACE_FILEAGE_listener_name
- TRACE_FILELEN_listener_name
- TRACE_FILENO_listener_name



7.6.1 LOG_DIRECTORY_listener_name

Purpose

To specify the destination directory of the listener log file.

Usage Notes

Use this parameter when ADR is not enabled.

Default

ORACLE HOME/network/log

Example

LOG DIRECTORY listener=/oracle/network/admin/log

7.6.2 LOG_FILE_listener_name

Purpose

To specify the name of the log file for the listener.

Usage Notes

Use this parameter when ADR is not enabled.

Default

listener.log

Example

LOG_FILE_listener=list.log

7.6.3 TRACE_DIRECTORY_listener_name

Purpose

To specify the destination directory of the listener trace file.

Usage Notes

Use this parameter when ADR is not enabled.

Default

ORACLE HOME/network/trace

Example

TRACE_DIRECTORY_listener=/oracle/network/admin/trace



7.6.4 TRACE_FILE_listener_name

Purpose

To specify the name of the trace file for the listener.

Usage Notes

Use this parameter when ADR is not enabled.

Default

listener.trc

Example

TRACE_FILE_listener=list.trc

7.6.5 TRACE_FILEAGE_listener_name

Purpose

To specify the maximum age of listener trace files in minutes.

Usage Notes

When the age limit is reached, the trace information is written to the next file. The number of files is specified with the TRACE_FILENO_listener_name parameter. Use this parameter when ADR is not enabled.

Default

Unlimited

This is the same as setting the parameter to 0.

Example 7-11 Example

TRACE FILEAGE listener=60

7.6.6 TRACE_FILELEN_listener_name

Purpose

To specify the size of the listener trace files in kilobytes (KB).

Usage Notes

When the size is met, the trace information is written to the next file. The number of files is specified using the TRACE_FILENO_listener_name parameter. Use this parameter when ADR is not enabled.

Default

Unlimited



Example

TRACE FILELEN listener=100

7.6.7 TRACE_FILENO_listener_name

Purpose

To specify the number of trace files for listener tracing.

Usage Notes

When this parameter is set along with the TRACE_FILELEN_listener_name parameter, trace files are used in a cyclical fashion. The first file is filled first, then the second file, and so on. When the last file has been filled, the first file is re-used, and so on.

The trace file names are distinguished from one another by their sequence number. For example, if the default trace file of listener.trc is used, and this parameter is set to 3, then the trace files would be named listener1.trc, listener2.trc and listener3.trc.

In addition, trace events in the trace files are preceded by the sequence number of the file. Use this parameter when ADR is not enabled.

Default

1

Example

TRACE_FILENO_listener=3

7.7 Class of Secure Transports Parameters

The class of secure transports (COST) parameters specify a list of transports that are considered secure for administration and registration of a particular listener.

The COST parameters identify which transports are considered secure for that installation and whether the administration of a listener requires secure transports. Configuring these parameters is optional.

- SECURE_REGISTER_listener_name
- Using COST Parameters in Combination
- DYNAMIC_REGISTRATION_listener_name
 DYNAMIC_REGISTRATION_listener_name is a class of secure transports (COST) parameter
 and it enables or disables dynamic registration of a listener.
- SECURE_PROTOCOL_listener_name
- SECURE_CONTROL_listener_name

See Also:

Oracle Database Net Services Administrator's Guide for additional information about COST parameters and listener security



7.7.1 SECURE_REGISTER_listener_name

Purpose

To specify the transports on which registration requests are to be accepted.

Usage Notes

If the SECURE_REGISTER_listener_name parameter is configured with a list of transport names, then only the connections arriving on the specified transports are able to register the service with the listener. Connections arriving by other transport protocols are refused. The following is an example:

```
SECURE REGISTER listener1 = (TCPS, IPC)
```

In the preceding example, registration requests are accepted only on TCPS and IPC transports.

If no values are entered for this parameter, then the listener accepts registration requests from any transport.

Syntax

```
SECURE_REGISTER_listener_name =
[(]transport1[,transport2, ....,transportn)]
```

In the preceding example, transport1, transport2, and transport n are valid, installed transport protocol names.

If this parameter and SECURE_CONTROL_listener_name are configured, then they override the SECURE_PROTOCOL_listener_name parameter.

Example

7.7.2 Using COST Parameters in Combination

COST parameters can also be used in combination to further control which transports accept service registration and control commands.

In Example 7-12, control commands are accepted only on the IPC channel and the TCPS transport, and service registrations are accepted only on an IPC channel.

Example 7-12 Combining COST Parameters



```
SECURE_CONTROL_listener1=(tcps,ipc)
SECURE REGISTER listener1=ipc
```

In Example 7-13, control commands are accepted only on the TCPS transport, and service registrations are accepted only on the IPC channel.

Example 7-13 Combining COST Parameters

```
LISTENER1=
(DESCRIPTION=
    (ADDRESS_LIST=
          (ADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=1521))
          (ADDRESS=(PROTOCOL=ipc) (KEY=extproc))
          (ADDRESS=(PROTOCOL=tcps) (HOST=sales-server) (PORT=1522))))
SECURE_CONTROL_listener1=tcps
SECURE_PROTOCOL_listener1=ipc
```

7.7.3 DYNAMIC_REGISTRATION_listener_name

DYNAMIC_REGISTRATION_listener_name is a class of secure transports (COST) parameter and it enables or disables dynamic registration of a listener.

Purpose

To enable or disable dynamic registration.

Usage Notes

Static registrations are not affected by this parameter.

Default

The default value is on. Unless this parameter is explicitly set to off, all registration connections are accepted.

Values

- on: The listener accepts dynamic registration.
- off: The listener refuses dynamic registration.

Example 7-14 Example

```
DYNAMIC REGISTRATION listener name=on
```

7.7.4 SECURE PROTOCOL listener name

Purpose

To specify the transports on which administration and registration requests are accepted.

Usage Notes

If this parameter is configured with a list of transport names, then the control commands and service registration can happen only if the connection belongs to the list of transports.

If this parameter is not present and neither SECURE_CONTROL_listener_name or SECURE_REGISTER_listener_name are configured, then all supported transports accept control and registration requests.

If the SECURE_CONTROL_listener_name and SECURE_REGISTER_listener_name parameters are configured, then they override the <code>SECURE_PROTOCOL_listener_name</code> parameter.

Syntax

```
SECURE_PROTOCOL_listener_name =
[(]transport1[,transport2, ....,transportn)]
```

In the preceding syntax, transport1, transport2, and transportn are valid, installed transport protocol names.

Example

7.7.5 SECURE CONTROL listener_name

Purpose

To specify the transports on which control commands are to be serviced.

Usage Notes

If the SECURE_CONTROL_listener_name parameter is configured with a list of transport names, then the control commands are serviced only if the connection is one of the listed transports. Connections arriving by other transport protocols are refused. The following is an example:

```
SECURE_CONTROL_listener1 = (TCPS,IPC)
```

In the preceding example, administration requests are accepted only on TCPS and IPC transports.

If no values are entered for this parameter, then the listener accepts any connection on any endpoint.

Syntax

```
SECURE_CONTROL_listener_name =
[(]transport1[,transport2, ....,transportn)]
```

In the preceding syntax, transport1, transport2, and transportn are valid, installed transport protocol names.

Example



Oracle Connection Manager Parameters

This chapter provides a complete listing of the cman.ora file configuration parameters.

- Overview of Oracle Connection Manager Configuration File
 Oracle Connection Manager configuration information is stored in the cman.ora file.
- Oracle Connection Manager Parameters
 This section lists and describes the following cman.ora file parameters:
- Oracle Connection Manager in Traffic Director Mode Parameters
 This section lists and describes the cman.ora file parameters.
- ADR Diagnostic Parameters for Oracle Connection Manager
 The diagnostic data for critical errors is quickly captured and stored in the ADR for Oracle Connection Manager.
- Non-ADR Diagnostic Parameters for Oracle Connection Manager
 This section lists the parameters used when ADR is disabled:

8.1 Overview of Oracle Connection Manager Configuration File

Oracle Connection Manager configuration information is stored in the cman.ora file.

Oracle Connection Manager Configuration File

Oracle Connection Manager configuration information consists of the following elements:

- Protocol address of the Oracle Connection Manager listener
- Access control parameters
- Performance parameters

By default, the cman.ora file is located in the <code>ORACLE_HOME/network/admin</code> directory. You can also store the cman.ora file in the following locations:

- The directory specified by the TNS ADMIN environment variable or registry value.
- On Linux and UNIX operating systems, the global configuration directory. For example, on the Oracle Solaris operating system, this directory is /var/opt/oracle.
- ORACLE BASE HOME/network/admin directory.
- ORACLE HOME/network/admin directory.

Example 8-1 Sample cman.ora File

```
(PARAMETER_LIST=
  (MAX_GATEWAY_PROCESSES=8)
  (MIN_GATEWAY_PRCESSSES=3)
  (DIAG_ADR_ENABLED=ON)
  (ADR_BASE=/oracle/log)))
```

cman.ora File Sections

- Listening address: Preceded by ADDRESS=, this section contains information pertinent to the listener. The ADDRESS parameter is required.
- Rule list: Preceded by RULE_LIST=, this section contains rule information. TheRULE parameter is listed in the rule list section of the file. The RULE parameter is required.
- Rule Group: Preceded by RULE_GROUP=, this section contains rule_list grouped by service names. You can use either the rule group syntax or the rule list syntax.
- Parameter list: Preceded by PARAMETER_LIST=, this section contains all other parameters
 including those listed in "ADR Diagnostic Parameters for Oracle Connection Manager",
 and "Non-ADR Diagnostic Parameters for Oracle Connection Manager".

The following parameters are allowed in the parameter list section of the cman.ora file. The default values are bold. To override the default setting for a parameter, enter the parameter and a nondefault value.

```
ASO AUTHENTICATION FILTER={off | on}
ADMIN GROUP=user group
CONNECTION STATISTICS={no | yes}
EVENT GROUP={init and term | memory ops | conn hdlg | proc mgmt | reg and load
| wake up | timer | cmd proc | relay}
IDLE TIMEOUT=0 or greater
INBOUND CONNECT TIMEOUT=0 or greater. The default value is 60.
LOG DIRECTORY=log directory. The default value is ORACLE HOME/network/log.
LOG LEVEL={off | user | admin | support}
MAX CMCTL SESSIONS= Any positive number. The default value is 4.
MAX CONNECTIONS= A value between 1 and 1024. The default value is 256.
MAX GATEWAY PROCESSES Any number greater than the minimum number of gateway
processes up to 64. The default value is 16.
MIN GATEWAY PROCESSES= Any positive number less than or equal to 64. Must be less than
or equal to the maximum number of gateway processes. The default value is 2.
OUTBOUND CONNECT TIMEOUT=0 or greater
PASSWORD instance name= Value is the encrypted instance password, if one has been set.
The default value is no value.
SESSION TIMEOUT=0 or greater
TRACE DIRECTORY=trace directory. The default value is ORACLE HOME/network/trace.
TRACE FILELEN= Any positive number. The default value is 0 (zero).
TRACE FILENO= Any positive number. The default value is 0 (zero).
TRACE LEVEL={off | user | admin | support}
```



TRACE TIMESTAMP={ off | on}



You cannot add the parameter $PASSWORD_instance_name$ directly to the cman.ora file. The parameter is added using the SAVE_PASSWD command.

```
(PARAMETER LIST=
      (ASO AUTHENTICATION FILTER=ON)
      (ADMIN GROUP=dba)
      (CONNECTION STATISTICS=NO)
      (EVENT GROUP=INIT AND TERM, MEMORY OPS, PROCESS MGMT)
      (IDLE TIMEOUT=30)
      (INBOUND CONNECT TIMEOUT=30)
      (LOG DIRECTORY=/home/user/network/admin/log)
      (LOG LEVEL=SUPPORT)
      (MAX CMCTL SESSIONS=6)
      (MAX CONNECTIONS=512)
      (MAX GATEWAY PROCESSES=10)
      (MIN GATEWAY PROCESSES=4)
      (OUTBOUND CONNECT TIMEOUT=30)
      (SESSION TIMEOUT=60)
      (TRACE DIRECTORY=/home/user/network/admin/trace)
      (TRACE FILELEN=100)
      (TRACE FILENO=2)
      (TRACE LEVEL=SUPPORT)
      (TRACE TIMESTAMP=ON)
      (VALID NODE CHECKING REGISTRATION=ON)
      (REGISTRATION EXCLUDED NODES = 10.1.26.*)
      (REGISTRATION INVITED NODES = 10.1.35.*)
```

8.2 Oracle Connection Manager Parameters

This section lists and describes the following cman.ora file parameters:

ADDRESS

The ADDRESS networking parameter specifies the protocol address of Oracle Connection Manager.

ADMIN GROUP

The ADMIN_GROUP parameter of the cman.ora file specifies the user group that can perform the Oracle Connection Manager (CMAN) administration.

ASO AUTHENTICATION FILTER

It is a networking parameter for Oracle Connection Manager. It instructs Oracle Connection Manager to check the connection requests for Secure Network Services (SNS).

COMPRESSION

The COMPRESSION parameter of the cman.ora file enables or disables data compression.

COMPRESSION LEVELS

The COMPRESSION_LEVELS networking parameter of the cman.ora file specifies the CPU usage and compression ratio.

COMPRESSION THRESHOLD

The COMPRESSION_THRESHOLD parameter of the cman.ora file specifies the minimum data size for which compression is required.

CONNECTION STATISTICS

CONNECTION_STATISTICS networking parameter of the cman.ora file specifies whether the SHOW CONNECTIONS command displays connection statistics.

EVENT GROUP

EVENT_GROUP networking parameter of the cman.ora file specifies which event groups are logged.

EXPIRE_TIME

The EXPIRE_TIME networking parameter of cman.ora file specifies a time interval, in minutes, to send a check to verify that client/gateway connections are active.

IDLE_TIMEOUT

INBOUND CONNECT TIMEOUT

The ${\tt INBOUND_CONNECT_TIMEOUT}$ parameter of the cman.ora file specifies the time limit that the Oracle Connection Manager listener waits for a valid connection before timing out.

LOG FILE NUM

The LOG_FILE_NUM networking parameter of the cman.ora file specifies the number of log file segments.

LOG_FILE_SIZE

 ${\tt LOG_FILE_SIZE}$ networking parameter of the cman.ora file specifies the size of each log file segment.

MAX_ALL_CONNECTIONS

MAX CMCTL SESSIONS

The MAX_CMCTL_SESSIONS parameter of the cman.ora file specifies the maximum number of concurrent local or remote sessions.

MAX CONNECTIONS

The MAX_CONNECTIONS parameter of the cman.ora file specifies the maximum number of connection slots that a gateway process can handle.

- MAX_GATEWAY_PROCESSES
- MAX_REG_CONNECTIONS
- MIN GATEWAY PROCESSES
- OUTBOUND_CONNECT_TIMEOUT
- PASSWORD_instance_name

REGISTRATION EXCLUDED NODES

The Oracle Connection Manager parameter file (cman.ora)

REGISTRATION_EXCLUDED_NODES specifies the list of nodes that cannot register with the listener.

REGISTRATION INVITED NODES

The Oracle Connection Manager parameter file (cman.ora) REGISTRATION_EXCLUDED_NODES parameter specifies the list of node that can register with the listener.



RULE

The RULE parameter of the cman.ora file specifies an access control rule list to filter incoming connections.

• SDU

Use the SDU parameter to specify the session data unit (SDU) size for connections.

SERVICE RATE

The SERVICE_RATE parameter of cman.ora file specifies incoming connection rate that is allowed per service for an instance.

SESSION TIMEOUT

The SESSION_TIMEOUT parameter of the cman.ora file specifies the maximum time in seconds allowed for a user session.

SSL CIPHER SUITES

Use the SSL_CIPHER_SUITES parameter to control the combination of authentication, encryption, and data integrity algorithms used by Transport Layer Security (TLS).

SSL CLIENT AUTHENTICATION

Use the SSL_CLIENT_AUTHENTICATION parameter to specify whether a client is authenticated using Transport Layer Security (TLS).

SSL VERSION

Use the ${\tt SSL_VERSION}$ parameter to define valid Transport Layer Security (TLS) versions to be used for connections.

TRACE FILE

USE SID AS SERVICE

The USE_SID_AS_SERVICE Oracle Connection Manager parameter enables the system identifier (SID) in the connect descriptor to be interpreted as a service name when a user attempts a database connection.

VALID NODE CHECKING REGISTRATION

The VALID_NODE_CHECKING_REGISTRATION parameter of the cman.ora file specifies whether valid node checking registration is performed.

WALLET LOCATION

8.2.1 ADDRESS

The ADDRESS networking parameter specifies the protocol address of Oracle Connection Manager.

Purpose

To specify the protocol address of Oracle Connection Manager.

Syntax

 $(\texttt{ADDRESS=(PROTOCOL=}protocol)\ (\texttt{HOST=}host_name)\ (\texttt{PORT=}port_number)$

Example

(ADDRESS=(PROTOCOL=tcp) (HOST=sales-server) (PORT=1521)



8.2.2 ADMIN_GROUP

The ADMIN_GROUP parameter of the cman.ora file specifies the user group that can perform the Oracle Connection Manager (CMAN) administration.

Purpose

To allow users of the specified group to perform CMAN administration operations. Users of the group are allowed to run Oracle Connection Manager control utility commands, such as show status, show services, show gateways, show rules, and so on.

Usage Notes

You can include <code>ADMIN_GROUP</code> parameter in the <code>PARAMETER_LIST</code> section of the <code>cman.ora</code> file to specify the group that can run CMAN control utility commands and carry out administration operations.

Default

By default, only CMAN user can perform the connection manager administration operations.

Example

ADMIN GROUP=dba

Related Topics

Oracle Connection Manager Control Utility Commands
 Use the Oracle Connection Manager Control utility commands to manage and configure
 Oracle Connection Manager instances.

8.2.3 ASO_AUTHENTICATION_FILTER

It is a networking parameter for Oracle Connection Manager. It instructs Oracle Connection Manager to check the connection requests for Secure Network Services (SNS).

Purpose

To specify whether Oracle Database security authentication settings must be used by the client.

Usage Notes

The global setting can be overridden by a rule-level setting in ACTION LIST.

Values

- on to instruct Oracle Connection Manager to reject connection requests that are not using Secure Network Services (SNS). SNS is part of Oracle Database security.
- off to instruct Oracle Connection Manager not to check for SNS between the client and server. This is the default.



