

ÍNDICE

Contenido

.....	1
ANEXO PAQUETES PREDETERMINADOS	9
Lista de paquetes predeterminados.....	9
DBMS_RANDOM.....	22
Procedimiento INITIALIZE	23
<i>SINTAXIS</i>	23
<i>DESCRIPCIÓN DE LOS PARÁMETROS</i>	23
Función NORMAL.....	23
<i>SINTAXIS</i>	23
<i>DESCRIPCIÓN DEL VALOR DE RETORNO DE LA FUNCIÓN</i>	24
Procedimiento RANDOM	24
<i>SINTAXIS</i>	24
<i>DESCRIPCIÓN DE LOS PARÁMETROS</i>	24

Procedimiento SEED	24
<i>SINTAXIS</i>	24
<i>DESCRIPCIÓN DE LOS PARÁMETROS</i>	25
Función STRING	25
<i>SINTAXIS</i>	25
<i>DESCRIPCIÓN DE LOS PARÁMETROS Y VALORES DE RETORNO</i>	25
Procedimiento TERMINATE	26
<i>SINTAXIS</i>	26
Función VALUE	26
<i>SINTAXIS</i>	26
<i>DESCRIPCIÓN DE LOS PARÁMETROS Y VALORES DE RETORNO</i>	27
DBMS_UTILITY	28
Errores de la sección EXCEPTION	30
Procedimiento ACTIVE_INSTANCES	31
<i>SINTAXIS</i>	31
<i>DESCRIPCIÓN DE LOS PARÁMETROS</i>	31
Procedimiento ANALYZE_DATABASE	32
<i>SINTAXIS</i>	32
<i>DESCRIPCIÓN DE LOS PARÁMETROS</i>	32
<i>ERRORES DE LA SECCIÓN EXCEPTION</i>	33
Procedimiento ANALYZE_PART_OBJECT	33
<i>SINTAXIS</i>	33
<i>DESCRIPCIÓN DE LOS PARÁMETROS</i>	33
Procedimiento ANALYZE_SCHEMA	34
<i>SINTAXIS</i>	35
<i>DESCRIPCIÓN DE LOS PARÁMETROS</i>	35
<i>ERRORES DE LA SECCIÓN EXCEPTION</i>	36
Procedimiento CANONICALIZE	36
<i>SINTAXIS</i>	36
<i>DESCRIPCIÓN DE LOS PARÁMETROS</i>	36
<i>VALORES RETORNADOS POR EL PROCEDIMIENTO</i>	37
<i>RESTRICCIONES DE USO</i>	37
<i>EJEMPLOS DE CONVERSIONES</i>	37
Procedimiento COMMA_TO_TABLE	37
<i>SINTAXIS</i>	38
<i>DESCRIPCIÓN DE LOS PARÁMETROS</i>	38
<i>VALORES RETORNADOS POR EL PROCEDIMIENTO</i>	38
<i>RESTRICCIONES DE USO</i>	39
Procedimiento COMPILE_SCHEMA	39

<i>SINTAXIS</i>	39
<i>DESCRIPCIÓN DE LOS PARÁMETROS</i>	39
<i>ERRORES DE LA SECCIÓN EXCEPTION</i>	40
<i>RESTRICCIONES DE USO</i>	40
Procedimiento CREATE_ALTER_TYPE_ERROR_TABLE	40
<i>SINTAXIS</i>	40
<i>DESCRIPCIÓN DE LOS PARÁMETROS</i>	41
<i>ERRORES DE LA SECCIÓN EXCEPTION</i>	41
Función DATA_BLOCK_ADDRESS_BLOCK	41
<i>SINTAXIS</i>	41
<i>DESCRIPCIÓN DE LOS PARÁMETROS</i>	41
<i>VALORES RETORNADOS POR LA FUNCIÓN</i>	42
<i>RESTRICCIONES DE USO</i>	42
Procedimiento DB_VERSION	42
<i>SINTAXIS</i>	42
<i>DESCRIPCIÓN DE LOS PARÁMETROS</i>	42
Procedimiento EXEC_DDL_STATEMENT	43
<i>SINTAXIS</i>	43
<i>DESCRIPCIÓN DE LOS PARÁMETROS</i>	43
Función FORMAT_CALL_STACK	43
<i>SINTAXIS</i>	43
<i>VALORES RETORNADOS POR LA FUNCIÓN</i>	44
Función FORMAT_ERROR_BACKTRACE	44
<i>SINTAXIS</i>	44
<i>VALORES RETORNADOS POR LA FUNCIÓN</i>	44
Función FORMAT_ERROR_STACK	44
<i>SINTAXIS</i>	44
<i>VALORES RETORNADOS POR LA FUNCIÓN</i>	45
Función GET_CPU_TIME	45
<i>SINTAXIS</i>	45
<i>VALORES RETORNADOS POR LA FUNCIÓN</i>	45
Procedimiento GET_DEPENDENCY	45
<i>SINTAXIS</i>	45
<i>DESCRIPCIÓN DE LOS PARÁMETROS</i>	45
<i>RESTRICCIONES DE USO</i>	46
Función GET_HASH_VALUE	46
<i>SINTAXIS</i>	46
<i>DESCRIPCIÓN DE LOS PARÁMETROS</i>	46
<i>VALORES RETORNADOS POR LA FUNCIÓN</i>	47
Función GET_PARAMETER_VALUE	47

<i>SINTAXIS</i>	47
<i>DESCRIPCIÓN DE LOS PARÁMETROS</i>	47
<i>VALORES RETORNADOS POR LA FUNCIÓN</i>	48
Función GET_TIME	48
<i>SINTAXIS</i>	48
<i>VALORES RETORNADOS POR LA FUNCIÓN</i>	48
Procedimiento INVALIDATE	49
<i>SINTAXIS</i>	49
<i>DESCRIPCIÓN DE LOS PARÁMETROS</i>	49
<i>ERRORES DE LA SECCIÓN EXCEPTION</i>	50
<i>RESTRICCIONES DE USO</i>	50
UTL_MAIL	51
Procedimiento SEND	52
<i>SINTAXIS</i>	52
<i>DESCRIPCIÓN DE LOS PARÁMETROS</i>	52
Procedimiento SEND_ATTACH_RAW	54
<i>SINTAXIS</i>	54
<i>DESCRIPCIÓN DE LOS PARÁMETROS</i>	54
Procedimiento SEND_ATTACH_VARCHAR2	56
<i>SINTAXIS</i>	56
<i>DESCRIPCIÓN DE LOS PARÁMETROS</i>	57
UTL_http	59
<i>EJEMPLO</i>	68
Errores de la sección EXCEPTION	68
Tipo REQ	70
<i>SINTAXIS</i>	70
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL TIPO DE DATO</i>	70
Tipo REQUEST_CONTEXT_TYPE	70
<i>SINTAXIS</i>	71
Tipo RESP	71
<i>SINTAXIS</i>	71
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL TIPO DE DATO</i>	71
Tipos COOKIE y COOKIE_TABLE	72
<i>SINTAXIS</i>	72
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL TIPO DE DATO COOKIE</i>	72
Tipo CONNECTION	73
<i>SINTAXIS</i>	74
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL TIPO DE DATO</i>	74

Procedimiento ADD_COOKIES.....	75
<i>SINTAXIS</i>	75
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO</i>	75
Función BEGIN_REQUEST	75
<i>SINTAXIS</i>	76
<i>DESCRIPCIÓN DE LOS PARÁMETROS DE LA FUNCIÓN</i>	76
Procedimiento CLEAR_COOKIES.....	77
<i>SINTAXIS</i>	77
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO</i>	77
Procedimiento CLOSE_PERSISTENT_CONN	77
<i>SINTAXIS</i>	77
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO</i>	78
Procedimiento CLOSE_PERSISTENT_CONNS	78
<i>SINTAXIS</i>	78
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO</i>	78
<i>EJEMPLOS</i>	79
Función CREATE_REQUEST_CONTEXT.....	79
<i>SINTAXIS</i>	79
<i>DESCRIPCIÓN DE LOS PARÁMETROS DE LA FUNCIÓN</i>	80
<i>EJEMPLO</i>	81
Procedimiento DESTROY_REQUEST_CONTEXT	81
<i>SINTAXIS</i>	81
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO</i>	81
<i>EJEMPLO</i>	82
Procedimiento END_REQUEST	82
<i>SINTAXIS</i>	82
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO</i>	82
Procedimiento END_RESPONSE	83
<i>SINTAXIS</i>	83
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO</i>	83
Procedimiento GET_AUTHENTICATION.....	83
<i>SINTAXIS</i>	83
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO</i>	83
Procedimiento GET_BODY_CHARSET	84
<i>SINTAXIS</i>	84
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO</i>	84
Función GET_COOKIE_COUNT.....	85
<i>SINTAXIS</i>	85
<i>DESCRIPCIÓN DE LOS PARÁMETROS DE LA FUNCIÓN</i>	85
Procedimiento GET_COOKIE_SUPPORT	85

<i>SINTAXIS</i>	85
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO</i>	86
Función GET_COOKIES	86
<i>SINTAXIS</i>	86
<i>DESCRIPCIÓN DE LOS PARÁMETROS DE LA FUNCIÓN</i>	86
Procedimiento GET_DETAILED_EXCP_SUPPORT	87
<i>SINTAXIS</i>	87
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO</i>	87
Función GET_DETAILED_SQLCODE	88
<i>SINTAXIS</i>	88
Función GET_DETAILED_SQLERRM	88
<i>SINTAXIS</i>	88
Procedimiento GET_FOLLOW_REDIRECT	88
<i>SINTAXIS</i>	88
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO</i>	88
Procedimiento GET_HEADER	89
<i>SINTAXIS</i>	89
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO</i>	89
Procedimiento GET_HEADER_BY_NAME	90
<i>SINTAXIS</i>	90
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO</i>	90
Función GET_HEADER_COUNT	91
<i>SINTAXIS</i>	91
<i>DESCRIPCIÓN DE LOS PARÁMETROS DE LA FUNCIÓN</i>	91
Procedimiento GET_PERSISTENT_CONN_SUPPORT	91
<i>SINTAXIS</i>	91
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO</i>	92
Procedimiento GET_PERSISTENT_CONNS	92
<i>SINTAXIS</i>	92
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO</i>	92
Función GET_PERSISTENT_CONN_COUNT	93
<i>SINTAXIS</i>	93
Procedimiento GET_PROXY	93
<i>SINTAXIS</i>	93
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO</i>	93
Función GET_RESPONSE	94
<i>SINTAXIS</i>	94
<i>DESCRIPCIÓN DE LOS PARÁMETROS DE LA FUNCIÓN</i>	94
Procedimiento GET_RESPONSE_ERROR_CHECK	94
<i>SINTAXIS</i>	94

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO	95
Procedimiento GET_TRANSFER_TIMEOUT	95
SINTAXIS	95
DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO	95
Procedimiento READ_LINE	95
SINTAXIS	96
DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO	96
Procedimiento READ_RAW	96
SINTAXIS	97
DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO	97
Procedimiento READ_TEXT	97
SINTAXIS	98
DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO	98
Función REQUEST	98
SINTAXIS	99
DESCRIPCIÓN DE LOS PARÁMETROS DE LA FUNCIÓN	99
EXCEPCIONES QUE PUEDE DEVOLVER LA FUNCIÓN	100
EJEMPLO	100
Función REQUEST_PIECES	100
SINTAXIS	101
DESCRIPCIÓN DE LOS PARÁMETROS DE LA FUNCIÓN	101
EXCEPCIONES QUE PUEDE DEVOLVER LA FUNCIÓN	102
Procedimiento SET_AUTHENTICATION	102
SINTAXIS	102
DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO	102
Procedimiento SET_AUTHENTICATION_FROM_WALLET	103
SINTAXIS	103
DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO	103
Procedimiento SET_BODY_CHARSET	104
SINTAXIS	104
DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO	105
Procedimiento SET_COOKIE_SUPPORT	105
SINTAXIS	105
DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO	106
Procedimiento SET_DETAILED_EXCP_SUPPORT	106
SINTAXIS	107
DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO	107
Procedimiento SET_FOLLOW_REDIRECT	107
SINTAXIS	107
DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO	108

Procedimiento SET_HEADER	108
<i>SINTAXIS</i>	108
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO</i>	108
Procedimiento SET_PERSISTENT_CONN_SUPPORT	109
<i>SINTAXIS</i>	109
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO</i>	109
Procedimiento SET_PROXY	110
<i>SINTAXIS</i>	110
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO</i>	110
Procedimiento SET_RESPONSE_ERROR_CHECK	110
<i>SINTAXIS</i>	111
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO</i>	111
Procedimiento SET_TRANSFER_TIMEOUT	111
<i>SINTAXIS</i>	111
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO</i>	111
Procedimiento SET_WALLET	112
<i>SINTAXIS</i>	112
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO</i>	112
Procedimiento WRITE_LINE	113
<i>SINTAXIS</i>	113
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO</i>	113
Procedimiento WRITE_RAW	113
<i>SINTAXIS</i>	113
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO</i>	114
Procedimiento WRITE_TEXT	114
<i>SINTAXIS</i>	114
<i>DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO</i>	114

ANEXO PAQUETES PREDETERMINADOS

En este capítulo se relacionan TODOS los paquetes que se instalan de forma predeterminada con una base de datos Oracle 12c, y también se desarrollan el resto de los paquetes que más habitualmente se utilizan en la programación contra bases de datos Oracle.

Hay que recordar que para consultar con más detalle cualquiera de los paquetes que se relacionan en este capítulo, pueden acceder a la guía oficial de Oracle *PL/SQL Package and Types Reference* (<http://docs.oracle.com/database/121/ARPLS/toc.htm>).

Lista de paquetes predeterminados

A continuación, se muestra la lista completa de paquete predeterminados de la versión Oracle 12c. Todos los paquetes que se relacionan a continuación pueden ser invocados por cualquier usuario con permisos para ejecutar procedimientos (GRANT RESOURCE), porque son creados con sinónimos públicos.

Nombre paquete	Descripción
APEX_CUSTOM_AUTH	Proporciona una interface para la autenticación y gestión de la sesión.
APEX_APPLICATION	Habilita a los usuarios para obtener las ventajas de uso de las variables globales.
APEX_ITEM	Habilita a los usuarios a crear elementos dinámicos de formulario basados en consultas SQL en vez de crear elementos individuales página a página.
APEX_UTIL	Proporciona utilidades para obtener y configurar el estado de la sesión, obteniendo ficheros, chequeando autorizaciones para usuario, reseteando diferentes estados para usuarios, y también obteniendo y configurando preferencias para usuarios.
CTX_ADM	Te permite administrar servidores y diccionarios de datos.
CTX_CLS	Te permite generar reglas CTXRULE para un conjunto de documentos.

CTX_DDL	Te permite crear y gestionar las preferencias, listas de secciones y grupos requeridos para los índices Text.
CTX_DOC	Te permite solicitar servicios de documentos.
CTX_OUTPUT	Te permite gestionar el log indexado.
CTX_QUERY	Te permite generar consultas, contadores y crear expresiones de consultas almacenadas.
CTX_REPORT	Te permite crear varios informes indexados.
CTX_THES	Te permite gestionar el buscador.
CTX_ULEXER	Para usar con el USER-LEXER.
DBMS_ADDM	Facilita el uso de Advisor.
DBMS_ADVANCED_REWRITE	Permite interceptar sentencias SQL y reemplazarlas por sentencias alternativas.
DBMS_ADVISOR	Parte del SQLAccess Advisor un sistema experto que identifica y te ayuda a resolver los problemas relacionados con la ejecución de sentencias SQL.
DBMS_ALERT	Proporciona soporte para la notificación asíncrona de los eventos de base de datos.
DBMS_APPLICATION_INFO	Te permite registrar un nombre de aplicación con la base de datos para auditarla.
DBMS_APPLY_ADM	Proporciona procedimientos administrativos para arrancar, parar y configurar un proceso de aplicación.
DBMS_AQ	Te permite añadir un mensaje (de un tipo de objeto predefinido) en una consulta.
DBMS_AQADM	Te permite administrar funciones en una consulta para mensajes de tipos de objetos predefinidos.
DBMS_AQELM	Proporciona procedimientos para gestionar la configuración de la notificación asíncrona de Advanced Queuing por e-mail y HTTP.
DBMS_AQIN	Juega un parte fundamental en el acceso de seguridad del interface de Oracle JMS.
DBMS_ASSERT	Proporciona un interface para validar las propiedades de los valores de entrada.
DBMS_AUTO_TASK_ADMIN	Usado por el Administrador de base de datos, junto con el Enterprise Manager permite acceder a los controles AUTOTASK.

DBMS_AUTO_TASK_IMMEDIATE	Consta de un subprograma cuya función es iniciar el optimizador de estadísticas que están retrasadas.
DBMS_AW_STATS	Contiene un subprograma que genera y almacena las estadísticas del optimizador para cubos y dimensiones.
DBMS_CAPTURE_ADM	Describe procedimientos administrativos para arrancar, parar y configurar un proceso de captura. Se usa en STREAMS.
DBMS_CDC_PUBLISH	Identifica nuevos datos que han sido añadidos a, modificados o borrados de, tablas relacionales y publicados los datos cambiados en un formulario que es utilizado por una aplicación.
DBMS_COMPARISON	Proporciona interfaces para comparar objetos de base de datos entre diferentes bases de datos.
DBMS_COMPRESSION	Proporciona un interface para facilitar la elección del nivel correcto de comprensión para una aplicación.
DBMS_CONNECTION_POOL	Proporciona un interface para gestionar Database Resident Connection Pool.
DBMS_CQ_NOTIFICATION	Es parte de un conjunto de características que los clientes usan para recibir notificaciones cuando el conjunto de resultados de una consulta ha cambiado.
DBMS_CRYPTO	Te permite encriptar y descifrar datos almacenados.
DBMS_CSX_ADMIN	Proporciona un interface para personalizar la configuración cuando se transporta un tablespace que contiene datos binarios XML.
DBMS_CUBE	Contiene subprogramas para crear cubos OLAP y dimensiones.
DBMS_CUBE_ADVISE	Contiene subprograma para evaluar las vistas materializadas de cubos.
DBMS_DATA_MINING	Implementa la interface de Oracle Data Mining para su gestión.
DBMS_DATA_MINING_TRANSFORM	Proporciona subrutinas que pueden ser usadas para preparar los datos para Oracle Data Mining.