8.2.4 COMPRESSION

The COMPRESSION parameter of the cman.ora file enables or disables data compression.

Purpose

To enable or disable data compression. If both the Oracle Connection Manager and the other end (server or client or Oracle Connection Manager) have this parameter set to on, then compression is used for the connection.

Default

off

Values

- on to enable data compression.
- off to disable data compression.

Example

COMPRESSION=on

8.2.5 COMPRESSION_LEVELS

The COMPRESSION_LEVELS networking parameter of the cman.ora file specifies the CPU usage and compression ratio.

Purpose

To specify the compression level.

Usage Notes

The compression levels are used at the time of negotiation to verify which levels are used at both ends, and select one level.

Default

low

Values

- low for low CPU usage and a low compression ratio.
- high for high CPU usage and a high compression ratio.

Example 8-2 Example

COMPRESSION LEVELS=high, low



8.2.6 COMPRESSION_THRESHOLD

The COMPRESSION_THRESHOLD parameter of the cman.ora file specifies the minimum data size for which compression is required.

Purpose

To specify the minimum data size, in bytes, for which compression is required.

Usage Notes

Compression is not be done if the size of the data to be sent is less than this value.

Default

1024 bytes

Example

COMPRESSION_THRESHOLD=1024

8.2.7 CONNECTION_STATISTICS

CONNECTION_STATISTICS networking parameter of the cman.ora file specifies whether the SHOW CONNECTIONS command displays connection statistics.

Purpose

To specify whether the SHOW CONNECTIONS command displays connection statistics.

Usage Notes

The global setting can be overridden by a rule-level setting in ACTION LIST.

Values

- yes to display statistics.
- no to not display statistics. This is the default.

8.2.8 EVENT GROUP

EVENT_GROUP networking parameter of the cman.ora file specifies which event groups are logged.

Purpose

To specify which event groups are logged.

Usage Notes

Multiple events may be designated using a comma-delimited list.

Values

- alert for alert notifications.
- cmd proc for command processing.



- conn hdlg for connection handling.
- init and term for initialization and termination.
- memory ops for memory operations.
- proc mgmt for process management.
- reg and load for registration and load update.
- relay for events associated with connection control blocks.
- timer for gateway timeouts.
- wake_up for events related to Connection Manager Administration (CMADMIN) wake-up queue.



The event group ALERT cannot be turned off.

8.2.9 EXPIRE_TIME

The EXPIRE_TIME networking parameter of cman.ora file specifies a time interval, in minutes, to send a check to verify that client/gateway connections are active.

Purpose

To specify a time interval, in minutes, to send a check to verify that client/server connections are active.

Usage Notes

Setting a value greater than 0 ensures that connections are not left open indefinitely, due to an unusual client termination. If the system supports TCP keepalive tuning, then Oracle Net Services automatically uses the enhanced detection model, and tunes the TCP keepalive parameters

If the probe finds a terminated connection, or a connection that is no longer in use, then it returns an error, causing the server process to exit.

This parameter is primarily intended for the database server, which typically handles multiple connections at any one time.

Limitations on using this terminated connection detection feature are:

- It is not allowed on bequeathed connections.
- Though very small, a probe packet generates additional traffic that may downgrade network performance.
- Depending on which operating system is in use, the server may need to perform additional
 processing to distinguish the connection probing event from other events that occur. This
 can also result in degraded network performance.

Values

0: To disable terminated connection detection.



 Any number greater than 0: To enable terminated connection detection. The number equals the time interval in minutes.

Default

0

Example 8-3 Example

EXPIRE TIME=10

8.2.10 IDLE_TIMEOUT

Purpose

To specify the amount of time that an established connection can remain active without transmitting data.

Usage Notes

The global setting can be overridden by a rule-level setting in ACTION LIST.

Values

- 0 to disable the timeout. This is the default.
- Any number greater than 0 to enable the timeout. The number equals the timeout period in seconds.

8.2.11 INBOUND_CONNECT_TIMEOUT

The INBOUND_CONNECT_TIMEOUT parameter of the cman.ora file specifies the time limit that the Oracle Connection Manager listener waits for a valid connection before timing out.

Purpose

To specify how long, in seconds, the Oracle Connection Manager listener waits for a valid connection from a client or another instance of Oracle Connection Manager.

Values

- 60 sec is the default. Use value 0 to disable timeout.
- Any number greater than 0 to enable the timeout. The number equals the timeout period in seconds.

Example

INBOUND CONNECT TIMEOUT=30



8.2.12 LOG_FILE_NUM

The LOG_FILE_NUM networking parameter of the cman.ora file specifies the number of log file segments.

Purpose

To specify the number of log file segments. At any point of time there can be only n log file segments where n is LOG_FILE_NUM and if the log grows beyond this number, then the older segments are deleted.

Default

No default. Number of segments grow indefinitely, if not specified or set to zero.

Value

Any integer value up to the maximum integer.

Example

LOG_FILE_NUM=3

8.2.13 LOG_FILE_SIZE

 ${\tt LOG_FILE_SIZE}$ networking parameter of the cman.ora file specifies the size of each log file segment.

Purpose

To specify the size of each log file segment. The size is in MB.

Default

300 MB

Values

Any integer value.

Example 8-4 Example

LOG FILE SIZE=10

8.2.14 MAX_ALL_CONNECTIONS

Purpose

To specify the maximum number of concurrent registration and client connection sessions that can be supported by Oracle Connection Manager.

Usage Notes

This number includes registration connections from databases, and ongoing client connection establishment requests. After a connection is established, the clients do not maintain a



connection to the listener. This limit only applies to client connections that are in the initial connection establishment phase from a listener perspective.

Default

Operating system-specific

Example

MAX_ALL_CONNECTIONS=40

8.2.15 MAX_CMCTL_SESSIONS

The MAX_CMCTL_SESSIONS parameter of the cman.ora file specifies the maximum number of concurrent local or remote sessions.

Purpose

To specify the maximum number of concurrent local or remote sessions of the Oracle Connection Manager control utility allowable for a given instance.

Usage Notes

One of the sessions must be a local session.

Value

Any number of sessions can be designated.

Example

MAX CMCTL SESSIONS=6

8.2.16 MAX_CONNECTIONS

The MAX_CONNECTIONS parameter of the cman.ora file specifies the maximum number of connection slots that a gateway process can handle.

Purpose

To specify the maximum number of connection slots that a gateway process can handle.

Values

Any number in the range of 1 to 1024.

Default

The default value is 256.

Example

MAX CONNECTIONS=100



8.2.17 MAX_GATEWAY_PROCESSES

Purpose

To specify the maximum number of gateway processes that an instance of Oracle Connection Manager supports.

Values

The number designated must be greater than the minimum number of gateway processes. The maximum is 64.

8.2.18 MAX_REG_CONNECTIONS

Purpose

To specify the maximum number of concurrent registration connection sessions that can be supported by Oracle Connection Manager.

Default

512

Example

MAX REG CONNECTIONS=20

8.2.19 MIN_GATEWAY_PROCESSES

Purpose

To specify the minimum number of gateway processes that an instance of Oracle Connection Manager supports.

Values

Any number of sessions can be designated up to 64.

8.2.20 OUTBOUND_CONNECT_TIMEOUT

Purpose

To specify the length of time in seconds that the Oracle Connection Manager instance waits for a valid connection to be established with the database server or with another Oracle Connection Manager instance.

Values

- 60 to disable the timeout. This is the default.
- Any number greater than 0 to enable the timeout. The number equals the timeout period in seconds.



8.2.21 PASSWORD_instance_name

Purpose

To specify the encrypted instance password, if one has been set.

8.2.22 REGISTRATION_EXCLUDED_NODES

The Oracle Connection Manager parameter file (cman.ora)
REGISTRATION_EXCLUDED_NODES specifies the list of nodes that cannot register with the listener.

Purpose

To specify the list of nodes that cannot register with the listener.

Usage Notes

The list can include host names or CIDR notation for IPv4 and IPv6 addresses. The wildcard format (*) is supported for IPv4 addresses. The presence of a host name in the list results in the inclusion of all IP addresses mapped to the host name. The host name should be consistent with the public network interface.

If the REGISTRATION_INVITED_NODES parameter and the REGISTRATION_EXCLUDED_NODES parameter are set, then the REGISTRATION EXCLUDED NODES parameter is ignored.

Values

Valid nodes and subnet IP addresses or names.

Example

```
REGISTRATION_EXCLUDED_NODES = 10.1.26.*, 10.16.40.0/24, \
2001:DB8:3eff:fe38, node2
```

8.2.23 REGISTRATION_INVITED_NODES

The Oracle Connection Manager parameter file (cman.ora) REGISTRATION_EXCLUDED_NODES parameter specifies the list of node that can register with the listener.

Purpose

To specify the list of node that can register with the listener.

Usage Notes

The list can include host names or CIDR notation for IPv4 and IPv6 addresses. The wildcard format (*) is supported for IPv4 addresses. The presence of a host name in the list results in the inclusion of all IP addresses mapped to the host name. The host name should be consistent with the public network interface.

If the REGISTRATION_INVITED_NODES parameter and the REGISTRATION_EXCLUDED_NODES parameter are set, then the REGISTRATION EXCLUDED NODES parameter is ignored.



Values

Valid nodes and subnet IP addresses or names.

Example

8.2.24 RULE

The RULE parameter of the cman.ora file specifies an access control rule list to filter incoming connections.

Purpose

To specify an access control rule list to filter incoming connections.

Usage Notes

A rule list specifies which connections are accepted, rejected, or dropped.

If no rules are specified, then all connections are rejected.

The source and destination can be a host name, IP address, or subnet mask.

There must be at least one rule for client connections and one rule for CMCTL connections. Omitting one or the other results in the rejection of all connections for the rule type omitted. The last rule in the example that follows is a CMCTL rule.

Oracle Connection Manager does not support wildcards for partial IP addresses. If you use a wildcard, then use it in place of a full IP address. The IP address of the client may, for example, be (SRC=*).

Oracle Connection Manager supports only the /nn notation for subnet addresses. In the first rule in Example "Sample cman.ora File", /27 represents a subnet mask that comprises 27 leftmost bits.

Values

This parameter is listed in the rule list section of the cman.ora file preceded by RULE LIST=.

Syntax

```
(RULE_LIST=
  (RULE=
      (SRC=host)
      (DST=host)
      (SRV=service_name)
      (ACT={accept|reject|drop})
      (ACTION_LIST=AUT={on|off}
      ((CONN_STATS={yes|no}) (MCT=time) (MIT=time) (MOCT=time)))
      (RULE= ...))
```

Additional Parameters

The RULE parameter filters a connection or group of connections using the following parameters:

SRC: The source host name or IP address of the client.

DST: The destination server host name or IP address of the database server.

SRV: The database service name of Oracle Database obtained from the SERVICE_NAME parameter in the initialization parameter file.

ACT: The action for the connection request. Use accept to accept incoming requests, reject to reject incoming requests, or drop to reject incoming requests without sending an error message.

ACTION_LIST: The rule-level parameter settings for some parameters. These parameters are as follows:

- AUT: Oracle Database security authentication on client side.
- CONN STATS: Log input and output statistics.
- MCT: Maximum connect time.
- MIT: Maximum idle timeout.
- MOCT: Maximum outbound connect time.

Rule-level parameters override their global counterparts.

Example

```
(RULE LIST=
  (RULE=
   (SRC=client1-pc)
   (DST=sales-server)
   (SRV=sales.us.example.com)
    (ACT=reject))
  (RULE=
    (SRC=192.0.2.45)
    (DST=192.0.2.200)
    (SRV=db1)
    (ACT=accept))
  (RULE=
    (SRC=sale-rep)
    (DST=sales1-server)
    (SRV=cmon)
    (ACT=accept)))
```

8.2.25 SDU

Use the SDU parameter to specify the session data unit (SDU) size for connections.

Purpose

To specify the SDU size, in bytes, for connections

Usage Notes

Oracle Connection Manager can negotiate large SDU with client and server when configured. When the configured values of client, database server, and Oracle Connection Manager do not match for a session, the least value of all the three values is used.

Default

8192 bytes (8 KB)



Value

512 to 2097152 bytes

Example

SDU=32768

8.2.26 SERVICE RATE

The SERVICE_RATE parameter of cman.ora file specifies incoming connection rate that is allowed per service for an instance.

Purpose

To specify incoming connection rate that is allowed per service for an instance.

Usage Notes

Any user-specified value greater than 0 sets the maximum limit on the number of new connections per service-instance handled by the proxy listener every second. Listener rejects connections after it reaches the maximum limit. Client side connection failure is reported with "TNS:listener: rate limit reached".

Values

- 0 to disable service rate limit. This is the default.
- Any number greater than 0 to enable service rate limit.

Example 8-5 Example

SERVICE RATE=10

8.2.27 SESSION_TIMEOUT

The SESSION_TIMEOUT parameter of the cman.ora file specifies the maximum time in seconds allowed for a user session.

Purpose

To specify the maximum time in seconds allowed for a user session.

Usage Notes

The global setting can be overridden by a rule-level setting in ACTION LIST.

Values

- 0 to disable the timeout. This is the default.
- Any number greater than 0 to enable the timeout. The number equals the timeout period in seconds.

Example

SESSION_TIMEOUT=60



8.2.28 SSL_CIPHER_SUITES

Use the SSL_CIPHER_SUITES parameter to control the combination of authentication, encryption, and data integrity algorithms used by Transport Layer Security (TLS).

Purpose

To control the combination of authentication, encryption, and data integrity algorithms used by Transport Layer Security (TLS). By default, the strongest protocol and cipher are negotiated between the database client and server. Setting this parameter will override the default behavior. You must use this parameter only if you have internal security controls that dictate the usage of certain protocol versions.

Usage Notes

Enclose the SSL_CIPHER_SUITES parameter value in parentheses. Otherwise, the cipher suite setting does not parse correctly.

Default

None

Values

Approved ciphers compatible with TLS 1.2:

- TLS ECDHE RSA WITH AES 256 GCM SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS ECDHE ECDSA WITH AES 256 GCM SHA384
- TLS ECDHE ECDSA WITH AES 128 GCM SHA256
- TLS DHE RSA WITH AES 256 GCM SHA384
- TLS DHE RSA WITH AES 128 GCM SHA256

Deprecated ciphers compatible with TLS 1.2:

- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS ECDHE RSA WITH AES 128 CBC SHA256
- TLS ECDHE ECDSA WITH AES 256 CBC SHA384
- TLS ECDHE ECDSA WITH AES 128 CBC SHA256
- TLS RSA WITH AES 256 GCM SHA384
- TLS RSA WITH AES 256 CBC SHA256
- TLS_RSA_WITH_AES_128_GCM_SHA256
- TLS RSA WITH AES 128 CBC SHA256
- TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384
- TLS ECDH RSA WITH AES 128 GCM SHA256
- TLS ECDH RSA WITH AES 256 CBC SHA384
- TLS ECDH RSA WITH AES 128 CBC SHA256
- TLS ECDH ECDSA WITH AES 256 GCM SHA384



- TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384
- TLS ECDH ECDSA WITH AES 128 CBC SHA256
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
- TLS DHE RSA WITH AES 128 CBC SHA256
- TLS_DH_anon_WITH_AES_256_GCM_SHA384
- TLS_DH_anon_WITH_AES_128_GCM_SHA256

Deprecated ciphers compatible with TLS 1.0, TLS 1.1, and TLS 1.2:

- TLS ECDHE RSA WITH AES 256 CBC SHA
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
- TLS ECDHE ECDSA WITH AES 128 CBC SHA
- TLS RSA WITH AES 256 CBC SHA
- TLS RSA WITH AES 128 CBC SHA
- TLS ECDH RSA WITH AES 256 CBC SHA
- TLS_ECDH_RSA_WITH_AES_128_CBC_SHA
- TLS ECDH ECDSA WITH AES 256 CBC SHA
- TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA
- TLS DHE RSA WITH AES 256 CBC SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_RSA_WITH_RC4_128_SHA
- TLS ECDHE ECDSA WITH RC4 128 SHA
- TLS ECDH RSA WITH RC4 128 SHA
- TLS ECDH ECDSA WITH RC4 128 SHA
- SSL_RSA_WITH_RC4_128_SHA
- SSL_RSA_WITH_RC4_128_MD5
- TLS_ECDHE_ECDSA_WITH_NULL_SHA
- TLS_ECDHE_RSA_WITH_NULL_SHA
- TLS_ECDH_ECDSA_WITH_NULL_SHA
- TLS_ECDH_RSA_WITH_NULL_SHA
- SSL_RSA_WITH_NULL_SHA
- SSL RSA WITH NULL MD5
- SSL_DH_anon_WITH_RC4_128_MD5

Deprecated ciphers compatible with TLS 1.0 and TLS 1.1:

- TLS ECDHE RSA WITH 3DES EDE CBC SHA
- TLS ECDHE ECDSA WITH 3DES EDE CBC SHA



- TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA
- TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA
- SSL RSA WITH 3DES EDE CBC SHA
- SSL_DH_anon_WITH_3DES_EDE_CBC_SHA



The $\mathtt{DH_anon}$ cipher suites do not provide authentication of the communicating parties, and can be vulnerable to man-in-the-middle attacks. Oracle recommends that you do not use these cipher suites to protect sensitive data.

Examples

```
SSL_CIPHER_SUITES=(TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384)

SSL_CIPHER_SUITES=(TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
```

Related Topics

- Set the TLS Cipher Suites on the Server
- Set the TLS Cipher Suites on the Client

8.2.29 SSL_CLIENT_AUTHENTICATION

Use the SSL_CLIENT_AUTHENTICATION parameter to specify whether a client is authenticated using Transport Layer Security (TLS).

Purpose

To specify whether a client is authenticated using Transport Layer Security (TLS).

Usage Notes

The database server authenticates the client. Therefore, this value should be set to false. If this parameter is set to true, then the listener attempts to authenticate the client, which can result in a failure.

Default

true

Values

true | false

Example

 ${\tt SSL_CLIENT_AUTHENTICATION=false}$



See Also:

Oracle Database Security Guide

8.2.30 SSL_VERSION

Use the SSL_VERSION parameter to define valid Transport Layer Security (TLS) versions to be used for connections.

Purpose

To define the version of TLS that must run on the systems with which the database server communicates. By default, the database server and client negotiate the strongest security protocol. Oracle does not recommend modifying this parameter, unless your security requirements mandate the usage of certain protocol versions.