DBMS_DATAPUMP	Te permite mover todo, o parte de una base de datos entre bases de datos, incluyendo tanto datos como metadatos.
DBMS_DBFS_CONTENT	Proporciona una interface para uno o más Store Providers.
DBMS_DBFS_CONTENT_SPI	Proporciona la especificación API para los proveedores de servicio del paquete DBMS_DBFS_CONTENT.
DBMS_DBFS_HS	Proporcionar a los usuarios la habilidad para usar los servicios de Amazon S3 Web.
DBMS_DBFS_SFS	Proporciona una interface para operar con SFS (SecureFile-based) almacenados descritos en el paquete DBMS_DBFS_CONTENT.
DBMS_DB_VERSION	Especifica el número de versión de Oracle y otra información útil para la compilación basada en la selección de la versión de Oracle.
DBMS_DDL	Proporciona acceso a algunas sentencias SQL DDL desde procedimientos almacenados.
DBMS_DEBUG	Implementa herramientas de debug para las rutinas implementadas en PL/SQL.
DBMS_DEFER	Proporciona una interface de usuario para la replicación transaccional. Requiere tener instalado Distributed Option.
DBMS_DEFER_QUERY	Permite consultar la cola de procesos RPC (Remote Procedure Calls). Requiere tener instalado Distributed Option.
DBMS_DEFER_SYS	Proporciona al administrador del sistema una interface para la replicación transaccional. Requiere tener instalado Distributed Option.
DBMS_DESCRIBE	Describe los argumentos de un procedimiento almacenado.
DBMS_DG	Permite a las aplicaciones la notificación de la base de datos primaria en un entorno de Oracle Data Guard.
DBMS_DIMENSION	Te habilita para verificar las relaciones entre dimensiones y proporcionar una alternativa a Enterprise Manager Dimension Wizard para visualizar la definición de las dimensiones.
DBMS_DISTRIBUTED_TRUST_ADMIN	Mantiene Trusted Database List, que es usado para determinar si un enlace de base

Antolín-Muñoz-Chaparro

Jefe de Proyecto-Sistemas Informáticos

División-III de Aplicaciones de Costes de Personal y Pensiones Públicas

Oficina de Informática Presupuestaria (DIR3: EA0027952)

Intervención General de la Administración del Estado

	de datos de un servidor particular puede ser aceptado.
DBMS_EPG	Implementa la pasarela de PL/SQL embebido que habilita a un explorador Web para invocar procedimientos PL/SQL almacenados a través de un listener HTTP.
DBMS_ERRLOG	Proporciona un procedimiento que te habilita para crear una tabla de errores capturados de modo que las operaciones DML pueden continuar después de ser encontrados en vez de abortarse la ejecución y realizar rollback.
DBMS_EXPFIL	Contiene todos los procedimientos usados para gestionar el conjunto de atributos, conjunto de expresiones, índices, estadísticas optimizadas y privilegios para Expression Filter.
DBMS_FGA	Proporciona funciones de seguridad.
DBMS_FILE_GROUP	Uno de los paquetes que permite gestionar los Streams.
DBMS_FILE_TRANSFER	Te permite copiar un fichero binario dentro de una base de datos o transferir un fichero binario entre bases de datos.
DBMS_FLASHBACK	Te permite crear un FDA (Flashback Data Archive).
DBMS_FLASHBACK_ARCHIVE	Contiene procedimientos para disociar y reasociar un archivo FDA.
DBMS_FREQUENT_ITEMSET	Habilita el conteo de conjuntos de ítems frecuentes.
DBMS_HM	Contiene constantes y declaración de procedimiento para Health Check Management.
DBMS_HPROF	Proporciona una interface para la ejecución de aplicaciones PL/SQL.
DBMS_HS_PARALLEL	Habilita el procesamiento paralelo.
DBMS_HS_PASSTHROUGH	Te permite usar Heterogeneous Services.
DBMS_IOT	Crea una tabla con referencias a las filas de una Index Organized Table.
DBMS_JAVA	Proporciona una interface PL/SQL para acceder a bases de datos con funcionalidad desde Java.

Antolín-Muñoz-Chaparro

Jefe de Proyecto-Sistemas Informáticos

División-III de Aplicaciones de Costes de Personal y Pensiones Públicas

Oficina de Informática-Presupuestaria (DIR3-EA0027952)

Intervención General de la Administración del Estado



DBMS_JOB	Gestiona los trabajos en cola.
DBMS_LDAP	Proporciona funciones y procedimientos para acceder a los datos de los servidores LDAP.
DBMS_LDAP_UTL	Proporciona la utilidad Oracle Extension para LDAP.
DBMS_LIBCACHE	Prepara la librería caché de una instancia Oracle.
DBMS_LOB	Proporciona rutinas de propósito general para operaciones con tipos de datos LOB: BLOB, CLOB y BFILES.
DBMS_LOCK	Te permite gestionar los servicios de Oracle Lock Management.
DBMS_LOGMNR	Proporciona funciones para inicializar y ejecutar el lector Log.
DBMS_LOGMNR_D	Consulta las tablas del diccionario de una base de datos.
DBMS_LOGSTDBY	Describe procedimiento para configurar y gestionar el Standby lógico de un entorno de base de datos.
DBMS_METADATA	Permite a los objetos llamadores recuperar definiciones completas de objetos metada del diccionario.
DBMS_METADATA_DIFF	Contiene una interface para comparar dos documentos de metadatos en formato SXML.
DBMS_MGD_ID_UTL	Proporciona un conjunto de utilidades para subprogramas.
DBMS_MGWADM	Describe los tipos de objeto, métodos, constantes y subprogramas que trabajan con Messaging Gateway. Se usa en Advanced Queuing.
DBMS_MONITOR	Te permite usar PL/SQL para monitorizar.
DBMS_MVIEW	Te permite refrescar Snapshots.
DBMS_NETWORK_ACL_ADMIN	Proporciona la interface para administrar ACL (Access Control List).
DBMS_NETWORK_UTL	Proporciona la interface para administrar ACL (Access Control List).
DBMS_OBFUSCATION_TOOLKIT	Proporciona procedimientos para Data Encryption Standards.
DBMS_ODCI	Devuelve el coste de CPU consumida por una función de usuario.

DBMS_OFFLINE_OG	Proporciona una interface pública para referenciación offline.
DBMS_OLAP	Proporciona procedimientos para gestión de OLAP.
DBMS_OUTLN	Proporciona el interface para procedimientos y funciones asociadas con OUTLN_PKG.
DBMS_OUTPUT	Acumula información en un buffer de modo que pueda ser recuperada posteriormente.
DBMS_PARALLEL_EXECUTE	Habilita al usuario para incrementar la actualización de datos de tablas en paralelo.
DBMS_PCLXUTIL	Para uso con índices particionados.
DBMS_PIPE	Proporciona un servicio de canal DBMS que habilita mensajes para ser enviados entre sesiones.
DBMS_PREDICTIVE_ANALYTICS	Proporciona subrutinas que implementan operaciones sobre Data Mining.
DBMS_PREPROCESSOR	Proporciona una interface para imprimir o recuperar el código fuente de una unidad PL/SQL.
DBMS_PROFILER	Proporciona una API Probe Profiler.
DBMS_PROPAGATION_ADM	Proporciona procedimientos administrativos para configurar la propagación de una consulta.
DBMS_RANDOM	Proporciona un generador de número aleatorios.
DBMS_RECTIFIER_DIFF	Proporciona una interface para detectar y resolver inconsistencias entre dos sitios replicados.
DBMS_REDEFINITION	Te permite realizar una reorganización de tablas online.
DBMS_REFRESH	Te permite crear grupos de Snapshots que pueden ser refrescados juntos.
DBMS_REPAIR	Proporciona procedimiento para reparar datos corruptos.
DBMS_REPCAT	Proporciona rutinas para administrar y actualizar el catálogo de replicación.
DBMS_REPCAT_ADMIN	Te permite crear usuarios con los privilegios necesarios para facilitar la replicación simétrica.

DBMS_REPCAT_INSTANTIATE	Plantillas de desarrollo. Requiere Replication Option.
DBMS_REPCAT_RGT	Plantillas para refrescar grupos. Requiere Replication Option.
DBMS_REPUTIL	Proporciona rutinas para generar tablas, triggers y paquetes para la replicación de tablas.
DBMS_RESCONFIG	Proporciona una interface para operar con Resource Configuration List.
DBMS_RESOURCE_MANAGER	Mantiene planes, grupos de consumo y directivas sobre planes de ejecución.
DBMS_RESOURCE_MANAGER_PRIVS	Mantiene privilegios asociados con los grupos de consumidores de recursos.
DBMS_RESULT_CACHE	Proporciona una interface para operar en Result Cache.
DBMS_RESUMABLE	Te permite suspender operaciones que llevan largo tiempo en ejecución.
DBMS_RLMGR	Contiene varios procedimientos para crear y gestionar reglas para Rules Manager.
DBMS_RLS	Proporciona una interface para administrar niveles de seguridad.
DBMS_ROWID	Proporcionar procedimientos para crear rowids y para interpretar su contenido.
DBMS_RULE	Describe el procedimiento EVALUATE usado en los Streams.
DBMS_RULE_ADM	Describe el interface administrativo para crear y gestionar reglas.
DBMS_SCHEDULER	Proporciona una colección de funciones del gestor de cola de trabajos.
DBMS_SERVER_ALERT	Te permite gestionar alertas.
DBMS_SERVICE	Te permite crear, borrar, activar y desactivar servicios para una instancia individual.
DBMS_SESSION	Proporcionar acceso a la sentencia SQL ALTER SESSION.
DBMS_SHARED_POOL	Te permite mantener objetos en la memoria compartida.
DBMS_SPACE	Proporciona información sobre los segmentos de espacio.

DBMS_SPACE_ADMIN	Proporciona administración sobre tablespaces y segmentos.
DBMS_SPM	Soporta la gestión de planes SQL.
DBMS_SQL	Te permite usar SQL dinámico.
DBMS_SQLDIAG	Proporciona una interface para SQL Diagnosability.
DBMS_SQLPA	Proporciona una interface para implementar el SQL Performance Analyzer.
DBMS_SQLTUNE	Proporciona el interface para sentencias SQL con mejora de rendimiento (tune).
DBMS_STAT_FUNCS	Proporciona funciones estadísticas.
DBMS_STATS	Proporciona un mecanismo a los usuarios para visualizar y modificar las estadísticas.
DBMS_STORAGE_MAP	Comunica con FMON para invocar operaciones de mapas.
DBMS_STREAMS	Describe la interface para convertir SYS.AnyData objetos en LCR objetos.
DBMS_STREAMS_ADMIN	Describe procedimientos administrativos para añadir y borrar reglas simples.
DBMS_STREAMS_ADVISOR_ADM	Proporcionar una interface para obtener información de Oracle Streams.
DBMS_STREAMS_AUTH	Proporciona una interface para la gestión de privilegios sobre Streams.
DBMS_STREAMS_HANDLER_ADM	Proporciona una interface para encolar mensajes.
DBMS_STREAMS_MESSAGING	Proporciona una interface para encolar mensajes.
DBMS_STREAMS_TABLESPACE_ADM	Proporcionar procedimientos administrativos para copiar tablespaces entre bases de datos.
DBMS_TDB	Informa si una base de datos puede ser transportada entre plataformas usando RMAN.
DBMS_TRACE	Proporciona rutinas para arrancar y parar el trazador PL/SQL.
DBMS_TRANSACTION	Proporciona acceso a sentencias de transacción SQL.
DBMS_TRANSFORM	Proporciona una interface para formatear los mensajes en Oracle Advanced Queuing.
DBMS_TTS	Chequeo sobre conjuntos transportables.

DBMS_TYPES	Consta de constantes que representan las built-in de Oracle y tipos definidos por el usuario.
DBMS_UTILITY	Proporciona varias rutinas.
DBMS_WARNING	Proporciona una interface para consultar, modificar y borrar las configuraciones del sistema o la sesión.
DBMS_WM	Describe como usar la interface para Oracle Database Workspace Manager.
DBMS_WORKLOAD_CAPTURE	Configura el sistema Workload Capture.
DBMS_WORKLOAD_REPLAY	Proporciona una interface para reproducir e imprimir un registro del Workload.
DBMS_WORKLOAD_REPOSITORY	Te permite gestionar el Workload Repository.
DBMS_XA	Contiene la interface de aplicaciones para llamar a la interface XA para PL/SQL.
DBMS_XDB	Describe Resource Management.
DBMS_XDB_ADMIN	Proporciona una interface para implementar operaciones de administración de XMLIndex.
DBMS_XDBRESOURCE	Proporciona una interface para operar con recursos de XDB.
DBMS_XDB_VERSION	Describe la versión de interface.
DBMS_XDBT	Describe como un administrador puede crear un índice ConText.
DBMS_XDBZ	Controla el repositorio de Oracle XML DB.
DBMS_XEVENT	Proporciona tipos relacionados con eventos y suporta subprogramas.
DBMS_XMLDOM	Explica el acceso a los objetos XMLType.
DBMS_XMLGEN	Convierte los resultados de una consulta SQL a formato XML.
DBMS_XMLINDEX	Proporciona una interface para implementar índices asíncronos.
DBMS_XMLPARSER	Explica el acceso a los contenidos y estructuras de documentos XML.
DBMS_XMLQUERY	Proporciona funcionalidad en el sentido base de datos a XMLType.
DBMS_XMLSAVE	Proporciona funcionalidad en el sentido XML a tipos de base de datos.
DBMS_XMLSCHEMA	Explica procedimientos para registrar y borrar esquema XML.

DBMS_XMLSTORE	Proporciona las habilidades para almacenar datos XML en tablas relacionales.
DBMS_XMLTRANSLATIONS	Proporciona una interface para realizar traducciones de modo que las cadenas de caracteres puedan ser buscadas o visualizadas en varios lenguajes.
DBMS_XPLAN	Describe como formatear la salida del comando EXPLAIN PLAN.
DBMS_XSLPROCESSOR	Explica el acceso a los contenidos y estructura de documento XML.
DEBUG_EXTPROC	Te permite revisar (debug) procedimientos externos.
HTF	Función Hypertext para generar tags HTML.
HTTP	Procedimientos Hypertext para generar tags HTML.
ORD_DICOM	Soporta la gestión y manipulación de imágenes digitales.
ORD_DICOM_ADMIN	Es utilizado por Oracle Multimedia Digital.
OWA_CACHE	Proporciona una interface que habilita la cache de PL/SQL.
OWA_COOKIE	Proporciona una interface para enviar y recibir cookies http desde exploradores del cliente.
OWA_CUSTOM	Proporciona funciones de Global PLSQL Agent Authorization.
OWA_IMAGE	Proporciona una interface para acceder a las coordenadas en las que el usuario pulsa una imagen.
OWA_OPT_LOCK	Contiene subprogramas con estrategias para prevenir actualizaciones perdidas.
OWA_PATTERN	Proporciona una interface para localizar patrones textuales dentro de cadenas de caracteres.
OWA_SEC	Proporciona una interface para autenticación del cliente.
OWA_TEXT	Contiene subprograma usados por OWA_PATTERN para manipular cadenas de caracteres.
OWA_UTIL	Contiene subprograma de utilidad.

SDO_CS	Proporciona funciones para coordinar transformaciones del sistema.
SDO_CSW_PROCESS	Contiene subprogramas para soportar CSW (Catalog Services for the Web).
SDO_GCDR	Contiene subprogramas de Oracle Spatial.
SDO_GEOM	Proporciona funciones para implementar operaciones relacionadas con GeoRaster.
SDO_GEOR	Contiene funciones y procedimientos de GeoRaster.
SDO_GEOR_ADMIN	Contiene subprogramas para administrar operaciones de GeoRaster.
SDO_GEOR_UTIL	Contiene funciones y procedimientos de utilidad para Spatial GeoRaster.
SDO_LRS	Proporciona funciones para soportar sistemas de referenciación lineal.
SDO_MIGRATE	Proporciona funciones para migrar datos espaciales de versiones anteriores.
SDO_NET	Proporciona funciones y procedimiento para trabajar con datos modelados y enlaces en una red.
SDO_NET_MEM	Contiene funciones y procedimiento para editar y analizar operaciones en redes.
SDO_OLS	Contiene funciones y procedimiento para editar y analizar operaciones en redes.
SDO_PC_PKG	Contiene subprogramas que soportan el uso de nubes de punto en Oracle Spatial.
SDO_SAM	Contiene funciones y procedimientos para analizar datos espaciales.
SDO_TIN_PKG	Contiene subprogramas que soportan el uso de redes trianguladas irregulares (TINs) en Spatial.
SDO_TOPO	Proporciona procedimientos para crear y manejar topologías de Spatial.
SDO_TOPO_MAP	Contiene programa para editar topología de Spatial.
SDO_TUNE	Proporciona funciones para manejo de esquemas de índices espaciales.
SDO_UTIL	Proporciona funciones de utilidad y procedimientos para Oracle Spatial.
SDO_WFS_LOCK	Contiene subprogramas para soportar WFS.