Usage Notes

Clients, listeners, and database servers must use compatible versions. Modify this parameter only when necessary to enforce the use of the more secure TLS protocol and not allow clients that only work with the older TLS protocols. If you need to specify TLS 1.0 or TLS 1.1, then also include TLS 1.2 to allow more secure connections. The current default uses TLS 1.2, which is the version required for multiple security compliance requirements.

If you set SSL_VERSION to undetermined, then the most secure TLS protocol version is used. You can use the SSL_VERSION=undetermined setting in the connect string for a specific connection to override the SSL_VERSION value configured in the sqlnet.ora file.

If you do not set SSL_VERSION to any value, then all the supported TLS protocol versions are tried starting with the most secure version. This is typically the most common configuration, ensuring that the strongest protocol is chosen during TLS negotiation.

Default

undetermined

Values

undetermined | 1.0 | 1.1 | 1.2

The version numbers correspond to the TLS versions, such as TLSv1.0, TLSv1.1, and TLSv1.2.



The sqlnet.ora parameter ADD SSLV3 TO DEFAULT has no impact on this parameter.

Syntax and Examples

To specify a single TLS version:

SSL_VERSION=TLS_protocol_version



For example:

```
SSL VERSION=1.2
```

To specify multiple TLS versions, use the or operator as follows:

```
SSL VERSION=TLS protocol version1 or TLS protocol version2
```

For example:

```
SSL_VERSION=1.1 or 1.2

SSL_VERSION=1.0 or 1.1 or 1.2
```

Related Topics

- Set the Required TLS Version on the Server
- Set the Required TLS Version on the Client

8.2.31 TRACE_FILE

Purpose

To specify the directory for Oracle Connection Manager trace files.

8.2.32 USE_SID_AS_SERVICE

The USE_SID_AS_SERVICE Oracle Connection Manager parameter enables the system identifier (SID) in the connect descriptor to be interpreted as a service name when a user attempts a database connection.

Purpose

To enable the system identifier (SID) in the connect descriptor to be interpreted as a service name when a user attempts a database connection.

Usage Notes

Database clients with earlier releases of Oracle Database that have hard-coded connect descriptors can use this parameter to connect to a container or pluggable database.

For an Oracle container database, the client must specify a service name in order to connect to it. Setting this parameter to on instructs the Oracle Connection Manager listener to use the SID in the connect descriptor as a service name and connect the client to the specified database.

Values

- off (default value)
- on

Example 8-6 Example

USE SID AS SERVICE=on

8.2.33 VALID_NODE_CHECKING_REGISTRATION

The VALID_NODE_CHECKING_REGISTRATION parameter of the cman.ora file specifies whether valid node checking registration is performed.

Purpose

To determine whether valid node checking registration is performed, and if the subnet is allowed.

Usage Notes

When set to on, valid node checking registration is performed at the listener for any incoming registration request, and only local IP addresses are allowed.

Default

on

Values

- off | 0 to specify valid node checking registration is off, and no checking is performed.
- on | 1 | local to specify valid node checking registration is on, and all local IP addresses can register. If a list of invited nodes is set, then all IP addresses, host names, or subnets in the list as well as local IP addresses are allowed.
- subnet | 2 to specify valid node checking registration is on, and all machines in the local subnets are allowed to register. If a list of invited nodes is set, then all nodes in the local subnets as well as all IP addresses, host names and subnets in the list are allowed.

Example

VALID_NODE_CHECKING_REGISTRATION=on

8.2.34 WALLET LOCATION

Purpose

To specify the location of wallets. Wallets are certificates, keys, and trustpoints processed by SSL.

Usage Notes

The key/value pair for Microsoft certificate store (MCS) omits the METHOD_DATA parameter because MCS does not use wallets. Instead, Oracle PKI (public key infrastructure) applications obtain certificates, trustpoints and private keys directly from the user's profile.

If an Oracle wallet is stored in the Microsoft Windows registry and the wallet's key (KEY) is SALESAPP, then the storage location of the encrypted wallet is

HKEY_CURRENT_USER\SOFTWARE\ORACLE\WALLETS\SALESAPP\EWALLET.P12. The storage location of the decrypted wallet is

HKEY CURRENT USER\SOFTWARE\ORACLE\WALLETS\SALESAPP\CWALLET.SSO.





This parameter must be specified outside Oracle Connection Manager alias

Syntax

The syntax depends on the wallet, as follows:

Oracle wallets on the file system:

```
WALLET_LOCATION=
  (SOURCE=
    (METHOD=file)
  (METHOD_DATA=
        (DIRECTORY=directory)
        [(PKCS11=TRUE/FALSE)]))
```

Microsoft certificate store:

Oracle wallets in the Microsoft Windows registry:

```
WALLET_LOCATION=
   (SOURCE=
          (METHOD=reg)
          (METHOD_DATA=
                (KEY=registry_key)))
```

Entrust wallets:

Additional Parameters

WALLET_LOCATION supports the following parameters:

- SOURCE: The type of storage for wallets, and storage location.
- METHOD: The type of storage.
- METHOD DATA: The storage location.
- DIRECTORY: The location of Oracle wallets on file system.
- KEY: The wallet type and location in the Microsoft Windows registry.
- PROFILE: The Entrust profile file (.epf).
- INIFILE: The Entrust initialization file (.ini).

Default

None



Values

true | false

Examples

Oracle wallets on file system:

Microsoft certificate store:

```
WALLET_LOCATION=
(SOURCE=
(METHOD=mcs))
```

Oracle Wallets in the Microsoft Windows registry:

```
WALLET_LOCATION=
(SOURCE=
(METHOD=REG)
(METHOD_DATA=
(KEY=SALESAPP)))
```

Entrust Wallets:

8.3 Oracle Connection Manager in Traffic Director Mode Parameters

This section lists and describes the cman.ora file parameters.

SERVICE AFFINITY

Use the cman.ora parameter SERVICE_AFFINITY to modify the default load distribution mechanism for Oracle Connection Manager in Traffic Director Mode.

- TDM
- TDM BIND THREAD
- TDM DATATYPE CHECK
- TDM_PRCP_MAX_CALL_WAIT_TIME
- TDM_PRCP_MAX_TXN_CALL_WAIT_TIME
- TDM SHARED THREADS MAX
- TDM_SHARED_THREADS_MIN
- TDM_THREADING_MODE



8.3.1 SERVICE_AFFINITY

Use the cman.ora parameter SERVICE_AFFINITY to modify the default load distribution mechanism for Oracle Connection Manager in Traffic Director Mode.

Purpose

To configure load distribution mechanism for Oracle Connection Manager in Traffic Director Mode. By default, Oracle Connection Manager in Traffic Director Mode uses service affinity to select a gateway for routing incoming connection requests. All new connection requests are routed to the gateways associated with database services.

Usage Notes

If you set this parameter to on, then all new connection requests are routed to the gateways associated with database services.

If you set this parameter to OFF, then all new connection requests are routed to the least-loaded gateways.

When using Proxy Resident Connection Pooling (PRCP), Oracle recommends that you set the SERVICE_AFFINITY parameter to OFF for better performance and resource utilization of gateway processes.

Values

ON | OFF

Default

ON

Example

```
SERVICE AFFINITY = {ON | OFF}
```

Related Topics

- Oracle Database Net Services Administrator's Guide
- Oracle Multitenant Administrator's Guide

8.3.2 TDM

Purpose

To configure Oracle Connection Manager to act as Oracle Connection Manager in Traffic Director Mode.

Default

FALSE

Values

TRUE



FALSE

Example

tdm = TRUE

8.3.3 TDM BIND THREAD

Purpose

To make the application connection hold on to the TDM thread and has different implications with and without PRCP. This parameter only applies when <code>TDM_THREADING_MODE</code> is set to <code>SHARED</code>.

Usage Notes

Without PRCP, setting this parameter to yes makes the application connection hold on the TDM worker thread as long as there is a transaction in progress.

With PRCP, setting this parameter to yes makes the application connection hold on to the TDM thread from the time <code>OCISessionGet</code> is done by the application till it does an <code>OCISessionRelease</code>.

Default

no

Values

- yes
- no

Example

TDM BIND THREAD = yes

8.3.4 TDM_DATATYPE_CHECK

Purpose

To validate all the inbound data to the database, of the data type <code>NUMBER</code>, <code>DATE</code>, <code>TIMESTAMP</code>, <code>TIMESTAMP</code> WITH <code>LOCAL TIMEZONE</code>, <code>TIMESTAMP</code> WITH <code>TIMEZONE</code>, <code>BLOB</code>, <code>CLOB</code>, <code>BFILE</code>, <code>UROWID</code> and <code>REF</code>. The following error is received by the application if there is any problem with the data sent to the Oracle Connection Manager in Traffic Director Mode.

ORA-03137: malformed TTC packet from client rejected: [3101]

Usage Notes

Turning ON/OFF this parameter enables or disables the data validation.

Default

OFF



Values

- ON
- OFF

Example

tdm datatype check={ON | OFF}

8.3.5 TDM_PRCP_MAX_CALL_WAIT_TIME

Purpose

To record the maximum time of inactivity, in seconds, for a client after obtaining a session from the PRCP pool. This parameter is applicable when the Oracle Connection Manager in Traffic Director Mode is configured to have Proxy Resident Connection Pool.

Usage Notes

After obtaining a session from the PRCP pool, if the client application does not issue a database call for the time specified by <code>TDM_PRCP_MAX_CALL_WAIT_TIME</code> parameter, then the PRCP session is freed and the client connection is terminated. As a result, if the client application attempts a round trip call on such a connection, then it receives an <code>ORA-3113</code> or <code>ORA-3115</code> error.

Default

30 seconds

Values

Any non negative value. However, Oracle recommends not to use a value of 0 as that implies that a connection can acquire a PRCP session for an indefinite amount of time

8.3.6 TDM_PRCP_MAX_TXN_CALL_WAIT_TIME

Purpose

To record the maximum time of inactivity, in seconds, for a client after it obtains a session from the Proxy Resident Connection Pool and starts a transaction. This parameter is applicable when the Oracle Connection Manager in Traffic Director Mode is configured to have PRCP.

Usage Notes

If the client application does not issue a database call for the time specified by <code>TDM_PRCP_MAX_TXN_CALL_WAIT_TIME</code> parameter while in a transaction, the PRCP session is freed, the transaction is rolled back, and the client connection is terminated. As a result, if the client application attempts a round trip call on such a connection, then it receives an <code>ORA-3113</code> or <code>ORA-3115</code> error.

Default

0



Values

Any nonnegative value. However, it is recommended not to use a value of 0 as it implies that a connection can acquire a PRCP session for an indefinite amount of time.

8.3.7 TDM SHARED THREADS MAX

Purpose

To configure the maximum number of threads that an Oracle Connection Manager process in Traffic Director Mode should have, when tdm threading mode is set to SHARED.

Values

Any number can be designated for the maximum number of threads. For DEDICATED mode, the maximum number of threads is same as the maximum number of connections. In SHARED mode, though there is no fixed upper bound, it should ideally be proportional to the load.

8.3.8 TDM_SHARED_THREADS_MIN

Purpose

To configure the minimum number of threads that an Oracle Connection Manager process in Traffic Director Mode should have, when tdm_threading_mode is set to SHARED.

Values

Any number can be designated for the minimum number of threads. For SHARED mode, there is no limit enforced. However, the number of threads should be proportional to the load.

8.3.9 TDM THREADING MODE

Purpose

To configure the usage of threads by the Oracle Connection Manager in Traffic Director Mode.

Usage Notes

If this parameter is set to <code>DEDICATED</code>, then a worker thread is spawned for each inbound connection and the maximum number of threads is determined by the <code>max_connections</code> parameter

If this parameter is set to SHARED, then a shared pool of worker threads handle all inbound connections. The minimum number of worker threads is specified by the tdm_shared_threads_min setting and the maximum number of worker threads is specified by the tdm_shared_threads_max setting. The thread pool is internally managed within these bounds.

Default

DEDICATED

Values

DEDICATED



SHARED

Example

```
tdm_threading_mode={DEDICATED | SHARED}
tdm_shared_threads_min = 4
tdm_shared_threads_max = 5
```

8.4 ADR Diagnostic Parameters for Oracle Connection Manager

The diagnostic data for critical errors is quickly captured and stored in the ADR for Oracle Connection Manager.

Since Oracle Database 11g, Oracle Database includes an advanced fault diagnosability infrastructure for preventing, detecting, diagnosing, and resolving problems. The problems are critical errors such as those caused by database code bugs, metadata corruption, and customer data corruption.

When a critical error occurs, it is assigned an incident number, and diagnostic data for the error, such as traces and dumps, are immediately captured and tagged with the incident number. The data is then stored in the Automatic Diagnostic Repository (ADR), a file-based repository outside the database.

This section describes the parameters used when ADR is enabled. ADR is enabled by default. Non-ADR parameters listed in the cman.ora file are ignored when ADR is enabled.

ADR BASE

It is a diagnostic parameter in the cman.ora file and it specifies the base directory to store tracing and logging incidents when ADR is enabled.

DIAG ADR ENABLED

 ${\tt DIAG_ADR_ENABLED} \ \ diagnostic \ parameter \ of \ the \ {\tt cman.ora} \ file \ indicates \ whether \ ADR \ tracing \ is \ enabled.$

- LOG LEVEL
- TRACE_LEVEL
- TRACE TIMESTAMP

8.4.1 ADR BASE

It is a diagnostic parameter in the $\mathtt{cman.ora}$ file and it specifies the base directory to store tracing and logging incidents when ADR is enabled.

Purpose

To specify the base directory to store tracing and logging incidents when ADR is enabled.

Default

The default is ORACLE BASE, or ORACLE HOME/log if ORACLE BASE is not defined.

Values

Any valid directory path to a directory with write permission.

Example 8-7 Example

ADR BASE=/oracle/network/trace

8.4.2 DIAG ADR ENABLED

DIAG_ADR_ENABLED diagnostic parameter of the cman.ora file indicates whether ADR tracing is enabled.

Purpose

To indicate whether ADR tracing is enabled.

Usage Notes

When the <code>DIAG_ADR_ENABLED</code> parameter is set to <code>OFF</code>, then non-ADR file tracing is used.

Values

on | off

Example 8-8 Example

DIAG ADR ENABLED=on

8.4.3 LOG_LEVEL

Purpose

To specify the level of logging performed by Oracle Connection Manager.

Usage Notes

This parameter is also applicable when non-ADR logging is used.

The following log files are used with Oracle Connection Manager:

- instance-name pid.log for the listener.
- instance-name cmadmin pid.log for CMADMIN.
- instance-name cmgw pid.log for the gateway processes.

The log files are located in the ORACLE HOME/network/log directory.

Default

off or 0

Values

- off or 0 for no log output.
- user or 4 for user log information.
- admin or 10 for administration log information.
- support or 16 for Oracle Support Services log information.

Example

LOG_LEVEL=admin



8.4.4 TRACE_LEVEL

Purpose

To specify the trace level for the Oracle Connection Manager instance.

Usage Notes

This parameter is also applicable when non-ADR tracing is used.

The following trace files are used with Oracle Connection Manager:

- instance-name_pid.trc for the listener.
- instance-name_cmadmin_pid.trc for CMADMIN.
- instance-name cmgw pid.trc for the gateway processes.

The log files are located in the ORACLE HOME/network/log directory.

Default

off

Values

- off for no trace output.
- user for user trace information.
- admin for administration trace information.
- support for Oracle Support Services trace information.

Example

TRACE LEVEL=admin

8.4.5 TRACE TIMESTAMP

Purpose

To add a time stamp in the form of dd-mmm-yyyy hh:mi:ss:mil to every trace event in the trace file for the listener.

Usage Notes

This parameter is used with the TRACE_LEVEL parameter. This parameter is also applicable when non-ADR tracing is used.

Default

on

Values

- on or true
- off or false



Example

TRACE TIMESTAMP=true

8.5 Non-ADR Diagnostic Parameters for Oracle Connection Manager

This section lists the parameters used when ADR is disabled:

- LOG_DIRECTORY
- TRACE_DIRECTORY
- TRACE_FILELEN
- TRACE_FILENO

8.5.1 LOG_DIRECTORY

Purpose

To specify the location of Oracle Connection Manager log files.

Usage Notes

Use this parameter when ADR is not enabled.

Default

ORACLE_BASE_HOME/network/log

Values

Any valid directory path to a directory with write permission.

Example

LOG_DIRECTORY=/oracle/network/log

8.5.2 TRACE_DIRECTORY

Purpose

To specify the location of the Oracle Connection Manager trace files.

Usage Notes

Use this parameter when ADR is not enabled.

Default

ORACLE BASE HOME/network/trace

Values

Any valid directory path to a directory with write permission.



Example

TRACE DIRECTORY=/oracle/network/admin/trace

8.5.3 TRACE_FILELEN

Purpose

To specify the size, in KB, of the trace file.

Usage Notes

When the size is met, the trace information is written to the next file. The number of files is specified with the TRACE_FILENO parameter. Any size can be designated. Use this parameter when ADR is not enabled.

Default

Unlimited

Example

TRACE FILELEN=100

8.5.4 TRACE_FILENO

Purpose

To specify the number of trace files for Oracle Connection Manager tracing.

Usage Notes

When this parameter is set along with the TRACE_FILELEN parameter, trace files are used in a cyclical fashion. The first file is filled first, then the second file, and so on. When the last file has been filled, the first file is reused, and so on. Any number of files can be designated.

The trace file names are distinguished from one another by their sequence number. For example, if this parameter is set to 3, then the gateway trace files would be named <code>instancename cmgw1 pid.trc</code>, <code>instance name cmgw2 pid.trc</code> and <code>instance name cmgw3 pid.trc</code>.

In addition, trace events in the trace files are preceded by the sequence number of the file. Use this parameter when ADR is not enabled.

Default

1

Example

TRACE FILENO=3



9

Directory Usage Parameters in the Idap.ora File

This chapter provides a complete listing of the ldap.ora file configuration parameters.

Overview of Directory Server Usage File

The ldap.ora file contains directory usage configuration parameters created by Oracle Internet Directory Configuration Assistant, or Oracle Net Configuration Assistant. Do not modify these parameters or their settings.

Directory Usage Parameters
 This section lists and describes the following ldap.ora file configuration parameters.

9.1 Overview of Directory Server Usage File

The ldap.ora file contains directory usage configuration parameters created by Oracle Internet Directory Configuration Assistant, or Oracle Net Configuration Assistant. Do not modify these parameters or their settings.