SDO_WFS_PROC	Proporciona funciones de utilidad y procedimientos para Oracle Spatial.
SEM_APIS	Contiene subprogramas para trabajar con el Resource Description Framework (RDF) en OWL (Ontology Language)
SEM_PERF	Contiene subprogramas para examinar y mejorar el RDF.
SEM_RDFCTX	Contiene subprogramas para manejar políticas de extracción y semántica de índices creados para los documentos.
SEM_RDFSA	Contiene subprograma para proporcionar control de acceso a los datos RDF.
UTL_COLL	Habilita programas PL/SQL que usan localizadores de colecciones de datos para consulta y actualización.
UTL_COMPRESS	Proporciona un conjunto de utilidades de compresión de datos.
UTL_DBWS	Proporciona servicios Web de base de datos.
UTL_ENCODE	Proporciona funciones para codificar datos RAW.
UTL_FILE	Habilita tus programas PL/SQL para leer y escribir ficheros de texto.
UTL_HTTP	Habilita llamadas http desde PL/SQL y SQL.
UTL_I18N	Proporciona un conjunto de servicios que ayudan a los desarrolladores a construir aplicaciones multilinguaje.
UTL_INADDR	Proporciona un procedimiento para soportar direccionamiento de internet.
UTL_IDENT	Especifica lo que la base de datos o un cliente PL/SQL está ejecutando.
UTL_LMS	Recupera y formatea mensajes de error en diferentes lenguajes.
UTL_MAIL	Una utilidad para manejar email.
UTL_NLA	Expone un subconjunto de BLAS y LAPACK.
UTL_RAW	Proporcionar funciones SQL para manipular tipos de datos RAW.
UTL_RECOMP	Recompila módulos inválidos de PL/SQL.
UTL_REF	Habilita a un programa PL/SQL para acceder a un objeto proporcionado una referencia al mismo.

UTL_SMTP	Proporcionar funcionalidad PL/SQL para enviar emails.
UTL_SPADV	Proporciona subprogramas para recoger y analizar estadísticas para Oracle Streams.
UTL_TCP	Proporciona funcionalidad PL/SQL para soportar comunicaciones simples basadas en TCP/IP.
UTL_URL	Proporciona mecanismos para caracteres URL.
WPG_DOCLOAD	Proporciona una interface para descargar ficheros de tipo BLOB y BFILE.

DBMS_RANDOM

El paquete DBMS_RANDOM proporciona mecanismos para generar números aleatorios.

Este paquete consta de los siguientes subprogramas:

Nombre Subprograma	Tipo	Descripción
INITIALIZE	Procedimiento	Inicializa el paquete con un valor a partir de otro de referencia.
NORMAL	Función	Devuelve números aleatorios en una distribución normal.
RANDOM	Procedimiento	Genera un número aleatorio.
SEED	Procedimiento	Resetea los valores de referencia utilizados para la generación de números aleatorios.
STRING	Función	Obtiene una cadena aleatoria.
TERMINATE	Procedimiento	Termina el paquete
VALUE	Función	Obtiene un número aleatorio, mayor o igual a 0 y menor que 1, con 28 dígitos a la derecha del punto decimal.

Los subprogramas INITIALIZE, RANDOM y TERMINATE se encuentran descatalogados para la versión de 12c de la base de datos de Oracle. La compañía recomienda no utilizarlos en esta versión. Se mantienen por compatibilidad con las versiones anteriores de las bases de datos Oracle.

DBMS_RANDOM.RANDOM genera números enteros en el rango -2^{31} y 2^{31} .

El paquete DBMS_RANDOM puede ser explícitamente inicializado, pero no necesita ser inicializado antes de la llamada al generador de números aleatorios. Este se inicializará automáticamente con la fecha, identificador de usuario e identificador de proceso.

Procedimiento INITIALIZE

Este procedimiento inicializa el generador de números aleatorios.

Este procedimiento está descatalogado para la versión 12c y aunque está soportado en el paquete y no genera errores de compilación si se usa, Oracle recomienda que no sea utilizado.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
DBMS_RANDOM.INITIALIZE (valor IN BINARY_INTEGER) ;
```

DESCRIPCIÓN DE LOS PARÁMETROS

A continuación se describen cada uno de los parámetros que se usan en el procedimiento INITIALIZE.

valor

Número de referencia usado para la generación del número aleatorio.

Función NORMAL

Esta función devuelve un número aleatorio en una distribución estándar.

SINTAXIS

La sintaxis de esta función es la siguiente:

```
DBMS_RANDOM.NORMAL RETURN NUMBER;
```

DESCRIPCIÓN DEL VALOR DE RETORNO DE LA FUNCIÓN

Esta función devuelve un número aleatorio de tipo NUMBER.

Procedimiento RANDOM

Este procedimiento genera un número aleatorio.

Este procedimiento está descatalogado para la versión 12c y aunque está soportado en el paquete y no genera errores de compilación si se usa, Oracle recomienda que no sea utilizado.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
DBMS_RANDOM.RANDOM RETURN binary_integer;
```

DESCRIPCIÓN DE LOS PARÁMETROS

Este procedimiento devuelve un valor de tipo BINARY_INTEGER correspondiente al número aleatorio, siendo este mayor o igual a -2^{31} y menor que 2^{31} .

Procedimiento SEED

Este procedimiento resetea los valores de referencia utilizados para la generación de números aleatorios.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
DBMS_RANDOM.SEED(p_referencia IN binary_integer);
```


DESCRIPCIÓN DE LOS PARÁMETROS

A continuación se describen cada uno de los parámetros que se usan en el procedimiento SEED.

p_referencia

Número o cadena de referencia usada para la generación del número aleatorio.

La cadena de texto usada como referencia puede superar la longitud de 2.000 caracteres.

Función STRING

Esta función obtiene una cadena de texto aleatoria.

SINTAXIS

La sintaxis de esta función es la siguiente:

```
DBMS_RANDOM.STRING (patron  IN CHAR,
                     longitud IN NUMBER)
                     RETURN VARCHAR2;
```

DESCRIPCIÓN DE LOS PARÁMETROS Y VALORES DE RETORNO

A continuación se describen cada uno de los parámetros que se usan en la función STRING, así como el valor de retorno de la misma.

patrón

Especifica que la cadena devuelta se parece a los siguientes patrones porque los incluye dentro de la misma:

- 'u', 'U'
- 'l', 'L'
- 'a', 'A'
- 'x', 'X'

- 'p', 'P'

longitud

Longitud de la cadena devuelta.

RETURN VARCHAR2

La función devuelve una cadena de tipo VARCHAR2.

Procedimiento TERMINATE

Cuando se quiere finalizar la ejecución del paquete se llama a este procedimiento.

Este procedimiento está descatalogado para la versión 12c y aunque está soportado en el paquete y no genera errores de compilación si se usa, Oracle recomienda que no sea utilizado.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
DBMS_RANDOM.TERMINATE;
```

Función VALUE

Esta función obtiene un número aleatorio mayor o igual a 0 y menor que 1 con 38 dígitos de precisión a la derecha decimal. Como alternativa a esta funcionalidad, también se permite obtener un número aleatorio mayor o igual al límite_inferior y menor que el límite_superior, cuando se le pasen estos valores a la llamada de la función.

SINTAXIS

La sintaxis de esta función es la siguiente:

```
DBMS_RANDOM.VALUE (límite_inferior    IN NUMBER,
                    límite_superior    IN NUMBER)
```

Antolín Muñoz Chaparro

Jefe de Proyecto-Sistemas Informáticos

División-III de Aplicaciones de Costes de Personal y Pensiones Públicas

Oficina de Informática Presupuestaria (DIR3: EAO027952)

Intervención General de la Administración del Estado



RETURN NUMBER;

DESCRIPCIÓN DE LOS PARÁMETROS Y VALORES DE RETORNO

A continuación se describen cada uno de los parámetros que se usan en la función VALUE, así como el valor de retorno de la misma.

límite_inferior

Especifica el número más pequeño que se puede obtener como aleatorio. El número aleatorio puede ser igual al límite_inferior.

límite_superior

Especifica el número más grande que se puede obtener como aleatorio. El número aleatorio tiene que ser menor al límite_superior.

RETURN NUMBER

La función devuelve un número de tipo NUMBER.

DBMS_UTILITY

El paquete DBMS_UTILITY ya existía en versiones anteriores de la base de datos, pero en la versión 12c ha incorporado un nuevo subprograma: EXPAND_SQL_TEXT.

Para que se puedan utilizar las llamadas a este paquete es necesario que el usuario de base de datos que lo invoque posea los privilegios NAME_RESOLVE y COMPILE_SCHEMA.

Este paquete no se le puede invocar desde el usuario SYS.