When created with Oracle Internet Directory Configuration Assistant, ldap.ora is located in the ORACLE_HOME/ldap/admin directory. When created with Oracle Net Configuration Assistant, the ldap.ora file is located in the ORACLE_HOME/network/admin directory. The ldap.ora file can also be stored in the directory specified by the LDAP ADMIN or TNS ADMIN environment variable.

Related Topics

- Oracle Internet Directory
- Oracle Net Configuration Assistant

9.2 Directory Usage Parameters

This section lists and describes the following ldap.ora file configuration parameters.

DEFAULT_ADMIN_CONTEXT

DEFAULT_ADMIN_CONTEXT ldap.ora file configuration parameter specifies the default directory for the creation, modification, or search of the connect identifiers.

DIRECTORY_SERVER_TYPE

DIRECTORY_SERVER_TYPE is a networking parameter of theldap.ora file and it specifies the type of directory server that is being used.

DIRECTORY SERVERS

DIRECTORY_SERVERS is a directory usage parameter and it lists the host names and port number of the primary and alternate LDAP directory servers.

9.2.1 DEFAULT_ADMIN_CONTEXT

DEFAULT_ADMIN_CONTEXT ldap.ora file configuration parameter specifies the default directory for the creation, modification, or search of the connect identifiers.

Purpose

To specify the default directory entry that contains an Oracle Context from which connect identifiers can be created, modified, or looked up.

Values

Valid distinguished name (DN)

Example 9-1 Example

DEFAULT ADMIN CONTEXT="o=OracleSoftware,c=US"

9.2.2 DIRECTORY_SERVER_TYPE

DIRECTORY_SERVER_TYPE is a networking parameter of theldap.ora file and it specifies the type of directory server that is being used.

Purpose

To specify the type of directory server that is being used.

Values

- oid for Oracle Internet Directory
- ad for Microsoft Active Directory

Example 9-2 Example

DIRECTORY SERVER TYPE=oid

9.2.3 DIRECTORY SERVERS

DIRECTORY_SERVERS is a directory usage parameter and it lists the host names and port number of the primary and alternate LDAP directory servers.

Purpose

To list the host names and port number of the primary and alternate LDAP directory servers.

Values

host:port[:sslport]

Example 9-3 Example

DIRECTORY SERVERS=(ldap-server1:389:636, ldap-server2:389:636)



Appendices

Review information about features no longer supported in this release, upgrade concerns, and information about the Oracle Net Services LDAP schema.

- Features Not Supported in this Release
 This appendix describes features no longer supported by Oracle Net Services.
- Upgrade Considerations for Oracle Net Services
 This appendix describes the coexistence and upgrade issues for Oracle Net Services.
- LDAP Schema for Oracle Net Services

 This appendix describes the Oracle schema object classes and attributes defined in the directory server for Oracle Net Services objects. It does not describe object classes and attributes reserved for future functionality or used by other Oracle products.

A

Features Not Supported in this Release

This appendix describes features no longer supported by Oracle Net Services.

Overview of Unsupported Features

This section describes the features and the configuration file that are no longer being supported in Oracle Database.

Unsupported Parameters

This section describes the unsupported or obsolete parameters.

Unsupported Control Utility Commands

This section describes the control utility commands not supported by this release.

Unsupported or Deprecated Protocols

This section describes the protocols not supported or deprecated since Oracle Database 12c.

A.1 Overview of Unsupported Features

This section describes the features and the configuration file that are no longer being supported in Oracle Database.

Oracle Net Connection Pooling

In Oracle Database 12c Release 2 (12.2), Oracle Net connection pooling is no longer supported

Oracle Names

Oracle Names is not supported in this release.

Oracle Net Listener Password

In Oracle Database 12c Release 2 (12.2), the Oracle Net Listener password feature is no longer supported.

A.1.1 Oracle Net Connection Pooling

In Oracle Database 12c Release 2 (12.2), Oracle Net connection pooling is no longer supported

Deprecation of Oracle Net Connection Pooling

It was deprecated in Oracle Database 11*g* release. Refer to My Oracle Support note 1469466.1.

A.1.2 Oracle Names

Oracle Names is not supported in this release.

Naming Method

Oracle Names has not been supported as a naming method since Oracle Database 11g. You must migrate to directory naming.

A.1.3 Oracle Net Listener Password

In Oracle Database 12c Release 2 (12.2), the Oracle Net Listener password feature is no longer supported.

Oracle Net Listener Password Support

In Oracle Database 12c Release 2 (12.2), the Oracle Net Listener password feature is no longer supported. This does not cause a loss of security because authentication is enforced through local operating system authentication.

A.2 Unsupported Parameters

This section describes the unsupported or obsolete parameters.

Table A-1 Unsupported Networking Parameters

File	Parameter	Description	Last Supported Release
sqlnet.ora	SQLNET.KERBEROS5_CO NF_MIT	This parameter was used to specify that MIT Kerberos configuration format was used. Starting with Oracle Database 12c Release 2 (12.2), only the current MIT Kerberos configuration is supported.	11.2
sqlnet.ora	SQLNET.ALLOWED_LOGO N_VERSION	This parameter has been divided into SQLNET.ALLOWED_LOGO N_VERSION_CLIENT and SQLNET.ALLOWED_LOGO N_VERSION_SERVER.	11.2

A.3 Unsupported Control Utility Commands

This section describes the control utility commands not supported by this release.



Table A-2 Unsupported Network Control Utility Commands

Control Utility	Commands	Description	Last Supported Release
Oracle Names Control Utility	All commands	Oracle Names is no longer supported.	9.2

A.4 Unsupported or Deprecated Protocols

This section describes the protocols not supported or deprecated since Oracle Database 12c.

Table A-3 Unsupported Protocols

Protocol	Description	Last Supported Release
NT LAN Manager (NTLM) protocol for domain authentication	NTLM domain authentication has been deprecated from the Oracle Windows adapter. Only Kerberos authentication is used for the NTS adapter.	11.2
	NTLM is still used for local user authentication, as well as in the case in which the database service runs as a local user.	



B

Upgrade Considerations for Oracle Net Services

This appendix describes the coexistence and upgrade issues for Oracle Net Services.

Anonymous Access to Oracle Internet Directory
 Typical users of directory naming (LDAP) require anonymous access to the Oracle Internet
 Directory for name lookup.

B.1 Anonymous Access to Oracle Internet Directory

Typical users of directory naming (LDAP) require anonymous access to the Oracle Internet Directory for name lookup.

Oracle Internet Directory Setting

If you upgrade your Oracle Internet Directory software release 11g or later, then the default setting for Oracle Internet Directory changes to disallow anonymous access to the directory. The directory administrator must configure the directory to enable anonymous binds after upgrading the directory to release 11g. In addition, the way anonymous binds are configured in Oracle Internet Directory changed between Oracle Database 10g and Oracle Database 11g.



C

LDAP Schema for Oracle Net Services

This appendix describes the Oracle schema object classes and attributes defined in the directory server for Oracle Net Services objects. It does not describe object classes and attributes reserved for future functionality or used by other Oracle products.

Structural Object Classes

The Oracle schema supports the structural object classes for Oracle Net directory naming lookups.

Attributes

It lists the attributes used for the object classes. This list is subject to change.

C.1 Structural Object Classes

The Oracle schema supports the structural object classes for Oracle Net directory naming lookups.

Table C-1 Oracle Net Structural Object Classes

Object Class	Attributes	Description
orclDBServer	orclNetDescNameorclVersion	Defines the attributes for database service entries.
orclNetAddress	orclNetAddressStringorclNetProtocolorclVersion	Specifies a listener protocol address.
orclNetAddressAux1	orclNetHostname	Specifies an auxiliary object class to add attributes to an orclNetAddress entry.
orclNetAddressList	 orclNetAddrList orclNetFailover orclNetLoadBalance orclNetSourceRoute orclVersion 	Specifies a list of protocol addresses.
orclNetDescription	 orclNetAddrList orclNetInstanceName orclNetConnParamList orclNetFailover orclNetLoadBalance orclNetSdu orclNetServiceName orclNetSourceRoute orclSid orclVersion 	Specifies a connect descriptor containing the protocol address of the listener and the connect information to the service.

Table C-1 (Cont.) Oracle Net Structural Object Classes

Object Class	Attributes	Description
		<u>'</u>
orclNetDescriptionAux1	orclNetSendBufSize	Specifies auxiliary object class to add attributes to an
	orclNetReceiveBufSize	orclNetDescription entry.
	 orclNetFailoverModeString 	ordinerbescription entry.
	 orclNetInstanceRole 	
orclNetDescriptionList	 orclNetDescList 	Specifies a list of connect
	 orclVersion 	descriptors.
orclNetService	 orclNetDescName 	Defines the attributes for network
	 orclVersion 	service name entries.
orclNetServiceAlias	 orclNetDescName 	Defines the attributes for network
	 orclVersion 	service alias entries.

C.2 Attributes

It lists the attributes used for the object classes. This list is subject to change.

Table C-2 LDAP Schema Attributes for Oracle Net Services

Attribute	Description
orclCommonContextMap	Allows the mapping of more than one default Oracle Context in the directory server.
orclNetAddrList	Identifies one or more listener protocol addresses.
orclNetAddressString	Defines a listener protocol address.
orclNetConnParamList	Placeholder for connect data parameters.
orclNetDescList	Identifies one or more connect descriptors.
orclNetDescName	Identifies a connect descriptor or a list of connect descriptors.
orclNetFailover	Turns connect-time failover on for a protocol address list.
orclNetFailoverModeString	Instructs Oracle Net to fail over to a different listener if the first listener fails during runtime. Depending on the configuration, session or any SELECT statements that were in progress are automatically failed over.
orclNetHostname	Specifies the host name.
orclNetInstanceName	Specifies the instance name to access.
orclNetInstanceRole	Specifies a connection to the primary or secondary instance of an Oracle Real Application Clusters (Oracle RAC) configuration.
orclNetLoadBalance	Turns client load balancing on for a protocol address list.
orclNetProtocol	Identifies the protocol used in the orclAddressString attribute.
orclNetReceiveBufSize	Specifies the buffer space limit for receive operations of sessions.



Table C-2 (Cont.) LDAP Schema Attributes for Oracle Net Services

Attribute	Description
orclNetSdu	Specifies the session data unit (SDU) size.
orclNetSendBufSize	Specifies the buffer space limit for send operations of sessions.
orclNetServiceName	Specifies the database service name in the CONNECT_DATA portion.
orclNetSourceRoute	Instructs Oracle Net to use each address in order until the destination is reached.
orclSid	Specifies the Oracle system identifier (SID) in the CONNECT_DATA portion of a connection descriptor.
orclVersion	Specifies the version of software used to create the entry.



Glossary

access control list (ACL)

The group of access directives that you define. The directives grant levels of access to specific data for specific clients or groups of clients.

ACL

See access control list (ACL).

access control

A feature of Oracle Connection Manager that sets rules for denying or allowing certain clients to access designated servers.

address

See protocol address.

ADR

See Automatic Diagnostic Repository (ADR).

alias

An alternative name for a network object in a server. An alias stores the name of the object it is referencing. When a client requests a lookup of an alias, Oracle completes the lookup as if it is the referenced object.

application gateway

A host computer that runs the Oracle Net Firewall Proxy. An application gateway looks and acts like a real server from the client's point of view, and a real client from the server's point of view. An application gateway sits between the Internet and company's internal network and provides middleman services (or proxy services) to users on either side.

ASCII character set

American Standard Code for Information Interchange character set, a convention for representing alphanumeric information using digital data. The collation sequence used by most computers with the exception of IBM and IBM-compatible computers.

attribute

A piece of information that describes an aspect of a directory entry. An entry comprises a set of attributes, each of which belongs to an object class. Moreover, each attribute has both a type, which describes the kind of information in the attribute, and a value which contains the actual data.

authentication method

A security method that enables you to have high confidence in the identity of users, clients, and servers in distributed environments. Network authentication methods can also provide the benefit of single sign-on for users. The following authentication methods are supported:

- Kerberos
- Microsoft Azure Active Directory (Azure AD) user authentication and authorization for Oracle Autonomous Cloud Databases (See Oracle Database Security Guide)
- Microsoft Windows NT native authentication
- Oracle Cloud Infrastructure (OCI) Identity and Access Management (IAM) user authentication and authorization for Oracle Autonomous Cloud Databases (See Oracle Database Security Guide)
- RADIUS
- Transport Layer Security (TLS)

Automatic Diagnostic Repository (ADR)

Automatic Diagnostic Repository is a systemwide central repository for tracing and logging files. The repository is a file-based hierarchical data store for depositing diagnostic information.

cache

Memory that stores recently-accessed data to so that subsequent requests to access the same data can be processed quickly.

CIDR

Classless Inter-Domain Routing. In CIDR notation, an IPv6 subnet is denoted by the subnet prefix and the size in bits of the prefix (in decimal), separated by the slash (/) character. For example, 2001:0db8:0000:0000::/64 denotes a subnet with addresses

2001:0db8:000:0000:0000:0000:0000:0000 through

2001:0db8:000:0000:FFFF:FFFF:FFFF. The CIDR notation includes support for IPv4



addresses. For example, 192.0.2.1/24 denotes the subnet with addresses 192.0.2.1 through 192.0.2.255.

Classless Inter-Domain Routing (CIDR)

See CIDR.

client

A user, software application, or computer that requests the services, data, or processing from another application or computer. The client is the user process. In a network environment, the client is the local user process and the server may be local or remote.

client load balancing

Load balancing, whereby if more than one listener services a single database, a client can randomly choose between the listeners for its connect requests. This randomization enables all listeners to share the burden of servicing incoming connect requests.

client profile

The properties of a client, which may include the preferred order of naming methods, client and server logging and tracing, the domain from which to request names, and other client options.

client/server architecture

Software architecture based on a separation of processing between two CPUs. One CPU acts as the client in the transaction, requesting and receiving services. The other acts as the server that provides service for the requests.

cman.ora file

An Oracle Connection Manager configuration file that specifies protocol addresses for incoming requests and administrative commands, as well as Oracle Connection Manager parameters and access control rules.

CMADMIN (Connection Manager Administration)

An Oracle Connection Manager process that monitors the health of the listener and Oracle Connection Manager gateway processes, shutting down and starting processes as needed. CMADMIN registers information about gateway processes with the listener and processes commands run with the Oracle Connection Manager Control utility.

CMGW (Connection Manager gateway)

An Oracle Connection Manager process that receives client connections screened and forwarded by the listener located at the Oracle Connection Manager instance. The gateway

process forwards the requests to the database server. In addition, it can multiplex or process multiple client connections through a single protocol connection.

connect data

A portion of the connect descriptor that defines the destination database service name or Oracle system identifier (SID). In the following example, SERVICE_NAME defines a database service called sales.us.example.com:

connect descriptor

A specially-formatted description of the destination for a network connection. A connect descriptor contains destination service and network route information.

The destination service is indicated by using its service name. The network route provides, at a minimum, the location of the listener through use of a network address.

connect identifier

A connect descriptor or a name that maps to a connect descriptor. A connect identifier can be a network service name, database service name, or network service alias. Users initiate a connect request by passing a user name and password along with a connect identifier in a connect string for the service to which they want to connect:

```
CONNECT username@connect identifier
```

connect string

Information the user passes to a service to connect, such as user name, and connect identifier:

```
CONNECT username@net_service_name
```

connect-time failover

A client connect request is forwarded to a another listener if a listener is not responding. Connect-time failover is enabled by service registration, because the listener knows if an instance is running to attempt a connection.

connection

An interaction between two processes on a network. Connections are originated by an initiator (client) that requests a connection with a destination (server).

connection load balancing

The method for balancing the number of active connections for the same service across the instances and dispatchers. Connection load balancing enables listeners to make routing decisions based on how many connections for each dispatcher and the load on the nodes.

connection pooling

A resource utilization and user scalability feature used to maximize the number of sessions over a limited number of protocol connections to a shared server.

connection request

A notification sent by an initiator and received by a listener that indicates that the initiator wants to start a connection.

data packet

See packet.

database link

A pointer that defines a one-way communication path from an Oracle database server to another database server. Public and private database links are a defined entries in a data dictionary table. Global database links are stored in an LDAP directory and can be accessed by all users on the network. To access public and private links, the user must be connected to the local database that contains the data dictionary entry.

A client connected to local database A can use a public or private link stored in database A to access information in remote database B. However, users connected to database B cannot use the same link to access data in database A. If local users on database B want to access data on database A, then a link must be defined and stored in the data dictionary of database B. Global links may be used between any clients and database on the network.

The following database links are supported:

- A private database link in a specific schema of a database. Only the owner of a private database link can use it.
- A public database link for a database. All users in the database can use it.
- A global database link is a database link stored in the LDAP directory.

dedicated connection

A dedicated server with a database session.

dedicated server

A server process that is dedicated to one client connection. Contrast with shared server.

default domain

The domain within which most client requests take place. It could be the domain where the client resides, or it could be a domain from which the client requests network services often. Default domain is also the client configuration parameter that determines what domain should be appended to unqualified network name requests. A name request is unqualified if it does not have a period (.) character within it.

directory information tree (DIT)

A hierarchical tree-like structure in a directory server of the distinguished names (DNs) of the entries. This structure is specific to x500 and LDAP.

directory naming

A naming method that resolves a database service, network service name, or network service alias to a connect descriptor stored in a central directory server. A directory server provides central administration of directory naming objects, reducing the work effort associated with adding or relocating services.

directory server

A directory server that is accessed with the Lightweight Directory Access Protocol (LDAP). Support of LDAP-compliant directory servers provides a centralized method for managing and configuring a distributed Oracle network. The directory server can replace client-side and server-side localized thinnames.ora files.

dispatcher

A process that enables many clients to connect to the same server without the need for a dedicated server process for each client. A dispatcher handles and directs multiple incoming network session requests to shared server processes.

distinguished name (DN)

Name of entry in a directory server. The DN specifies where the entry resides in the LDAP directory hierarchy, similar to the way a directory path specifies the exact location of a file.

distributed processing

Division of front-end and back-end processing to different computers. Oracle Net Services supports distributed processing by transparently connecting applications to remote databases.

domain

Any tree or subtree within the Domain Name System (DNS) namespace. Domain most commonly refers to a group of computers whose host names share a common suffix, the domain name.



Domain Name System (DNS)

A system for naming computers and network services that is organized into a hierarchy of domains. DNS is used in TCP/IP networks to locate computers through user-friendly names. DNS resolves a friendly name into an IP address, which is understood by computers.

For Oracle Net Services, DNS translates the host name in a TCP/IP address into an IP address.

DNS

See Domain Name System (DNS).

enterprise role

An enterprise role is analogous to a regular database role, except that it spans authorization on multiple databases. An enterprise role is a category of roles that define privileges on a particular database. An enterprise role is created by the database administrator of a particular database. An enterprise role can be granted to or revoked from one or more enterprise users. The information for granting and revoking these roles is stored in the directory server.

enterprise user

A user that has a unique identity across an enterprise. Enterprise users connect to individual databases through a schema. Enterprise users are assigned enterprise roles that determine their access privileges on databases.

entry

The building block of a directory server, it contains information about an object of interest to directory users.

external naming

A **naming method** that uses a third-party naming service, such as Network Information Service (NIS).

external procedure

Function or procedure written in a third-generation language (3GL) that can be called from PL/SQL code. Only C is supported for external procedures.

failover

See connect-time failover.



firewall support

See access control.

FTP

File Transfer Protocol. A client/server protocol which allows a user on one computer to transfer files to and from another computer over a TCP/IP network.

global database link

A database link definition stored in an LDAP directory which can be accessed by all users on the network. This definition is the same as the one used for client connections to the database (name/connect-descriptor).

Global database links cannot include user or password clauses. They only work when the database initiating the link uses the identity of the existing client to establish the link.

global database name

The full name of the database which uniquely identifies it from any other database. The global database name is of the form "database_name.database_domain," for example, sales.us.example.com.

The database name portion, sales, is a simple name to call a database. The database domain portion, us.example.com, specifies the database domain which the database is located, making the global database name unique. When possible, Oracle recommends that your database domain mirror the network domain.

The global database name is the default service name of the database, as specified by the SERVICE NAMES parameter in the initialization parameter file.

Heterogeneous Services

An integrated component that provides the generic technology for accessing non-Oracle systems from the Oracle database server. Heterogeneous Services enables you to:

- Use Oracle SQL to transparently access data stored in non-Oracle systems as if the data resides within an Oracle server.
- Use Oracle procedure calls to transparently access non-Oracle systems, services, or application programming interfaces (APIs), from your Oracle distributed environment.

hierarchical naming model

An infrastructure in which names are divided into multiple hierarchically-related domains.

host naming

A naming method resolution that enables users in a TCP/IP environment to resolve names through their existing name resolution service. This name resolution service might be Domain Name System (DNS), Network Information Service (NIS), or simply a centrally-maintained set

of /etc/hosts files. Host naming enables users to connect to an Oracle database server by simply providing the server computer's host name or host name alias. No client configuration is required to take advantage of this feature. This method is recommended for simple TCP/IP environments.

HTTP

Hypertext Transfer Protocol. A protocol that provides the language that enables Web browsers and application Web servers to communicate.

identity management realm

A collection of identities, all of which are governed by the same administrative policies. In an enterprise, all employees having access to the intranet may belong to one realm, while all external users who access the public applications of the enterprise may belong to another realm. An identity management realm is represented in the directory by a specific entry with a special object class associated with it.

instance

The combination of the System Global Area (SGA) and the Oracle background processes. When a database is started on a database server (regardless of the type of computer), Oracle allocates a memory area called the SGA, and starts one or more Oracle processes. The memory and processes of an instance efficiently manage the associated database data and serve the database users. You can connect to any instance to access information within a cluster database.

instance name

A name of an Oracle database instance. The instance name is identified by the INSTANCE_NAME parameter in the database initialization parameter file. INSTANCE_NAME corresponds to the Oracle system identifier (SID) of the instance. Clients can connect to a specific instance by specifying the INSTANCE_NAME parameter in the connect descriptor.

The instance name is included in the connect data part of the connect descriptor.

IP address

Used to identify a node on a network. Each computer on the network is assigned a unique Internet Protocol (IP) address, which is made up of the network ID, and a unique host ID. This address is typically represented in dotted-decimal notation, with the decimal value of each octet separated by a period, for example 192.0.2.22.



IPC

Interprocess Communication is a protocol used by client applications that resides on the same node as the listener to communicate with the database. IPC can provide a faster local connection than TCP/IP.

IPv4

Internet Protocol Version 4. IPv4 is the current standard for the IP protocol. IPv4 uses 32-bit (four-byte) addresses, which are typically represented in dotted-decimal notation. The decimal value of each octet is separated by a period, as in 192.0.2.22.

IPv6

Internet Protocol Version 6. The protocol designed to replace IPv4. In IPv6, an IP address is typically represented in eight fields of hexadecimal values separated by colons, as in 2001:0db8:0000:0000:0000:0000:1428:57AB. In some cases, fields with 0 values can be compressed, as in 2001:DB8::1428:57AB.

IP Version 4 (IPv4)

See IPv4.

IP Version 6 (IPv6)

See IPv6.

Java Database Connectivity (JDBC) Driver

A driver that provides Java applications and applets access to an Oracle database.

JDBC OCI Driver

A Type II driver for use with client/server Java applications. This driver requires an Oracle client installation.

JDBC Thin Driver

A Type IV driver for Oracle JDBC applets and applications. Because it is written entirely in Java, this driver is platform-independent. It does not require any additional Oracle software on the client side. The Thin driver communicates with the server using Two-Task Common (TTC), a protocol developed by Oracle to access the database server.

Kerberos

A network authentication service that strengthens security in distributed environments. Kerberos is a trusted third-party authentication system that relies on shared secrets and assumes that the third party is secure. It provides single sign-on capabilities and database link

authentication (MIT Kerberos only) for users, provides centralized password storage, and enhances PC security.

keyword-value pair

The combination of a keyword and a value, used as the standard unit of information in connect descriptors and many configuration files. Keyword-value pairs may be nested; that is, a keyword may have another keyword-value pair as its value.

latency

The amount of time it takes to send a request and receive an answer.

LDAP Data Interchange Format (LDIF)

See LDIF

Idap.ora file

A file created by Oracle Internet Directory Configuration Assistant or Oracle Net Configuration Assistant that contains the following directory server access information:

- Type of directory server
- Location of the directory server
- Default Oracle Context that the client or server use to look up or configure connect identifiers for connections to database services

When created with Oracle Internet Directory Configuration Assistant, the ldap.ora file is located in the ORACLE_HOME/ldap/admin directory. When created with Oracle Net Configuration Assistant, the ldap.ora file is located in the ORACLE HOME/network/admin directory.

LDIF

LDAP Data Interchange Format (LDIF) is the set of standards for formatting an input file for any of the LDAP command line utilities.

Lightweight Directory Access Protocol (LDAP)

A standard, extensible directory access protocol. It is a common language that LDAP clients and servers use to communicate. The framework of design conventions supporting industry-standard directory servers.

link qualifier

An extension to the database link name which specifies the connect name used to connect to the database. It provides alternate settings for the database user name and password credentials. For example, a link qualifier of fieldrep can be appended to a global database link of sales.us.example.com.

SQL> SELECT * FROM emp@sales.us.example.com@fieldrep

listener

See Oracle Net Listener.

Listener Control utility

A utility included with Oracle Net Services to control listener functions, such as starting, stopping, and getting the status of the listener.

listener.ora file

A configuration file for the listener that identifies the following for a listener:

- Unique name
- Protocol addresses that it is accepting connection requests on
- Services it is listening for

The listener.ora file typically resides in the ORACLE HOME/network/admin directory.

Oracle does not require identification of the database service because of service registration. However, static service configuration is required if you plan to use Oracle Enterprise Manager Cloud Control.

Listener Registration (LREG)

As a part of service registration, LREG registers instance information with the listener. LREG is an instance background process of each database instance that is configured in the database initialization parameter file.

load balancing

A feature by which client connections are distributed evenly among multiple listeners, dispatchers, instances, and nodes so that no single component is overloaded.

Oracle Net Services support client load balancing and connection load balancing.

local naming

A naming method that locates network addresses by using information configured and stored on each individual client's tnsnames.ora file. Local naming is most appropriate for simple distributed networks with a small number of services that change infrequently.

location transparency

A distributed database characteristic that enables applications to access data tables without knowing where they reside. All data tables appear to be in a single database, and the system determines the actual data location based on the table name. The user can reference data on multiple nodes in a single statement, and the system automatically and transparently routes

(parts of) SQL statements to remote nodes for execution if needed. The data can move among nodes with no impact on the user or application.

logging

A feature in which errors, service activity, and statistics are written to a log file. The log file provides additional information for an administrator when the error message on the screen is inadequate to understand the failure. The log file, by way of the error stack, shows the state of the software at various layers.

See also tracing.

loopback test

A connection from the server back to itself. Performing a successful loopback verifies that Oracle Net is functioning on the database server.

map

Files used by the Network Information Service (NIS) ypserv program to handle name requests.

Microsoft Active Directory

An LDAP-compliant directory server included with Microsoft Windows 2000 Server. It stores information about objects on the network, and makes this information available to users and network administrators. Active Directory also provides access to resources on the network using a single logon process.

Microsoft Active Directory can be configured as a directory naming method to store service information that clients can access.

Microsoft Windows NT native authentication

An authentication method that enables a client single login access to a Microsoft Windows NT server and a database running on the server.

Named Pipes protocol

A high-level interface protocol providing interprocess communications between clients and servers using distributed applications. Named Pipes enables client/server conversation over a network using Named Pipes protocol.

naming context

A subtree that resides entirely on one directory server. It is a contiguous subtree, that is, it must begin at an entry that serves as the top of the subtree, and extend downward to either leaf entries or references to subordinate naming contexts. It can range in size from a single entry to the entire directory information tree (DIT).

Oracle Context can be created under a naming context.

naming method

The resolution method used by a client application to resolve a connect identifier to a connect descriptor when attempting to connect to a database service. Oracle Net provides four naming methods:

- Domain Name System (DNS)
- directory naming
- Easy Connect naming
- external naming

network service alias

An alternative name for a directory naming object in a directory server. A directory server stores network service aliases for any defined network service name or database service. A network service alias entry does not have connect descriptor information. Instead, it only references the location of the object for which it is an alias. When a client requests a directory lookup of a network service alias, the directory determines that the entry is a network service alias and completes the lookup as if it was actually the entry it is referencing.

network service name

A simple name for a service that resolves to a connect descriptor. Users initiate a connect request by passing a user name and password along with a network service name in a connect string for the service to which they want to connect:

```
CONNECT username/password@net_service_name
```

Depending on your needs, network service names can be stored in a variety of places, including:

- Local configuration file, tnsnames.ora, on each client
- Directory server
- External naming service, such as NIS

network

A group of two or more computers linked through hardware and software to allow the sharing of data and peripherals.

network administrator

The person who performs network management tasks such as installing, configuring, and testing network components. The administrator typically maintains the configuration files, connect descriptors and service names, aliases, and public and global database links.

network character set

As defined by Oracle, the set of characters acceptable for use as values in keyword-value pairs (that is, in connect descriptors and configuration files). The set includes alphanumeric uppercase, and lowercase, and some special characters.

Network Information Service (NIS)

The client/server protocol for distributing system configuration data such as user and host names between computers on a network. This service was formerly known as "Sun Microsystems Yellow Pages (yp)."

Network Interface (NI)

A network layer that provides a generic interface for Oracle clients, servers, or external processes to access Oracle Net functions. The network interface layer handles the break and reset requests for a connection.

network listener

See listener.

network object

Any service that can be directly addressed on a network, such as a listener.

network protocol

See Oracle protocol support.

Network Program Interface

An interface for server-to-server interactions that performs all of the functions that the Oracle Call Interface (OCI) does for clients, allowing a coordinating server to construct SQL requests for additional servers.

Network Session (NS)

A session layer that is used in typical Oracle Net connections to establish and maintain the connection between a client application and a database server.

NIS

See Network Information Service (NIS).

node

A computer or terminal that is part of a network

object class

In a directory server, a named group of attributes. To assign attributes to an entry, do so by assigning the object classes that hold those attributes to that entry.

All objects associated with the same object class share the attributes of that object class.

OCI

See Oracle Call Interface (OCI).

OPI

See Oracle Program Interface (OPI).

Open Systems Interconnection (OSI)

Open Systems Interconnection is a network architecture model developed by ISO as a framework for international standards in heterogeneous computer network architecture.

The OSI architecture has seven layers, from lowest to highest:

- Physical layer
- 2. Data link layer
- 3. Network layer
- 4. Transport layer
- Session layer
- Presentation layer
- Application layer

Oracle Advanced Security

An Oracle product that provides Transparent Data Encryption (TDE) and data redaction.

Oracle Call Interface (OCI)

An application programming interface (API) that enables creation of applications that use the native procedures or function calls of a third-generation language to access an Oracle database server and control all phases of SQL statement execution. OCI supports the data types, calling conventions, syntax, and semantics of a number of third-generation languages including C, C++, COBOL and FORTRAN.

Oracle Connection Manager

A router through which a client connection request may be sent either to its next hop or directly to the database server. Clients who route their connection requests through Oracle Connection

Manager can then take advantage of the session multiplexing, access control, or protocol conversion features configured for that Oracle Connection Manager.

Oracle Connection Manager Control utility

A utility included with Oracle Net Services to control various functions, such as starting, stopping, and getting the status of Oracle Connection Manager.

Oracle Context

An entry in an LDAP-compliant Internet directory called <code>cn=OracleContext</code>, under which all Oracle software relevant information is kept, including entries for Oracle Net Services directory naming and checksumming security. There may be one or more than one Oracle Context in a directory. An Oracle Context entry can be associated with a directory naming context.

Oracle Internet Directory automatically creates an Oracle Context at the root of the DIT structure. This root Oracle Context has a DN of dn:cn=OracleContext.

Oracle Enterprise Manager Cloud Control

A separate Oracle product that combines a graphical console, agents, common services, and tools to provide an integrated and comprehensive systems management platform for managing Oracle products.

Oracle Identity Management

An infrastructure enabling deployments to manage centrally and securely all enterprise identities and their access to various applications in the enterprise.

Oracle Internet Directory

A directory server implemented as an application on the Oracle database. It enables retrieval of information about dispersed users and network resources. It combines Lightweight Directory Access Protocol (LDAP) Version 3, the open Internet standard directory server access protocol, with the high performance, scalability, robustness, and availability of the Oracle database.

Oracle Net

Communication software that enables a network session from a client application to an Oracle database server. After a network session is established, Oracle Net acts as a data courier for the client application and the database server. It is responsible for establishing and maintaining the connection between the client application and database server, as well as exchanging messages between them. Oracle Net can perform these jobs because it is located on each computer in the network.

Oracle Net Configuration Assistant

A postinstallation tool that configures basic network components after installation, including:

- Listener names and protocol addresses
- Naming methods the client uses to resolve connect identifiers
- Net service names in a tnsnames.ora file
- Directory server usage

Oracle Net Firewall Proxy

Product offered by some firewall vendors that supplies Oracle Connection Manager functionality.

Oracle Net foundation layer

A networking communication layer that is responsible for establishing and maintaining the connection between the client application and server, as well as exchanging messages between them.

Oracle Net Listener

A process that resides on the server whose responsibility is to listen for incoming client connection requests and manage the traffic to the server.

When a client requests a network session with a database server, a listener receives the actual request. If the client information matches the listener information, then the listener grants a connection to the database server.

Oracle Net Manager

A tool that combines configuration abilities with component control to provide an integrated environment for configuring and managing Oracle Net Services.

You can use Oracle Net Manager to configure the following network components:

Naming

Define connect identifiers and map them to connect descriptors to identify the network location and identification of a service. Oracle Net Manager supports configuration of connect descriptors in a local tnsnames.ora file or directory server.

Naming Methods

Configure the ways in which connect identifiers are resolved into connect descriptors.

Listeners

Create and configure listeners to receive client connections.

Oracle Net Services

A suite of networking components that provide enterprise-wide connectivity solutions in distributed, heterogeneous computing environments. Oracle Net Services is comprised of

Oracle Net, listener, Oracle Connection Manager, Oracle Net Configuration Assistant, and Oracle Net Manager.

Oracle Program Interface (OPI)

Oracle Program Interface is the networking layer responsible for responding to each of the possible messages sent by OCI. For example, an OCI request to fetch 25 rows would have an OPI response to return the 25 rows after they have been fetched.

Oracle protocol support

A software layer responsible for mapping Transparent Network Substrate (TNS) functionality to industry-standard protocols used in the client/server connection.

Oracle Real Application Clusters (Oracle RAC)

An architecture that allows multiple instances to access a shared database of data files. Oracle RAC is also a software component that provides the necessary cluster database scripts, initialization files, and data files needed for Oracle Enterprise Edition and Oracle RAC.

Oracle Rdb

A database for Digital's 64-bit platforms. Because Oracle Rdb has its own listener, the client interacts with Rdb in the same manner as it does with an Oracle database.

Oracle schema

A set of rules that determine what can be stored in a directory server. Oracle has its own schema that is applied to many types of Oracle entries, including Oracle Net Services entries. The Oracle schema for Oracle Net Services entries includes the attributes the entries may contain.

Oracle system identifier (SID)

A name that identifies a specific instance of an Oracle database. For any database, there is at least one instance referencing the database.

For Oracle databases earlier than release 8.1, a SID is used to identify the database. The SID is included in the connect descriptor of a the think and in the definition of the listener in the listener ora file.

Oracle XML DB

A high-performance XML storage and retrieval technology provided with Oracle database server. It is based on the W3C XML data model.



ORACLE HOME

An alternate name for the top directory in the Oracle directory hierarchy on some directory-based operating systems.

OSI

See Open Systems Interconnection (OSI).

packet

A block of information sent over the network each time a connection or data transfer is requested. The information contained in packets depends on the type of packet, such as connect, accept, redirect, data, and so on. Packet information can be useful in troubleshooting.

PMON process

A process monitor (PMON) database process that performs process recovery when a user process fails. PMON is responsible for cleaning the cache and freeing resources that the process was using. PMON also checks on dispatcher and server processes and restarts them if they have failed.

presentation layer

A networking communication layer that manages the representation of information that application layer entities either communicate or reference in their communication. Two-Task Common (TTC) is an example of presentation layer.

private database link

A database link created by one user for exclusive use.

See also database link and public database link.

profile

A collection of parameters that specifies preferences for enabling and configuring Oracle Net Services features on the client or server. A profile is stored and implemented through the sqlnet.ora file.

protocol

A set of rules that defines how data is transported across the network.

protocol address

An address that identifies the network address of a network object.