Este paquete consta de los siguientes subprogramas:

Nombre Subprograma	Tipo	Descripción
ACTIVE_INSTANCES	Procedimiento	Devuelve la instancia activa.
ANALYZE_DATABASE	Procedimiento	Analiza todas las tablas, clusters e índices de la B.D.
ANALYZE_PART_OBJECT	Procedimiento	Analiza las tablas e índices que se indiquen.
ANALYZE_SCHEMA	Procedimiento	Analiza todas las tablas, clusters e índices en un esquema dado.
CANONICALIZE	Procedimiento	Pone en forma canónica un texto.
COMMA_TO_TABLE	Procedimiento	Convierte una lista de nombres delimitados por comas en una tabla de nombres PL/SQL.
COMPILE_SCHEMA	Procedimiento	Compila todos los procedimientos, funciones, paquetes, vistas y triggers en un esquema dado.
CREATE_ALTER_TYPE_ERROR_TABLE	Procedimiento	Crea una tabla de errores utilizada en la cláusula EXCEPTION de la sentencia ALTER TYPE.
CURRENT_INSTANCE	Función	Devuelve el número de instancia conectada actualmente.
DATA_BLOCK_ADDRESS_BLOCK	Función	Obtiene la parte del número de bloque de una dirección de bloque de dato.
DATE_BLOCK_ADDRESS_FILE	Función	Obtiene la parte del número de fichero de una dirección de bloque de datos.

DB_VERSION	Procedimiento	Devuelve la información de la versión de B.D.
EXEC_DDL_STATEMENT	Procedimiento	Ejecuta una sentencia DDL in PARSE_STRING
EXPAND_SQL_TEXT	Procedimiento	Reemplaza recursivamente cualquier referencia a una vista en una consulta de entrada SQL con la correspondiente subconsulta de vista.
FORMAT_CALL_STACK	Función	Formatea la pila de llamadas actual.
FORMAT_ERROR_BACKTRACE	Función	Formatea el "punto de retroceso (backtrace)" desde el punto del error actual hasta el manejador de la excepción donde se ha capturado el error.
FORMAT_ERROR_STACK	Función	Formatea la pila de error actual.
GET_CPU_TIME	Función	Devuelve el tiempo actual de la CPU en centésimas de segundo.
GET_DEPENDENCY	Procedimiento	Muestra las dependencias de un objeto que se le pase.
GET_ENDIANNES	Función	Obtiene el "infinito (endianness) de una plataforma de base de datos.
GET_HASH_VALUE	Función	Calcula un valor hash de un string dado
GET_PARAMETER_VALUE	Función	Obtiene el valor del parámetro init.ora que se especifique.
GET_SQL_HASH	Función	Calcula el valor hash de un string dado, usando un algoritmo MD5.
GET_TIME	Función	Devuelve la hora actual en centésimas de segundo.
GET_TZ_TRANSITIONS	Procedimiento	Devuelve la zona horaria para el valor REGIONID del fichero TIMEZONE.DAT
INVALIDATE	Procedimiento	Invalida un objeto de base de datos y opcionalmente modifica parámetros de configuración de su compilador PL/SQL.
IS_BIT_SET	Función	Comprueba la configuración de BIT para el BIT dado en un valor RAW proporcionado.
IS_CLUSTER_DATABASE	Función	Determina si la base de datos está funcionando en modo cluster de b.d.

Antolín-Muñoz-Chaparro

Jefe de Proyecto-Sistemas Informáticos

División-III de Aplicaciones de Costes de Personal y Pensiones Públicas

Oficina de Informática-Presupuestaria (DIR3-EA0027952)

Intervención General de la Administración del Estado



MAKE_DATA_BLOCK_ADDRESS	Función	Crea una dirección de bloque de datos, proporcionando un número de fichero y un número de bloque.
NAME_RESOLVE	Procedimiento	Resuelve el nombre dado.
NAME_TOKENIZE	Procedimiento	Llama al "parser" para analizar un nombre dado.
OLD_CURRENT_SCHEMA	Función	Devuelve el valor de la sesión de SYS_CONTEXT ('USERENV','CURRENT_SCHEMA')
OLD_CURRENT_USER	Función	Devuelve el valor de la sesión de SYS_CONTEXT ('USERENV','CURRENT_USER')
PORT_STRING	Función	Devuelve un string que identifica de forma única la versión de Oracle y el sistema operativo.
SQLID_TO_SQLHASH	Función	Convierte un ID SQL en un valor hash.
TABLE_TO_COMMA	Procedimiento	Convierte una tabla PL/SQL de nombres en una lista de nombres separada por comas.
VALIDATE	Procedimiento	Comprueba si un objeto de base de datos es válido o no.
WAIT_ON_PENDING_DML	Función	Espera hasta que todas las transacciones de las tablas especificadas (distintas a la sesión del llamante de la función WAIT_ON...) hayan finalizado y obtiene el valor SCN en cada instante tanto si se hace commit como rollback.

Errores de la sección EXCEPTION

Los subprogramas de DBMS_UTILITY pueden generar alguno de los errores que se especifican en la siguiente tabla:

Error	Descripción
INV_NOT_EXISTS_OR_NO_PRIV	Se produce por el subprograma INVALIDATE cuando el argumento OBJECT_ID es NULL o inválido, o cuando la llamada no tiene

	privilegios CREATE en el objeto que está siendo invalidado.
INV_MALFORMED_SETTINGS	Se produce por el subprograma INVALIDATE si una configuración del compilador se especifica más de una vez en el parámetro P_PLSQL_OBJECT_SETTINGS
INV_RESTRICTED_OBJETC	Se produce por el subprograma INVALIDATE cuando diferentes combinaciones de condiciones relacionados con el parámetro P_OBJECT_ID son contrapuestos.

Procedimiento ACTIVE_INSTANCES

Este procedimiento devuelve la instancia activa.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
DBMS_UTILITY.ACTIVE_INSTANCES (
    instance_table OUT INSTANCE_TABLE,
    instance_count OUT NUMBER);
```

DESCRIPCIÓN DE LOS PARÁMETROS

A continuación se describen cada uno de los parámetros que se usan en el procedimiento.

instance_table

Contiene una lista de números y nombres de instancias activas. Cuando ninguna instancia está levantada, la lista se devuelve vacía.

instance_count

Número de instancias activas.

Procedimiento ANALYZE_DATABASE

Este procedimiento está obsoleto desde la versión de Oracle 10g, pero se mantiene por compatibilidad con las versiones anteriores.

Este procedimiento analiza todas las tablas, clusters e índices de la base de datos. Use este procedimiento para recoger estadísticas no optimizadas. Para estadísticas optimizadas debe utilizar el procedimiento DBMS_STATS.GATHER_DATABASE_STATS.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
DBMS_UTILITY.ANALYZE_DATABASE (
    method          VARCHAR2,
    estimate_rows    NUMBER      DEFAULT NULL,
    estimate_percent NUMBER      DEFAULT NULL,
    method_opt       VARCHAR2    DEFAULT NULL);
```

DESCRIPCIÓN DE LOS PARÁMETROS

A continuación se describen cada uno de los parámetros que se usan en el procedimiento .

method

Puede ser uno de los siguientes valores: ESTIMATE, COMPUTE o DELETE. Si se indica el valor ESTIMATE entonces los parámetros *estimate_rows* o *estimate_percent* no pueden ser cero.

estimate_rows

Número de filas estimadas.

estimate_percent

Porcentaje de filas estimadas. Si se especifica un valor para este parámetro, entonces el valor del parámetro *estimate_rows* será ignorado.

method_opt

Opciones asociadas al método con el siguiente formato:

```
[FOR TABLE]
[FOR ALL [INDEXED] COLUMNS] [SIZE n]
[FOR ALL INDEXES]
```

ERRORES DE LA SECCIÓN EXCEPTION

Este procedimiento puede provocar las siguientes excepciones:

Error	Descripción
ORA-20000	Insuficientes privilegios para algunos objetos de esta base de datos.