When a connection is made, the client and the receiver of the request, such as the listener or Oracle Connection Manager, are configured with identical protocol addresses. The client uses this address to send the connection request to a particular network object location, and the recipient listens for requests on this address. It is important to install the same protocols for the client and the connection recipient, as well as configure the same addresses.

protocol conversion

A feature of Oracle Connection Manager that enables a client and server with different networking protocols to communicate with each other. This feature replaces functionality previously provided by the Oracle Multi-Protocol Interchange with SQL*Net version 2.

protocol stack

Designates a particular presentation layer and session layer combination.

proxy server

A server that substitutes for a real server, forwarding client connection requests to the real server or to other proxy servers. Proxy servers provide access control, data and system security, monitoring, and caching.

public database link

A database link created by a DBA on a local database that is accessible to all users on that database.

See also database link and private database link.

RADIUS

Remote Authentication Dial-In User Service (RADIUS) is a client-server protocol and software that enables remote access servers to communicate with a central server. This helps in authenticating dial-in users and authorizing their access to the requested system or service.

realm Oracle Context

An Oracle Context contained in each identity management realm. It stores the following information:

- User naming policy of the identity management realm, that is, how users are named and located
- Mandatory authentication attributes
- Location of groups in the identity management realm
- Privilege assignments for the identity management realm, for example, who has privileges to add more users to the realm
- Application specific data for that realm including authorizations



RDBMS

Relational Database Management System.

RDN

See relative distinguished name (RDN).

relative distinguished name (RDN)

The local, most granular level entry name. It has no other qualifying entry names that would serve to address the entry uniquely. It is a fully-qualified X.500 name. For example, cn=sales, dc=us, dc=example, dc=com, cn=sales is a RDN.

root Oracle Context

In the Oracle Identity Management infrastructure, the root Oracle Context is an entry in Oracle Net Services containing a pointer to the default identity management realm in the infrastructure. It also contains information about how to locate an identity management realm given the simple name of the realm.

RPC

Remote procedure call.

SDP

Sockets Direct Protocol.

Secure Sockets Layer (SSL)

An industry standard protocol for securing network connections. SSL provides authentication, encryption, and data integrity using public key infrastructure (PKI). The Transport Layer Security (TLS) protocol is a successor to the SSL protocol.

See The Difference Between Transport Layer Security and Secure Sockets Layer.

server parameter file

A binary file containing initialization parameter settings that is maintained on the Oracle Database host. You cannot manually edit this file with a text editor. A server parameter file is initially built from a text initialization parameter file by means of the CREATE SPFILE statement or created directly.

server process

Database processes that handle a client request on behalf of a database.

service

A program that responds to requests from various clients or performs some operation. The database is a service that stores and retrieves data for clients.

service handler

A process that acts a connection point from the listener to the database server. A service handler can be a dispatcher or dedicated server.

service name

A logical representation of a database, which is the way a database is presented to clients. The service name is a string that is the global database name, that is, a name comprised of the database name and domain name, entered during installation or database creation. If you are not sure what the global database name is, then you can obtain it from the value of the SERVICE_NAMES parameter in the initialization parameter file.

The service name is included in the connect data part of the connect descriptor.

service registration

A feature by which the Listener Registration (LREG) automatically registers information with a listener. Because this information is registered with the listener, the listener.ora file does not need to be configured with this static information.

Service registration provides the listener with information about:

- Service names for each running instance of the database
- Instance names of the database
- Service handlers (dispatcher or dedicated server) available for each instance
 These enable the listener to direct a client request appropriately.
- Dispatcher, instance, and node load information

This load information enables the listener to determine which dispatcher can best handle a client connection request. If all dispatchers are blocked, then the listener can spawn a dedicated server for the connection.

session data unit (SDU)

A buffer that Oracle Net uses to place data before transmitting it across the network. Oracle Net sends the data in the buffer either when requested or when it is full.

session layer

A network layer that provides the services needed by the protocol address entities that enable them to organize and synchronize their dialog and manage their data exchange. This layer

establishes, manages, and terminates network sessions between the client and server. An example of a session layer is Network Session (NS).

session multiplexing

Combining multiple sessions for transmission over a single network connection to conserve the operating system's resources.

shared server

A database server that is configured to allow many user processes to share very few server processes, so the number of users that can be supported is increased. With shared server configuration, many user processes connect to a dispatcher. The dispatcher directs multiple incoming network session requests to a common queue. An idle shared server process from a shared pool of server processes picks up a request from the queue. This means that a small pool of server processes can serve a large number of clients. Contrast with dedicated server.

shared server process

A process type used with shared server configuration.

SID

See Oracle system identifier (SID).

SID_LIST_listener_name

A section of the <code>listener.ora</code> file that defines the Oracle system identifier (SID) of the database served by the listener. This section is valid only for Oracle databases release 8.0, as information for Oracle8*i* or later instances is automatically registered with the listener. Static configuration is also required for other services, such as external procedure calls and Heterogeneous Services.

single sign-on

The ability for a user to log in to different servers using a single password. This permits the user to authenticate to all servers the user is authorized to access.

sqlnet.ora file

A configuration file for the client or server that specifies:

- Client domain to append to unqualified service names or net service names
- Order of naming methods the client should use when resolving a name
- · Logging and tracing features to use
- Route of connections



- External naming parameters
- Oracle Advanced Security parameters

The sqlnet.ora file typically resides in the ORACLE HOME/network/admin directory.

SSL

See Secure Sockets Layer (SSL).

System Global Area (SGA)

A group of shared memory structures that contain data and control information for an Oracle instance.

TCP/IP

Transmission Control Protocol/Internet Protocol. The standard communication protocol used for client/server conversation over a network.

TCP/IP with TLS protocol

A protocol that enables an Oracle application on a client to communicate with remote Oracle databases through the TCP/IP and Transport Layer Security (TLS).

tick

The amount of time it takes for a message to be sent and processed from the client to the server or from the server to the client.

TNS

See Transparent Network Substrate (TNS).

tnsnames.ora file

A configuration file that maps network service names to connect descriptors. This file is used for the local naming method. The tnsnames.ora file typically resides in the <code>ORACLE_HOME/network/admin</code> directory.

tracing

A facility that writes detailed information about an operation to an output file. The trace facility produces a detailed sequence of statements that describe the events of an operation as they are run. Administrators use the trace facility for diagnosing an abnormal condition. It is not normally turned on.

See also logging.

Transparent Application Failover (TAF)

A runtime failover for high-availability environments, such as Oracle Real Application Clusters and Oracle Fail Safe, that refers to the failover and re-establishment of application-to-service connections. It enables client applications to automatically reconnect to the database if the connection fails, and, optionally, resume a SELECT statement that was in progress. This reconnect happens automatically from within the Oracle Call Interface (OCI) library.

Transparent Network Substrate (TNS)

A foundation technology, built into the Oracle Net foundation layer that works with any standard network transport protocol.

transport

A networking layer that maintains end-to-end reliability through data flow control and error recovery methods. The Oracle Net foundation layer uses Oracle protocol support for the transport layer.

Transport Layer Security (TLS)

An industry standard protocol for securing network connections. The TLS protocol is a successor to the Secure Sockets Layer (SSL) protocol. It provides authentication, encryption, and data integrity using public key infrastructure (PKI).

See The Difference Between Transport Layer Security and Secure Sockets Layer.

TTC

See Two-Task Common (TTC).

Two-Task Common (TTC)

A presentation layer type that is used in a typical Oracle Net connection to provide character set and data type conversion between different character sets or formats on the client and server.

UPI

User Program Interface.

virtual circuit

A piece of shared memory used by the dispatcher for client database connection requests and replies. The dispatcher places a virtual circuit on a common queue when a request arrives. An idle shared server picks up the virtual circuit from the common queue, services the request, and relinquishes the virtual circuit before attempting to retrieve another virtual circuit from the common queue.



WebDAV protocol

World Wide Web Distributed Authoring and Versioning. A protocol with a set of extensions to HTTP which allows users to manage files on remote Web servers.



Index

Symbols	ASO_AUTHENTICATION_FILTER networking
	parameter, 8-6
.IGNORE_ANO_ENCRYPTION_FOR_TCPS networking parameter, 6-30	attributes
() (parenthesis) symbol	orclCommonContextMap, C-2
reserved in configuration files, 3-2	orclDescList, <i>C-2</i> orclDescName, <i>C-2</i>
# (quotation mark) symbol	
reserved in configuration files, 3-2	orclLoadBalance, <i>C-2</i> orclNetAddrList, <i>C-2</i>
= (equal sign) symbol	orclNetAddrString, C-2
reserved in configuration files, 3-2	orclNetConnParamList, C-2
reserved in comiguration mes, 3-2	orclNetFailover, <i>C-2</i>
	orclNetFailoverModeString, C-2
Numerics	orclNetHostname, <i>C-2</i>
1024 port 4.4	orclNetInstanceName, C-2
1024 port, <i>4-4</i> 1521 port, <i>4-4</i>	orclNetInstanceRole, C-2
1575 port, 4-4	orclNetProtocol, C-2
1630 port, 4-4	orclNetReceiveBufSize, C-2
1646 port, 5-60	orclNetSdu, C-2
1830 port, 4-4	orclNetSendBufSize, C-2
2482 port, <i>4-4</i>	orclNetServiceName, C-2
2484 port, <i>4-4</i>	orclNetSourceRoute, C-2
	orclSid, C-2
Λ.	orclVersion, C-2
A	authentication ability, 5-32
ACCEPT_MD5_CERTS networking parameter,	automatic diagnostic repository, 8-30
5-7	
ACCEPT_SHA1_CERTS networking parameter,	В
5-7, 5-8	
ACT networking parameter, 8-16	BEQUEATH_DETACH networking parameter, 5-8
ACTION_LIST networking parameter, 8-16	
ADDRESS networking parameter, 4-1, 6-7, 7-3,	С
8-5	
ADDRESS_LIST networking parameter, 4-2, 6-9	character sets
ADMIN_GROUP networking parameter, 8-6	for net service name, 3-3
ADMINISTER command, 2-4	network, for keyword values, 3-2
ADR, 8-30	class of secure transports parameters
ADR diagnostic parameters	See COST parameters, 7-33
sqlnet.ora	client load balancing
ADR_BASE, 5-91	configuring, 6-11
DIAG_ADR_ENABLED, 5-91	CLOSE CONNECTIONS command, 2-5
ADR_BASE diagnostic parameter, 5-91, 8-30	cman.ora file
ADR_BASE_diagnostic parameter, 7-27	control parameters
ALLOW_MULTIPLE_REDIRECTS_listener_name	USE_SID_AS_SERVICE_listener_name,
control parameter, 7-11	8-22
anonymous access, <i>B-1</i>	diagnostic parameters
	ADR_BASE, 8-30



cman.ora file (continued)	cman.ora file (continued)
diagnostic parameters (continued)	parameters (continued)
DIAG_ADR_ENABLED, 8-31	TDM_SHARED_THREADS_MIN, 8-29
LOG_DIRECTORY, 8-33	TDM_THREADING_MODE, 8-29
LOG_LEVEL, 8-31	TRACE FILE, 8-22
TRACE_DIRECTORY, 8-33	WALLET LOCATION, 8-23
TRACE_FILELEN, 8-34	rule list section, 8-1
TRACE_FILENO, 8-34	COLOCATION_TAG networking parameter, 6-16
TRACE LEVEL, 8-32	combining COST parameters, 7-34
TRACE_TIMESTAMP, 8-32	comments in configuration files, 3-2
example, 8-1	COMPRESSION networking parameter, 6-60, 8-7
listening address section, 8-1	Compression networking parameters, 6-60
networking parameters	COMPRESSION_LEVELS networking parameter,
LOG_FILE_NUM, 8-11	6-61, 8-7
LOG_FILE_NOM, 0 11 LOG_FILE_SIZE, 8-11	COMPRESSION_THRESHOLD networking
parameter list section, 8-1	parameter, 8-8
parameters	connect descriptors, 6-2
ACT, 8-16	CONNECT_DATA networking parameter, 6-17
ACTION_LIST, 8-16	CONNECT_TIMEOUT, 6-56
ADDRESS, 8-5	CONNECT_TIMEOUT networking parameter,
ADMIN_GROUP, 8-6	6-56
ASO_AUTHENTICATION_FILTER, 8-6	CONNECTION_RATE_listener_name
COMPRESSION, 8-7	configuration parameter, 7-7
COMPRESSION_LEVELS, 8-7	CONNECTION_STATISTICS networking
COMPRESSION_THRESHOLD, 8-8	parameter, 8-8
CONNECTION_STATISTICS, 8-8	connections
DST, 8-16	adjusting listener queue size to avoid errors,
EVENT_GROUP, 8-8	7-4
EXPIRE_TIME, 8-9	control parameters
IDLE_TIMEOUT, 8-10	listener.ora
INBOUND_CONNECT_TIMEOUT, 8-10	SSL_CLIENT_AUTHENTICATION, 5-69,
MAX_ALL_CONNECTIONS, 8-11	7-20, 8-20
MAX_CMCTL_SESSIONS, 8-12	control utilities
MAX_CONNECTIONS, 8-12	Listener Control utility, 1-26
MAX_GATEWAY_PROCESSES, 8-13	Oracle Connection Manager Control utility,
MAX_REG_CONNECTIONS, 8-13	2-3
MIN_GATEWAY_PROCESSES, 8-13	COST parameters, 7-33
OUTBOUND_CONNECT_TIMEOUT, 8-13	combining, 7-34
PARAMETER_LIST, 8-1	DYNAMIC_REGISTRATION_listener_name,
PASSWORD_instance_name, 8-14	7-35
RULE, 8-15	SECURE_CONTROL_listener_name, 7-36
SDU, 8-16	SECURE_PROTOCOL_listener_name, 7-35
SERVICE_RATE, 8-17	SECURE_REGISTER_listener_name, 7-34
SERVICE_RATE_ <i>listener_name</i> , 7-17	CRS_NOTIFICATION_listener_name control
SESSION_TIMEOUT, 8-17	parameter, 7-12
SRC, 8-16	
SRV, 8-16	D
SSL_CIPHER_SUITES, 5-66, 7-18, 8-18	
SSL_VERSION, 5-72, 6-44, 7-22, 8-21	database resident connection pooling, 6-25
TDM, 8-26	DEDICATED_THROUGH_BROKER_LISTENER
TDM_BIND_THREAD, 8-27	networking parameter, 7-12
TDM_DATATYPE_CHECK, 8-27	DEFAULT_ADMIN_CONTEXT networking
TDM_PRCP_MAX_CALL_WAIT_TIME, 8-28	parameter, 9-2
TDM_PRCP_MAX_TXN_CALL_WAIT_TIME,	DEFAULT_SDU_SIZE networking parameter, 5-9
8-28	DEFAULT_SERVICE_listener_name control
TDM_SHARED_THREADS_MAX, 8-29	parameter, 7-12

DELAY networking parameter, 6-18	diagnostic parameters (continued)
DESCRIPTION networking parameter, 6-6, 7-3	sqlnet.ora diagnostic reference, 5-90
DESCRIPTION_LIST networking parameter, 6-6	directory naming
DIAG_ADR_ENABLED diagnostic parameter,	configuring, 5-12
5-91, 8-31	DIRECTORY_SERVER_TYPE networking
DIAG_ADR_ENABLED_listener_name diagnostic	parameter, 9-2
parameter, 7-27	DIRECTORY_SERVERS, 9-2
diagnostic parameters	DISABLE_INTERRUPT networking parameter,
cman.ora	5-9
ADR_BASE, 8-30	DISABLE_OOB networking parameter, 5-10
DIAG_ADR_ENABLED, 8-31	DISABLE_OOB_AUTO networking parameter,
LOG_DIRECTORY, 8-33	5-10
LOG_LEVEL, 8-31	DST networking parameter, 8-16
TRACE_DIRECTORY, 8-33	DYNAMIC_REGISTRATION_listener_name
TRACE_FILELEN, 8-34	COST parameter, 7-35
TRACE_FILENO, 8-34	
TRACE_LEVEL, 8-32	E
TRACE_TIMESTAMP, 8-32	<u> </u>
listener.ora, 7-29	ENABLE networking parameter, 6-10
ADR_BASE_listener_name, 7-27	ENABLE_EXADIRECT_listener_name control
LOG_DIRECTORY_listener_name, 7-31	parameter, 7-11
LOG_FILE_listener_name, 7-31	error messages
LOG_FILE_NUM_listener_name, 7-28	
LOG_FILE_SIZE_listener_name, 7-28	ORA-12170, 5-73
TRACE_DIRECTORY_listener_name,	ORA-12525, 1-13, 7-13
7-31	ORA-12535, 5-61, 5-62
	ORA-12608, 5-62
TRACE_FILE_listener_name, 7-32	EVENT_GROUP networking parameter, 8-8
TRACE_FILEAGE_listener_name, 7-32	Exadirect protocol
TRACE_FILELEN_listener_name, 7-32	parameters for addresses, 4-2
TRACE_FILENO_listener_name, 7-33	EXADIRECT_FLOW_CONTROL networking
TRACE_LEVEL_listener_name, 7-29	parameter, 5-10
TRACE_TIMESTAMP_listener_name,	EXADIRECT_RECVPOLL networking parameter,
7-30	5-11
Oracle Net Listener diagnostic reference, 7-26	EXIT command
sqlnet.ora	Listener Control utility, 1-5
ADR BASE, 5-91	Oracle Connection Manager Control utility,
DIAG_ADR_ENABLED, 5-91	2-7
LOG_DIRECTORY_CLIENT, 5-94	EXPIRE_TIME networking parameter, 8-9
LOG_DIRECTORY_SERVER, 5-95	external naming
LOG_FILE_CLIENT, 5-95	Network Information Service (NIS), 5-12
LOG_FILE_SERVER, 5-96	Network information Service (NIS), 3-12
TRACE DIRECTORY CLIENT, 5-96	
TRACE_DIRECTORY_SERVER, 5-96	F
TRACE_FILE_CLIENT, 5-97	failover
TRACE_FILE_SERVER, 5-97	connect-time, 6-10
TRACE_FILEAGE_CLIENT, 5-97	Transparent Application Failover, 6-17
TRACE_FILEAGE_SERVER, 5-98	FAILOVER networking parameter, 6-10, 6-11
TRACE_FILELEN_CLIENT, 5-98	FAILOVER MODE networking parameter, 6-17
TRACE_FILELEN_SERVER, 5-99	Firewall networking parameter, 7-4
TRACE_FILENO_CLIENT, 5-99	5 F · · · · · · · · · · ·
TRACE_FILENO_SERVER, 5-99	0
TRACE_LEVEL_CLIENT, 5-92	G
TRACE_LEVEL_SERVER, 5-92	CLOBAL NAME potugrating parameter C 10
TRACE_TIMESTAMP_CLIENT, 5-93	GLOBAL_NAME networking parameter, 6-18
TRACE_TIMESTAMP_SERVER, 5-93	
TRACE_UNIQUE_CLIENT, 5-100	