Procedimiento ANALYZE_PART_OBJECT

Este procedimiento es equivalente a ejecutar la siguiente sentencia:

```
ANALYZE TABLE|INDEX [<schema.><object_name> PARTITION
<pname> [<command_opt>] [<sample_clause>]
```

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
DBMS_UTILITY.ANALYZE_PART_OBJECT (
    schema           IN VARCHAR2      DEFAULT NULL,
    object_name      IN VARCHAR2      DEFAULT NULL,
    object_type      IN CHAR          DEFAULT 'T',
    command_type     IN CHAR          DEFAULT 'E',
    command_opt      IN VARCHAR2      DEFAULT NULL,
    sample_clause    IN VARCHAR2      DEFAULT 'sample
5 percent');
```

DESCRIPCIÓN DE LOS PARÁMETROS

A continuación se describen cada uno de los parámetros que se usan en el procedimiento .

Antolín-Muñoz-Chaparro
 Jefe-de-Proyecto-Sistemas-Informáticos
 División-III-de-Aplicaciones-de-Costes-de-Personal-y-Pensiones-Públicas
 Oficina-de-Informática-Presupuestaria-(DIR3-EA0027952)
 Intervención-General-de-la-Administración-del-Estado



schema

Nombre del esquema asociado al parámetro *object_name*.

object_name

Nombre del objeto que se analizará y que deberá estar particionado.

object_type

Tipo de objeto. Debe ser una T (tabla) o I (índice).

command_type

Debe tener el valor V (estructura validada).

command_opt

Otras opciones para el tipo de comando. Para 'C' y 'E' puede ser 'FOR' para tabla, 'FOR' para todos los índices 'LOCAL', 'FOR' para todas las columnas o combinación de las opciones 'FOR' de análisis de estadísticas de tabla. Para 'V' puede ser 'CASCADE' cuando el *object_type* es 'T'.

sample_clause

Este parámetro se usa cuando el parámetro *command_type* es 'E'.

Procedimiento ANALYZE_SCHEMA

Este procedimiento está obsoleto desde la versión de Oracle 10g, pero se mantiene por compatibilidad con las versiones anteriores.

Este procedimiento ejecuta el comando ANALYZE en todas las tablas, clusters e índices de un esquema dado. Use este procedimiento para recoger estadísticas no optimizadas. Para estadísticas optimizadas debe utilizar el procedimiento DBMS_STATS.GATHER_SCHEMA_STATS.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
DBMS_UTILITY.ANALYZE_SCHEMA (
    schema          VARCHAR2,
    method          VARCHAR2,
    estimate_rows   NUMBER    DEFAULT NULL,
    estimate_percent NUMBER    DEFAULT NULL,
    method_opt      VARCHAR2  DEFAULT NULL);
```

DESCRIPCIÓN DE LOS PARÁMETROS

A continuación se describen cada uno de los parámetros que se usan en el procedimiento .

schema

Nombre del esquema.

method

Puede ser uno de los siguientes valores: ESTIMATE, COMPUTE o DELETE. Si se indica el valor ESTIMATE entonces los parámetros *estimate_rows* o *estimate_percent* no pueden ser cero.

estimate_rows

Número de filas estimadas.

estimate_percent

Porcentaje de filas estimadas. Si se especifica un valor para este parámetro, entonces el valor del parámetro *estimate_rows* será ignorado.

method_opt

Opciones asociadas al método con el siguiente formato:

[FOR TABLE]

```
[FOR ALL [INDEXED] COLUMNS] [SIZE n]
[FOR ALL INDEXES]
```

ERRORES DE LA SECCIÓN EXCEPTION

Este procedimiento puede provocar las siguientes excepciones:

Error	Descripción
ORA-20000	Insuficientes privilegios para algún objeto de este esquema.

Procedimiento CANONICALIZE

Este procedimiento pone en forma canónica un texto. Este procedimiento maneja una palabra clave o reservada (tal como 'table'), y elimina cualquier espacio en blanco del identificador, de forma que ' table ' se convierte en TABLE.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
DBMS_UTILITY.CANONICALIZE (
    name          IN   VARCHAR2,
    canon_name    OUT  VARCHAR2,
    canon_len     IN   BINARY_INTEGER);
```

DESCRIPCIÓN DE LOS PARÁMETROS

A continuación se describen cada uno de los parámetros que se usan en el procedimiento .

name

El texto que se va a poner en forma canónica.

canon_name

El texto canonizado.

canon_len

La longitud del texto a canonizar en bytes .

VALORES RETORNADOS POR EL PROCEDIMIENTO

Este procedimiento devuelve los primeros bytes del parámetro *canon_len* en el parámetro *canon_name*:

RESTRICCIONES DE USO

- Si el parámetro *name* es nulo, entonces el parámetro *canon_name* se convertirá en nulo.
- Si el parámetro *name* no es un texto con puntos, y comienza y termina con una doble comilla, entonces se eliminan las comillas.
- Si el parámetro *name* es un texto con puntos (como por ejemplo a."b".c), por cada componente entre los puntos, en el caso de que el componente comience y termine con dobles comillas, entonces no se realizarán transformaciones del componente.
- Cualquier otro carácter después de a.[b]* será ignorado.
- Este procedimiento no maneja casos como 'A B.'

EJEMPLOS DE CONVERSIONES

- a se convierte en A
- "a" se convierte en a
- "a" . b se convierte en "a" . "B"
- "a" . b , c . f se convierte en "a" . "B" ingnorándose " , c . f"

Procedimiento COMMA_TO_TABLE

Este procedimiento convierte una lista de nombres delimitados por comas en una tabla de nombres PL/SQL. De las dos sintaxis que se indican, la segunda soporta nombres de atributos completamente calificados.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
DBMS_UTILITY.COMMA_TO_TABLE (
    list           IN    VARCHAR2,
    tablen         OUT   BINARY_INTEGER,
    tab            IN    UNCL_ARRAY);
```

```
DBMS_UTILITY.COMMA_TO_TABLE (
    list           IN    VARCHAR2,
    tablen         OUT   BINARY_INTEGER,
    tab            IN    LNAME_ARRAY);
```

DESCRIPCIÓN DE LOS PARÁMETROS

A continuación se describen cada uno de los parámetros que se usan en el procedimiento .

list

Una lista de tablas separadas por comas.

tablen

Número de tablas en la tabla PL/SQL.

tab

Tabla PL/SQL que contiene la lista de nombres de tabla.

VALORES RETORNADOS POR EL PROCEDIMIENTO

Este procedimiento devuelve una tabla PL/SQL con valores 1..n y n+1 es NULL.

RESTRICCIONES DE USO

- El parámetro *list* debe ser una lista NO vacía de elementos delimitados por comas. Cualquier otra lista que no esté delimitada por comas será rechazada. Las comas que se introduzcan con dobles comillas no se tendrán en cuenta.
- Las entradas de la lista delimitada por comas, no puede incluir caracteres multibyte como los guiones.

Procedimiento COMPILE_SCHEMA

Este procedimiento compila todos los procedimientos, funciones, paquetes y triggers en un esquema especificado.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
DBMS_UTILITY.COMPILE_SCHEMA (  
    schema            VARCHAR2,  
    compile_all        BOOLEAN DEFAULT TRUE,  
    reuse_settings     BOOLEAN DEFAULT FALSE);
```

DESCRIPCIÓN DE LOS PARÁMETROS

A continuación se describen cada uno de los parámetros que se usan en el procedimiento .

schema

Nombre del esquema.

compile_all

Si tiene el valor TRUE, entonces compilará todo dentro del esquema independientemente de que los objetos estén válidos.

Si tiene el valor FALSE, compilará únicamente los objetos inválidos.

reuse_settings

Indica si se debería reutilizar la configuración de la sesión en los objetos o si por lo contrario se debería adoptar la configuración actual de la sesión.

ERRORES DE LA SECCIÓN EXCEPTION

Este procedimiento puede provocar las siguientes excepciones:

Error	Descripción
ORA-20000	Insuficientes privilegios para algún objeto de este esquema.
ORA-20001	No se puede recompilar objetos del usuario SYS
ORA-20002	Se excedió el máximo de iteraciones. Algunos objetos puede que no hayan sido recompilados.

RESTRICCIONES DE USO

Después de ejecutar este procedimiento, se debería consultar la vista ALL_OBJECTS para ver los objetos que están en estado INVALID (se han compilado con errores), utilizando la instrucción siguiente:

```
SELECT * FROM ALL_OBJECTS WHERE STATUS = 'INVALID';
```

Para ver los errores asociados a los objetos INVALID se puede utilizar el siguiente comando:

```
SHOW ERRORS <type> <schema>.<name>
```

Procedimiento CREATE_ALTER_TYPE_ERROR_TABLE

Este procedimiento crea una tabla de errores utilizada en la cláusula EXCEPTION de la sentencia ALTER TYPE.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
DBMS_UTILITY.CREATE_ALTER_TYPE_ERROR_TABLE (
```



```

schema_name      IN VARCHAR2,
table_name       IN VARCHAR2);

```

DESCRIPCIÓN DE LOS PARÁMETROS

A continuación se describen cada uno de los parámetros que se usan en el procedimiento .

schema_name

Nombre del esquema.

table_name

Nombre de la tabla creada.

ERRORES DE LA SECCIÓN EXCEPTION

Se producirá un error si la tabla ya existe.

Función DATA_BLOCK_ADDRESS_BLOCK

Este procedimiento obtiene la parte del número de bloque de una dirección de bloque de dato.

SINTAXIS

La sintaxis de esta función es la siguiente:

```

DBMS_UTILITY.DATA_BLOCK_ADDRESS_BLOCK (
    dba          NUMBER)
RETURN NUMBER;

```

DESCRIPCIÓN DE LOS PARÁMETROS

A continuación se describen cada uno de los parámetros que se usan en el procedimiento .

dba

Dirección del bloque de datos.