Н	Listener Control utility (continued)
HELP command	commands (continued)
	SET INBOUND_CONNECT_TIMEOUT,
of Listener Control utility, 1-6	1-13
of Oracle Connection Manager Control utility,	SET LOG_DIRECTORY, 1-14
2-7	SET LOG_FILE, 1-14
heterogeneous services, 6-19	SET LOG_STATUS, 1-15
HS networking parameter, 6-19	SET TRC_DIRECTORY, 1-17
HTTPS_PROXY networking parameter, 6-7	SET TRC_FILE, 1-18
HTTPS_PROXY_PORT networking parameter,	SET TRC_LEVEL, 1-18
6-8	SET USE_PLUGANDPLAY, 1-19
	SHOW, 1-2, 1-19
1	SHOW CURRENT_LISTENER, 1-19
	SHOW DISPLAYMODE, 1-19
IDLE_TIMEOUT networking parameter, 8-10	SHOW INBOUND_CONNECT_TIMEOUT,
improving	1-19
client load balancing, 6-11	SHOW LOG_DIRECTORY, 1-19
	SHOW LOG_FILE, 1-19
INBOUND_CONNECT_TIMEOUT networking	SHOW LOG_STATUS, 1-19
parameter, 8-10	SHOW RAWMODE, 1-19
INBOUND_CONNECT_TIMEOUT_listener_name	
control parameter, 7-13	SHOW SAVE_CONFIG_ON_STOP, 1-19
INBOUND_CONNECT_TIMEPUT_listener_name	SHOW TRC_DIRECTORY, 1-19
networking parameter, 7-13	SHOW TRC_FILE, 1-19
INSTANCE_NAME networking parameter, 6-19	SHOW TRC_LEVEL, 1-19
IP networking parameter, 7-4	START, 1-21
IPC protocol	STATUS, <i>1-23</i>
addresses, 4-2	STOP, <i>1-25</i>
KEY parameter, 4-2	TRACE, <i>1-25</i>
IPC, parameters for addresses, 4-2	VERSION, 1-26
IPC.KEYPATH networking parameter, 5-11	distributed operation, 1-3
	function of and syntax format, 1-1
K	remote administration, 1-3
N.	Listener Control utility access, 1-3
keepalive feature, 6-10	listener.ora file
keyword syntax rules for configuration files, 3-2	ADR diagnostic parameters, 7-26
keyword values and network character sets, 3-2	configuration parameters
Reyword values and network character sets, 5 2	CONNECTION_RATE_listener_name, 7-7
	RATE_LIMIT, 7-7
L	control parameters, 7-10, 7-24
	ALLOW_MULTIPLE_REDIRECTS_listener_name, 7-11
Idap.ora file	CRS_NOTIFICATION_listener_name, 7-12
DEFAULT_ADMIN_CONTEXT parameter, 9-2	DEFAULT_SERVICE_listener_name, 7-12
DIRECTORY_SERVER_TYPE parameter,	ENABLE_EXADIRECT_listener_name, 7-11
9-2	INBOUND CONNECT TIMEOUT listener name, 7-13
Listener Control utility	MAX_ALL_CONNECTIONS_listener_name, 7-14
command reference, 1-26	MAX_REG_CONNECTIONS_listener_name, 7-15
commands	REGISTRATION_EXCLUDED_NODES_listener_name,
EXIT, 1-5	7-15, 8-14
HELP, 1-6	REGISTRATION_INVITED_NODES_listener_name, 7-1
QUIT, 1-7	8-14
RELOAD, 1-7	
SAVE_CONFIG, 1-8	REMOTE_REGISTRATION_ADDRESS_listener_name,
SERVICES, 1-9	SAVE_CONFIG_ON_STOP_listener_name, 7-17
SET, 1-2, 1-10	SERVICE_RATE_listener_name, 7-17
SET CONNECT_TIMEOUT, 1-11	SSL_CLIENT_AUTHENTICATION, 5-69, 7-20, 8-20
SET CURRENT_LISTENER, 1-11	SUBSCRIBE_FOR_NODE_DOWN_EVENT_listener_na
SET DISPLAYMODE, 1-12	7-23

listener.ora file (continued)	LOG_FILE_NUM_listener_name diagnostic
control parameters (continued)	parameter, 7-28
USE_SID_AS_SERVICE_listener_name, 7-23	LOG_FILE_SERVER diagnostic parameter, 5-96
VALID_NODE_CHECKING_REGISTRATION_listene	ek@amelLE_SIZE networking parameter, 8-11
8-23	LOG_FILE_SIZE_listener_name diagnostic
COST parameters, 7-33	parameter, 7-28
diagnostic parameters, 7-29	LOG_LEVEL diagnostic parameter, 8-31
ADR_BASE_listener_name, 7-27	LOGGING_listener_name
DIAG_ADR_ENABLED_listener_name, 7-27	diagnostic parameter, 7-29
LOG_DIRECTORY_listener_name, 7-31	
LOG_FILE_listener_name, 7-31	N.A.
LOG_FILE_NUM_listener_name, 7-28	M
LOG_FILE_SIZE_listener_name, 7-28	MAX_ALL_CONNECTIONS control parameter,
TRACE_DIRECTORY_listener_name, 7-31	8-11
TRACE_FILE_listener_name, 7-32	
TRACE_FILEAGE_listener_name, 7-32	MAX_ALL_CONNECTIONS_listener_name
TRACE_FILELEN_listener_name, 7-32	control parameter, 7-14
TRACE_FILENO_listener_name, 7-33	MAX_CMCTL_SESSIONS networking parameter,
TRACE_LEVEL_listener_name, 7-29	8-12
TRACE_IEVEE_insterior_name, 7-29 TRACE_TIMESTAMP_listener_name, 7-30	MAX_CONNECTIONS networking parameter,
parameters	8-12
·	MAX_GATEWAY_PROCESSES networking
ADDRESS, 7-3	parameter, 8-13
DEDICATED_THROUGH_BROKER_LISTENER, 7-1	
DESCRIPTION, 7-3	8-13
Firewall, 7-4	MAX_REG_CONNECTIONS_listener_name
INBOUND_CONNECT_TIMEOUT_listener_name, 7	ooners person, r 20
IP, 7-4	METHOD networking parameter, 6-18
QUEUESIZE, 7-4	MIN_GATEWAY_PROCESSES networking
RECV_BUF_SIZE, 7-5	parameter, 8-13
SEND_BUF_SIZE, 7-6	multiple listeners, 7-2
SSL_CIPHER_SUITES, 5-66, 7-18, 8-18	
SSL_VERSION, 5-72, 6-44, 7-22, 8-21	N
WALLET_LOCATION, 7-25	
listeners	NAMES.DEFAULT.DOMAIN networking
adjusting queue size, 7-4	parameter, 5-11
connect-request timeouts, 7-13	NAMES.DIRECTORY_PATH networking
multiple, 7-2	parameter, 5-12
load balancing	ezconnect, 5-12
client, 6-11	hostname, 5-12
LOAD_BALANCE networking parameter, 6-11	NAMES.LADP_AUTHENTICATE_BIND
local naming	networking parameter, 5-13
configuring, 5-12	NAMES.LDAP CONN TIMEOUT networking
LOCAL_REGISTRATION_ADDRESS_listener_na	parameter, 5-13
me control parameter, 7-14	NAMES.LDAP_PERSISTENT_SESSION
LOG_DIRECTORY diagnostic parameter, 8-33	networking parameter, 5-13
LOG_DIRECTORY_CLIENT diagnostic	- •
parameter, 5-94	NAMES.NIS.META_MAP networking parameter,
LOG DIRECTORY listener name diagnostic	5-14
parameter, 7-31	NAMESCTL.TRACE_UNIQUE networking
LOG_DIRECTORY_SERVER diagnostic	parameter, 5-24
parameter, 5-95	network character sets, keyword values, 3-2
LOG_FILE_CLIENT diagnostic parameter, 5-95	network configuration files
LOG_FILE_listener_name diagnostic parameter,	sqlnet.ora, 5-2, 5-90
7-31	syntax rules, 3-1
LOG_FILE_NUM networking parameter, 8-11	Network Information Service external naming
LOG_FILE_NOW Helworking parameter, 0-11	configuring, 5-12

networking parameters	Oracle Connection Manager Control utility (continued)
cman.ora	commands (continued)
LOG_FILE_SIZE, 8-11	SHOW PARAMETERS, 2-25
listener.ora	SHOW RULES, 2-26
LOG_FILE_NUM, 8-11 sqlnet.ora configuration reference, 5-2	SHOW STATUS 2.27
Squiet.ora configuration reference, 5-2	SHOW VERSION 2.28
	SHOW VERSION, 2-28 SHUTDOWN, 2-29
0	STARTUP, 2-29, 2-30
1	SUSPEND GATEWAY, 2-30
object classes	Oracle Connection Manager parameters
orclNetAddress, C-1	ADDRESS, 8-5
orclNetAddressList, C-1	ASO_AUTHENTICATION_FILTER, 8-6
orclNetDescription, <i>C-1</i>	CONNECTION_STATISTICS, 8-8
orclNetDescriptionList, C-1	EVENT GROUP, 8-8
orclNetServiceAlias, C-1	IDLE TIMEOUT, 8-10
obsolete parameters, <i>A-2</i>	INBOUND_CONNECT_TIMEOUT, 8-10
ORA-12170 error message, 5-73	MAX_CMCTL_SESSIONS, 8-12
ORA-12525 error message, <i>1-13</i> , <i>7-13</i>	MAX_CONNECTIONS, 8-12
ORA-12535 error message, <i>5-61</i> , <i>5-62</i>	MAX_GATEWAY_PROCESSES, 8-13
ORA-12608 error message, 5-62	MIN GATEWAY PROCESSES, 8-13
Oracle Connection Manager	OUTBOUND_CONNECT_TIMEOUT, 8-13
client load balancing, 6-4	PASSWORD instance name, 8-14
connect-time failover, 6-4	RULE, 8-15
SOURCE_ROUTE networking parameter,	SESSION_TIMEOUT, 8-17
6-14 Oracle Connection Manager Control utility	TDM, 8-26
Oracle Connection Manager Control utility	TDM_BIND_THREAD, 8-27
command reference, 2-3	TDM_DATATYPE_CHECK, 8-27
commands	TDM_PRCP_MAX_CALL_WAIT_TIME, 8-28
ADMINISTER, 2-4 CLOSE CONNECTIONS, 2-5	TDM_PRCP_MAX_TXN_CALL_WAIT_TIME,
EXIT, 2-7	8-28
HELP, 2-7	TDM_SHARED_THREADS_MAX, 8-29
QUIT, 2-8	TDM SHARED THREADS MIN, 8-29
RELOAD, 2-8	TDM_THREADING_MODE, 8-29
RESUME GATEWAYS, 2-9	TRACE_FILE, 8-22
SAVE_PASSWD, 2-10	Oracle Internet Directory access, <i>B-1</i>
SET, 2-10	Oracle Names support, A-2
SET ASO AUTHENTICATION FILTER, 2-11	Oracle Net Connection Pooling, A-1
SET CONNECTION_STATISTICS, 2-11	Oracle Net Listener Password, A-2
SET EVENT, 2-12, 2-13	Oracle protocol support
SET IDLE TIMEOUT, 2-13	configuring addresses, 4-2
SET INBOUND_CONNECT_TIMEOUT, 2-13	Exadirect, 4-2
SET LOG_DIRECTORY, 2-14	IPC, 4-2
SET LOG_LEVEL, 2-15, 2-18	Named Pipes, 4-2
SET OUTBOUND_CONNECT_TIMEOUT,	SDP, 4-2
2-15	TCP/IP, 4-2
SET PASSWORD, 2-16	TCP/IP with TLS, 4-2
SET SESSION TIMEOUT, 2-17	Oracle Real Application Clusters
SET TRACE_DIRECTORY, 2-17	connect-time failover, 6-11
SET TRACE_LEVEL, 2-18	FAILOVER networking parameter, 6-11
SHOW, <i>2-19</i>	INSTANCE_NAME networking parameter,
SHOW ALL, 2-19	6-20
SHOW CONNECTIONS, 2-20	orclCommonContextMap attribute, C-2
SHOW DEFAULTS, 2-23	orclDescList attribute, C-2
SHOW EVENTS, 2-24	orclDescName attribute, C-2
SHOW CATEMAYS 2-24	orclLoadBalance attribute. C-2

· · · · · · · · · · · · · · · · · · ·	QUIT command
orclNetAddressList object class, <i>C-1</i>	of Listener Control utility, 1-7
orclNetAddrList attribute, C-2	of Oracle Connection Manager Control utility,
orclNetAddrString attribute, C-2	2-8
orclNetConnParamList attribute, C-2	
orclNetDescription object class, C-1	R
orcinetDescriptionList object class, C-1	
orclNetFailover attribute, C-2	RATE_LIMIT configuration parameter, 7-7
orclNetFailoverModeString attribute, C-2	RDB_DATABASE networking parameter, 6-21
orclNetHostname attribute, C-2	RECV_BUF_SIZE, 5-24
	RECV_BUF_SIZE networking parameter, 6-12,
orclNetInstanceRole attribute, C-2	7-5
orclNetReceiveBufSize attribute, C-2	RECV_TIMEOUT networking parameter, 6-59
	reference
orclNetSendBufSize attribute, C-2	ADR for Oracle Net Listener, 7-26
orclNetServiceAlias object class, C-1	ADR for sqlnet.ora, 5-90
orclNetServiceName attribute, C-2	for Listener Control utility commands, 1-26
orclNetSourceRoute attribute, C-2	for Oracle Connection Manager Control utility
orclProtocol attribute, C-2	commands, 2-3
orclSid attribute, C-2	for sqlnet.ora, 5-2
orclVersion attribute, C-2	REGISTRATION_EXCLUDED_NODES_listener_
outbound connect timeout interval, 5-54	name control parameter, 7-15, 8-14
	REGISTRATION_INVITED_NODES_listener_na
parameter, 8-13	me control parameter, 7-15, 8-14
	RELOAD command, 2-8
P	of Listener Control utility, 1-7
	REMOTE_REGISTRATION_ADDRESS_listener_
PARAMETER_LIST networking parameter, 8-1	name control parameter, 7-16
PASSWORD_instance_name networking	RESUME GATEWAYS command, 2-9
parameter, 8-14	RETRIES networking parameter, 6-18
port 1024, 4-4	RETRY_COUNT, 6-57
port 1521, 4-4	RETRY_COUNT networking parameter, 6-57
port 1575, 4-4	RETRY_DELAY, 6-58
port 1630, 4-4	RETRY_DELAY networking parameter, 6-58
port 1646, 5-60	RULE networking parameter, 8-15
port 1830, 4-4	rules, syntax for network configuration files, 3-1
port 2483, <i>4-4</i>	
port 2484, <i>4-4</i>	S
port numbers, allowed, 4-4	
ports	SAVE_CONFIG command
privileged, 4-4	of Listener Control utility, 1-8
privileged ports, 4-4	SAVE_CONFIG_ON_STOP_listener_name
protocols	control parameter, 7-17
authentication ability, 5-32	SAVE_PASSWD command, 2-10
c	SDP protocol, parameters for addresses, 4-2
_ "	SDP.PF_INET_SDP networking parameter, 5-24
100 40	SDU networking parameter, 6-13, 8-16
	SEC_USER_AUDIT_ACTION_BANNER
parameters, 4-2	networking parameter, 5-25
SDP, 4-2	SEC_USER_UNAUTHORIZED_ACCESS_BANN
TCP/IP, 4-2	ER networking parameter, 5-25
TCP/IP with TLS, 4-2	SECURE_CONTROL_listener_name
	COST parameter, 7-36
Q	SECURE_PROTOCOL_listener_name
<u>~</u>	COST parameter, 7-35
QUEUESIZE networking parameter, 7-4	•

SECURE_REGISTER_listener_name	SET OUTBOUND_CONNECT_TIMEOUT
COST parameter, 7-34	command, <i>2-15</i>
security	SET PASSWORD command
database server	of Oracle Connection Manager Control utility,
client network timeouts, 5-60, 5-61	2-16
connect-request timeouts, 5-25, 5-29,	SET SAVE CONFIG ON STOP command, 1-16
5-30	of Listener Control utility, 1-16
listeners	SET SESSION_TIMEOUT command, 2-17
connect-request timeouts, 7-13	SET TRACE_DIRECTORY command, 2-17
remote registration, 7-15, 8-14	SET TRACE_LEVEL command, 2-18
restricting runtime administration, 7-10	SET TRC_DIRECTORY command, 1-17
SECURITY networking parameter, 6-41	SET TRC_FILE command, 1-18
SEND_BUF_SIZE networking parameter, 5-25,	SET TRC_LEVEL command, 1-18
6-13, 7-6	SET USE_PLUGANDPLAY command, 1-19
SERVER networking parameter, 6-25	SHARDING_KEY networking parameter, 6-22
server type	SHOW ALL command, 2-19
dedicated, 6-25	SHOW rommand
pooled, 6-25	of Listener Control utility, 1-19
shared, 6-25	of Oracle Connection Manager Control utility,
service name	2-19
character set keyword values, 3-3	SHOW CONNECTIONS command, 2-20
SERVICE NAME networking parameter, 6-26	SHOW CURRENT LISTENER command, 1-19
SERVICE_NAME networking parameter, 8-20 SERVICE_RATE networking parameter, 8-17	SHOW DEFAULTS command, 2-23
-	SHOW DISPLAYMODE command
SERVICE_RATE parameter, 8-17	
SERVICE_RATE_listener_name networking	of Listener Control utility, 1-19
parameter, 7-17	SHOW EVENTS command, 2-24
SERVICE_RATE_listener_name parameter, 7-17	SHOW JAPOUND, CONNECT, TIMEOUT
SERVICES command, 1-9	SHOW INBOUND_CONNECT_TIMEOUT
SESSION_TIMEOUT networking parameter, 8-17	command, 1-19
SET ASO_AUTHENTICATION_FILTER	SHOW LOG_DIRECTORY command, 1-19
command, <i>2-11</i>	SHOW LOG_FILE command, 1-19
SET command	SHOW LOG_STATUS command, 1-19
of Listener Control utility, 1-10	SHOW PARAMETERS command, 2-25
of Oracle Connection Manager Control utility,	SHOW RAWMODE command, 1-19
2-10	SHOW RULES command, 2-26
SET CONNECT_TIMEOUT command, 1-11	SHOW SAVE_CONFIG_ON_STOP command,
SET CONNECTION_STATISTICS command,	1-19
2-11	SHOW SERVICES command, 2-27
SET CURRENT_LISTENER command, 1-11	SHOW STATUS command, 2-27
SET DISPLAYMODE command	SHOW TRC_DIRECTORY command, 1-19
of Listener Control utility, 1-12	SHOW TRC_FILE command, 1-19
SET EVENT command, 2-12, 2-13	SHOW TRC_LEVEL command, 1-19
SET IDLE_TIMEOUT command, 2-13	SHOW VERSION command, 2-28
SET INBOUND_CONNECT_TIMEOUT command	SHUTDOWN command, 2-29
of Listener Control utility, 1-13	SOURCE_ROUTE networking parameter, 6-14
SET INBOUND_CONNECT_TIMEOUT	SQLNET.ALLOWED_LOGON_VERSION_CLIEN
command, of Oracle Connection Manager	T networking parameter, 5-29
Control utility, 2-13	SQLNET.ALLOWED_LOGON_VERSION_SERVE
SET LOG_DIRECTORY command	R networking parameter, 5-30
of Listener Control utility, 1-14	SQLNET.AUTHENTICATION_KERBEROS5_
of Oracle Connection Manager Control Utility,	SERVICE networking parameter, 5-24
2-14	SQLNET.AUTHENTICATION_SERVICES
SET LOG_FILE command, 1-14	networking parameter, 5-35
SET LOG_LEVEL command, 2-15, 2-18	SQLNET.CLIENT_REGISTRATION networking
SET LOG STATUS command, 1-15	parameter, 5-37

SQLNET.CLOUD_USER networking parameter,	sqlnet.ora file (continued)
5-37	diagnostic parameters (continued)
SQLNET.COMPRESSION compression	LOG_DIRECTORY_CLIENT, 5-94
parameter, 5-39	LOG_DIRECTORY_SERVER, 5-95
SQLNET.COMPRESSION_ACCELERATION	LOG_FILE_CLIENT, 5-95
compression parameter, 5-39	LOG_FILE_SERVER, 5-96
SQLNET.COMPRESSION_LEVELS compression	TRACE_DIRECTORY_CLIENT, 5-96
parameter, 5-40	TRACE_DIRECTORY_SERVER, 5-96
SQLNET.COMPRESSION_THRESHOLD	TRACE FILE CLIENT, 5-97
compression parameter, 5-40	TRACE_FILE_SERVER, 5-97
SQLNET.CRYPTO_CHECKSUM_CLIENT	TRACE_FILEAGE_CLIENT, 5-97
networking parameter, 5-40	TRACE_FILEAGE_SERVER, 5-98
SQLNET.CRYPTO_CHECKSUM_SERVER	TRACE_FILELEN_CLIENT, 5-98
networking parameter, 5-41	TRACE_FILELEN_SERVER, 5-99
SQLNET.CRYPTO CHECKSUM TYPE CLIENT	TRACE_FILENO_CLIENT, 5-99
networking parameter, 5-42	TRACE_FILENO_SERVER, 5-99
SQLNET.CRYPTO_CHECKSUM_TYPE_SERVE	TRACE_LEVEL_CLIENT, 5-92
R networking parameter, 5-42	TRACE_LEVEL_SERVER, 5-92
SQLNET.DBFW_PUBLIC_KEY networking	TRACE TIMESTAMP CLIENT, 5-93
parameter, 5-43	TRACE TIMESTAMP SERVER, 5-93
·	TRACE_TIMESTAMF_SERVER, 5-95 TRACE_UNIQUE_CLIENT, 5-100
SQLNET.DOWN_HOSTS_TIMEOUT networking	
parameter, 5-44	parameters
SQLNET.ENCRYPTION_CLIENT networking	ACCEPT_MD5_CERTS, 5-7
parameter, 5-44	ACCEPT_SHA1_CERTS, 5-7, 5-8
SQLNET.ENCRYPTION_SERVER networking	BEQUEATH_DETACH, 5-8
parameter, 5-45	DEFAULT_SDU_SIZE, 5-9
SQLNET.ENCRYPTION_TYPES_CLIENT	DISABLE_INTERRUPT, 5-9
networking parameter, 5-45	DISABLE_OOB, 5-10
SQLNET.ENCRYPTION_TYPES_SERVER	DISABLE_OOB_AUTO, 5-10
networking parameter, 5-46	EXADIRECT_FLOW_CONTROL, 5-10
SQLNET.EXPIRE_TIME networking parameter,	EXADIRECT_RECVPOLL, 5-11
5-47	IPC.KEYPATH, 5-11
SQLNET.FALLBACK_AUTHENTICATION	NAMES.DEFAULT_DOMAIN, 5-11
networking parameter, 5-49	NAMES.DIRECTORY_PATH, 5-12
SQLNET.IGNORE_ANO_ENCRYPTION_FOR_T	NAMES.LADP_AUTHENTICATE_BIND, 5-13
CPS networking parameter, 5-48	NAMES.LDAP_CONN_TIMEOUT, 5-13
SQLNET.KERBEROS5_CC_NAME networking	NAMES.LDAP_PERSISTENT_SESSION, 5-13
parameter, 5-50	NAMES.NIS.META_MAP, 5-14
SQLNET.KERBEROS5_CLOCKSKEW	NAMESCTL.TRACE_UNIQUE, 5-24
networking parameter, 5-51	RECV_BUF_SIZE, 5-24
SQLNET.KERBEROS5_CONF networking	SDP.PF_INET_SDP, 5-24
parameter, 5-51	SEC_USER_AUDIT_ACTION_BANNER, 5-25
SQLNET.KERBEROS5_CONF_LOCATON	SEC_USER_UNAUTHORIZED_ACCESS_BANNER,
networking parameter, 5-52	5-25
SQLNET.KERBEROS5_KEYTAB networking	SEND_BUF_SIZE, 5-25
parameter, 5-52	SQLNET.ALLOWED_LOGON_VERSION_CLIENT,
SQLNET.KERBEROS5_REALMS networking	5-29
parameter, 5-53	SQLNET.ALLOWED_LOGON_VERSION_SERVER,
SQLNET.KERBEROS5_REPLAY_CACHE	5-30
networking parameter, 5-53	SQLNET.AUTHENTICATION_KERBEROS5_
sqlnet.ora file	SERVICE, 5-24
ADR diagnostic parameters, 5-90	SQLNET.AUTHENTICATION_SERVICES, 5-35
configuration parameter reference, 5-2	SQLNET.CLIENT_REGISTRATION, 5-37
diagnostic parameters	SQLNET.CLOUD_USER, 5-37
ADR_BASE, 5-91	SQLNET.CLOOD_USER, 5-37 SQLNET.COMPRESSION, 5-39
DIAG_ADR_ENABLED, 5-91	SQLNET.COMPRESSION, 5-39 SQLNET.COMPRESSION_ACCELERATION, 5-39
DIAG ADA ENADLED, J-J1	JULINE I. CONTRESSION ACCELERATION, 3-39

sqlnet.ora file (continued)	sqlnet.ora file (continued)
parameters (continued)	parameters (continued)
SQLNET.COMPRESSION_LEVELS, 5-40	TCP.INVITED_NODES, 5-74
SQLNET.COMPRESSION_THRESHOLD, 5-40	TCP.NODELAY, 5-74
SQLNET.CRYPTO_CHECKSUM_CLIENT, 5-40	TCP.QUEUESIZE, 5-74
SQLNET.CRYPTO_CHECKSUM_SERVER, 5-41	TCP.VALIDNODE_CHECKING, 5-75
SQLNET.CRYPTO_CHECKSUM_TYPE_CLIENT,	TNSPING.TRACE_DIRECTORY, 5-76
5-42	TNSPING.TRACE_LEVEL, 5-76
SQLNET.CRYPTO_CHECKSUM_TYPE_SERVER,	TOKEN_AUTH, 5-76, 6-45
5-42	TOKEN_LOCATION, 5-82, 6-50
SQLNET.DBFW_PUBLIC_KEY, 5-43	USE_DEDICATED_SERVER, 5-87
SQLNET.DOWN_HOSTS_TIMEOUT, 5-44	WALLET_LOCATION, 5-88
SQLNET.ENCRYPTION_SERVER, 5-45	SQLNET.OUTBOUND_CONNECT_TIME
SQLNET.ENCRYPTION_TYPES_CLIENT, 5-45	networking parameter, 5-53
SQLNET.ENCRYPTION_TYPES_SERVER, 5-46	SQLNET.OUTBOUND_CONNECT_TIMEOUT
SQLNET.ENCYRPTION_CLIENT, 5-44	networking parameter, 5-53
SQLNET.EXPIRE_TIME, 5-47	SQLNET.RADIUS_ALTERNATE networking
SQLNET.FALLBACK_AUTHENTICATION, 5-49	parameter, 5-54
SQLNET.IGNORE_ANO_ENCRYPTION_FOR_TCPS	
5-48	networking parameter, 5-55
SQLNET.KERBEROS5_CC_NAME, 5-50	SQLNET.RADIUS_ALTERNATE_RETRIES
SQLNET.KERBEROS5_CLOCKSKEW, 5-51	networking parameter, 5-55
	SQLNET.RADIUS_AUTHENTICATION
SQLNET.KERBEROS5_CONF_LOCATION, 5-52	networking parameter, 5-56
SQLNET.KERBEROS5_KEYTAB, 5-52	SQLNET.RADIUS_AUTHENTICATION_INTERFA
SQLNET.KERBEROS5_REALMS, 5-53	CE networking parameter, 5-56
	SQLNET.RADIUS_AUTHENTICATION_PORT
SQLNET.OUTBOUND_CONNECT_TIME, 5-53	networking parameter, 5-57
SQLNET.OUTBOUND_CONNECT_TIMEOUT, 5-53	
SQLNET.RADIUS_ALTERNATE, 5-54	networking parameter, 5-57
	SQLNET.RADIUS_AUTHENTICATION_TIMEOUT
SQLNET.RADIUS_ALTERNATE_RETRIES, 5-55	networking parameter, 5-58
	SQLNET.RADIUS CHALLENGE RESPONSE
SQLNET.RADIUS_AUTHENTICATION_INTERFACE	
	SQLNET.RADIUS_SECRET networking
SQLNET.RADIUS_AUTHENTICATION_PORT, 5-57	parameter, 5-59
SQLNET.RADIUS_AUTHENTICATION_RETRIES,	
5-57	networking parameter, 5-60
SQLNET.RADIUS_AUTHENTICATION_TIMEOUT,	
5-58	5-60
SQLNET.RADIUS_CHALLENGE_RESPONSE, 5-59	
SQLNET.RADIUS_SECRET, 5-59	5-61
	SQLNET.URI networking parameter, 5-62
·	SQLNET.USE_HTTPS_PROXY networking
SQLNET.SEND_TIMEOUT, 5-61	parameter, 5-63
	SQLNET.WALLET_OVERRIDE, 5-63
	SRC networking parameter, 8-16
	SRV networking parameter, 8-16
	SSL_CERT_REVOCATION networking
SSL_CRL_FILE, 5-65	parameter, 5-64
	SSL_CIPHER_SUITES networking parameter,
SSL_EXTENDED_KEY_USAGE, 5-70	5-66, 7-18, 8-18
	SSL_CLIENT_AUTHENTICATION control
SSL_VERSION, 5-72, 6-44, 7-22, 8-21	parameter, 5-69, 7-20, 8-20
-	SSL_CRL_FILE networking parameter, 5-65
	SSL_CRL_PATH networking parameter, 5-66

SSL_EXTENDED_KEY_USAGE networking	TDM_THREADING_MODE networking
parameter, 5-70	parameter, 8-29
SSL_SERVER_CERT_DN networking parameter,	terminated connection detection
6-42	EXPIRE_TIME parameter, 8-9
SSL_SERVER_DN_MATCH networking	limitations, 5-47, 8-9
parameter, 5-71, 6-43	SQLNET.EXPIRE_TIME parameter, 5-47
SSL_VERSION networking parameter, 5-72,	tnames.ora file
6-44, 7-22, 8-21	parameters
START command	COMPRESSION, 6-60
of Listener Control utility, 1-21	COMPRESSION_LEVELS, 6-61
	tnsnames.ora file
STARTUS command	
STATUS command	parameters, 6-60
of Listener Control utility, 1-23	ADDRESS, 6-7
STOP command	ADDRESS_LIST, 6-9
of Listener Control utility, 1-25	CONNECT_DATA, 6-17
Structural Object Classes, <i>C-1</i>	CONNECT_TIMEOUT, 6-56
SUBSCRIBE_FOR_NODE_DOWN_EVENT_liste	DELAY, 6-18
ner_name control parameter, 7-23	DESCRIPTION, 6-6
SUPER_SHARDING_KEY networking parameter,	DESCRIPTION_LIST, 6-6
6-24	ENABLE, <i>6-10</i>
SUSPEND GATEWAY command, 2-30	FAILOVER, 6-10, 6-11
syntax	FAILOVER_MODE, 6-17
rules for network configuration files, 3-1	GLOBAL_NAME, 6-18
	HS, 6-19
Т	HTTPS_PROXY, 6-7
<u> </u>	HTTPS_PROXY_PORT, 6-8
TAF	IGNORE_ANO_ENCRYPTION_FOR_TCPS,
see Transparent Application Failover (TAF),	6-30
6-17	INSTANCE_NAME, 6-19
TCP.CONNECT_TIMEOUT networking	LOAD_BALANCE, 6-11
parameter, 5-73	METHOD, 6-18
TCP.EXCLUDED_NODES networking parameter,	RDB_DATABASE, 6-21
5-73	RECV BUF SIZE, 6-12
TCP.INVITED_NODES networking parameter,	RECV_TIMEOUT, 6-59
5-74	RETRIES, 6-18
TCP.NODELAY networking parameter, 5-74	RETRY_COUNT, 6-57
TCP.QUEUESIZE networking parameter, 5-74	RETRY DELAY, 6-58
TCP.VALIDNODE_CHECKING networking	SDU, 6-13
parameter, 5-75	SECURITY, 6-41
TCP/IP protocol, parameters for addresses, 4-2	SEND_BUF_SIZE, 6-13
TCP/IP with TLS protocol, parameters for	SERVER, 6-25
addresses, 4-2	SERVICE_NAME, 6-26
tdm file parameters, 8-25	SHARDING_KEY, 6-22
•	SOURCE_ROUTE, 6-14
TDM networking parameter, 8-26	SSL_SERVER_CERT_DN, 6-42
TDM_BIND_THREAD networking parameter,	SSL_SERVER_DN_MATCH, 5-71, 6-43
8-27	SUPER_SHARDING_KEY, 6-24
TDM_DATATYPE_CHECK networking parameter,	TOKEN_AUTH, 5-76, 6-45
8-27	-
TDM_PRCP_MAX_CALL_WAIT_TIME	TOKEN_LOCATION, 5-82, 6-50
networking parameter, 8-28	TRANSACTION, 6-18
TDM_PRCP_MAX_TXN_CALL_WAIT_TIME	TRANSPORT_CONNECT_TIMEOUT, 6-58
networking parameter, 8-28	TYPE, 6-18
TDM_SHARED_THREADS_MAX networking	TYPE_OF_SERVICE, 6-15
parameter, 8-29	TNSPING.TRACE_DIRECTORY networking
TDM_SHARED_THREADS_MIN networking parameter, 8-29	parameter, 5-76

TNSPING.TRACE_LEVEL networking parameter, 5-76	TRACE_TIMESTAMP diagnostic parameter, 8-32 TRACE_TIMESTAMP_CLIENT diagnostic
TRACE command, 1-25	parameter, 5-93
TRACE_DIRECTORY diagnostic parameter, 8-33	TRACE_TIMESTAMP_listener_name diagnostic
TRACE_DIRECTORY_CLIENT diagnostic	parameter, 7-30
parameter, 5-96	TRACE_TIMESTAMP_SERVER diagnostic
TRACE_DIRECTORY_listener_name diagnostic	parameter, 5-93
parameter, 7-31	TRACE_UNIQUE_CLIENT diagnostic parameter,
TRACE_DIRECTORY_SERVER diagnostic	5-100
parameter, 5-96	TRANSACTION networking parameter, 6-18
TRACE_FILE networking parameter, 8-22	Transparent Application Failover (TAF)
TRACE_FILE_CLIENT diagnostic parameter,	parameters, 6-17
5-97	TRANSPORT_CONNECT_TIMEOUT networking
TRACE_FILE_listener_name diagnostic	parameter, 6-58
parameter, 7-32	TYPE networking parameter, 6-18
TRACE_FILE_SERVER diagnostic parameter,	TYPE_OF_SERVICE networking parameter, 6-15
5-97	
TRACE_FILEAGE_CLIENT diagnostic parameter,	U
5-97	<u> </u>
TRACE_FILEAGE_listener_name diagnostic	Unsupported Control Utility Commands, A-2
parameter, 7-32	Unsupported Protocols, A-3
TRACE_FILEAGE_SERVER diagnostic	USE_DEDICATED_SERVER networking
parameter, 5-98	parameter, 5-87
TRACE_FILELEN diagnostic parameter, 8-34	USE_SID_AS_SERVICE_listener_name control
TRACE_FILELEN_CLIENT diagnostic parameter,	parameter, 7-23, 8-22
5-98	·
TRACE_FILELEN_listener_name diagnostic	V
parameter, 7-32	<u>v</u>
TRACE_FILELEN_SERVER diagnostic	VALID_NODE_CHECKING_REGISTRATION_list
parameter, 5-99	ener_name, 7-24
TRACE_FILENO diagnostic parameter, 8-34	VALID_NODE_CHECKING_REGISTRATION_list
TRACE_FILENO_CLIENT diagnostic parameter,	ener_name control parameter, 7-24, 8-23
5-99	VERSION command
TRACE_FILENO_listener_name diagnostic	of Listener Control utility, 1-26
parameter, 7-33	•
TRACE_FILENO_SERVER diagnostic parameter,	W
5-99	VV
TRACE_LEVEL diagnostic parameter, 8-32	WALLET_LOCATION networking parameter,
TRACE_LEVEL_CLIENT diagnostic parameter,	5-88, 7-25, 8-23
5-92	2 20, 7 20, 0 20
TRACE_LEVEL_listener_name, 7-29	
TRACE_LEVEL_SERVER diagnostic parameter,	
5-92	