VALORES RETORNADOS POR LA FUNCIÓN

Esta función devuelve el fichero que contiene al bloque.

RESTRICCIONES DE USO

No se debería utilizar esta función para bloques de datos que pertenecen a Tablespaces muy voluminosos.

Procedimiento DB_VERSION

Este procedimiento devuelve la información de la versión de base de datos.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
DBMS_UTILITY.DB_VERSION (
    version          OUT VARCHAR2,
    compatibility    OUT VARCHAR2);
```

DESCRIPCIÓN DE LOS PARÁMETROS

A continuación se describen cada uno de los parámetros que se usan en el procedimiento .

version

Un texto que representa la versión del software interno de la base de datos (por ejemplo 7.1.0.0.0). La longitud del texto es variable y se determina por la versión de la base de datos.

compatibility

La configuración de compatibilidad de la base de datos, determinada por el parámetro COMPATIBLE del fichero INIT.ORA. Si el parámetro no está especificado en el fichero INIT.ORA entonces se devolverá nulo.

Procedimiento EXEC_DDL_STATEMENT

Este procedimiento ejecuta la sentencia DDL que se incluya en el parámetro PARSE_STRING.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
DBMS_UTILITY.EXEC_DDL_STATEMENT (
    parse_string      IN VARCHAR2);
```

DESCRIPCIÓN DE LOS PARÁMETROS

A continuación se describen cada uno de los parámetros que se usan en el procedimiento .

parse_string

La sentencia DDL que se quiere ejecutar.

Función FORMAT_CALL_STACK

Esta función formatea la pila de llamadas actual. Se puede utilizar sobre cualquier procedimiento almacenado o trigger que acceda a la pila de llamadas, y puede ser muy útil para el proceso de debug (rastreo de errores).

SINTAXIS

La sintaxis de esta función es la siguiente:

```
DBMS_UTILITY.FORMAT_CALL_STACK
```

```
RETURN VARCHAR2;
```

VALORES RETORNADOS POR LA FUNCIÓN

Esta función devuelve la pila de llamadas hasta los 2000 bytes.

Función `FORMAT_ERROR_BACKTRACE`

Esta función formatea el "punto de retroceso (backtrace)" desde el punto del error actual hasta el manejador de la excepción donde se ha capturado el error. La salida es similar a la función `SQLERRM`.

SINTAXIS

La sintaxis de esta función es la siguiente:

```
DBMS_UTILITY.FORMAT_ERROR_BACKTRACE  
RETURN VARCHAR2;
```

VALORES RETORNADOS POR LA FUNCIÓN

Esta función devuelve un string. Devolverá nulo si ningún error está siendo manejado en ese momento.

Función `FORMAT_ERROR_STACK`

Esta función formatea la pila de error actual. Se puede utilizar en manejadores de excepciones para observar la pila de error al completo.

SINTAXIS

La sintaxis de esta función es la siguiente:

```
DBMS_UTILITY.FORMAT_ERROR_STACK  
RETURN VARCHAR2;
```

VALORES RETORNADOS POR LA FUNCIÓN

Esta función devuelve la pila de error hasta los 2000 bytes.

Función GET_CPU_TIME

Esta función devuelve el tiempo actual en centésimas de segundo.

SINTAXIS

La sintaxis de esta función es la siguiente:

```
DBMS_UTILITY.GET_CPU_TIME
RETURN NUMBER;
```

VALORES RETORNADOS POR LA FUNCIÓN

Esta función devuelve un número correspondiente a la centésima de segundo tomada desde alguna época arbitraria.

Procedimiento GET_DEPENDENCY

Este procedimiento muestra las dependencias sobre el objeto que se le pasa.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
DBMS_UTILITY.GET_DEPENDENCY (
    type           IN VARCHAR2
    schema         IN VARCHAR2,
    name           IN VARCHAR2);
```

DESCRIPCIÓN DE LOS PARÁMETROS

A continuación se describen cada uno de los parámetros que se usan en el procedimiento .

type

El tipo de objeto, por ejemplo si el objeto es una tabla dada, el tipo será 'TABLE'.

schema

Nombre del esquema al que pertenece el objeto.

name

El nombre del objeto.

RESTRICCIONES DE USO

Este procedimiento utilizar el paquete DBMS_OUTPUT para visualizar los resultados, por lo que es necesario declarar previamente la instrucción:

```
SET SERVEROUTPUT ON;
```

Función GET_HASH_VALUE

Esta función calcula el valor hash de un string dado.

SINTAXIS

La sintaxis de esta función es la siguiente:

```
DBMS_UTILITY.GET_HASH_VALUE (
    name          VARCHAR2
    base          NUMBER,
    hash_size     NUMBER)
RETURN NUMBER;
```

DESCRIPCIÓN DE LOS PARÁMETROS

A continuación se describen cada uno de los parámetros que se usan en la función.

name

String para el que se va a calcular el valor hash.

base

Valor base actual para devolver el valor hash y por el que se comienza.

hash_size

Tamaño deseado de la tabla hash.

VALORES RETORNADOS POR LA FUNCIÓN

Esta función devuelve un valor hash basado en el texto de entrada. Por ejemplo, para obtener el valor hash de un string donde el valor hash debería estar entre 1000 y 3047, usando 1000 como valor base y 2048 como el valor del parámetro HASH_SIZE. Usando una potencia de 2 para el parámetro HASH_SIZE, funciona mejor.

Función GET_PARAMETER_VALUE

Esta función obtiene del fichero INIT.ORA el valor del parámetro correspondiente que se especifique en la función.

SINTAXIS

La sintaxis de esta función es la siguiente:

```
DBMS_UTILITY.GET_PARAMETER_VALUE (
    parnam          IN          VARCHAR2
    intval          IN OUT      BINARY_INTEGER,
    strval          IN OUT      VARCHAR2)
RETURN BINARY_INTEGER;
```

DESCRIPCIÓN DE LOS PARÁMETROS

A continuación se describen cada uno de los parámetros que se usan en la función.

parnam

Nombre del parámetro.

intval

Valor de un parámetro con tipo INTEGER o la longitud de un valor correspondiente a un parámetro de tipo carácter.

strval

Valor de un parámetro de tipo carácter

VALORES RETORNADOS POR LA FUNCIÓN

Esta función devuelve dos posibles valores:

- 0 si el parámetro es de tipo INTEGER o BOOLEAN.
- 1 si el parámetro es del tipo carácter o fichero.

Función GET_TIME

Esta función devuelve la hora actual en centésimas de segundo. Este subprograma se suele utilizar principalmente para determinar el tiempo transcurrido. Se invoca dos veces; al comienzo y al final de algún proceso y entonces se determina el tiempo transcurrido entre ambas llamadas.

SINTAXIS

La sintaxis de esta función es la siguiente:

```
DBMS_UTILITY.GET_TIME (
    RETURN NUMBER;
```

VALORES RETORNADOS POR LA FUNCIÓN

Esta función devuelve un número en centésimas de segundo.

Antolín Muñoz-Chaparro

Jefe de Proyecto-Sistemas Informáticos

División III de Aplicaciones de Costes de Personal y Pensiones Públicas

Oficina de Informática Presupuestaria (DIR3: EAO02/952)

Intervención General de la Administración del Estado



Procedimiento INVALIDATE

Este procedimiento invalida un objeto de base de datos y opcionalmente modifica la configuración de parámetros del compilador PL/SQL. También invalida cualquier objeto que (directa o indirectamente) dependa de objeto inicial invalidado.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
DBMS_UTILITY.INVALIDATE (
    p_objetc_id          NUMBER,
    p_plsql_object_settings VARCHAR2 DEFAULT
    NULL,
    p_option_flags       PLS_INTEGER DEFAULT
    0);
```

DESCRIPCIÓN DE LOS PARÁMETROS

A continuación se describen cada uno de los parámetros que se usan en el procedimiento .

schema

Nombre del esquema.

compile_all

Si tiene el valor TRUE, entonces compilará todo dentro del esquema independientemente de que los objetos estén válidos.

Si tiene el valor FALSE, compilará únicamente los objetos inválidos.

reuse_settings

Indica si se debería reutilizar la configuración de la sesión en los objetos o si por lo contrario se debería adoptar la configuración actual de la sesión.

ERRORES DE LA SECCIÓN EXCEPTION

Este procedimiento puede provocar las siguientes excepciones:

Error	Descripción
ORA-20000	Insuficientes privilegios para algún objeto de este esquema.
ORA-20001	No se puede recompilar objetos del usuario SYS
ORA-20002	Se excedió el máximo de iteraciones. Algunos objetos puede que no hayan sido recompilados.

RESTRICCIONES DE USO

Después de ejecutar este procedimiento, se debería consultar la vista ALL_OBJECTS para ver los objetos que están en estado INVALID (se han compilado con errores), utilizando la instrucción siguiente:

```
SELECT * FROM ALL_OBJECTS WHERE STATUS = 'INVALID';
```

Para ver los errores asociados a los objetos INVALID se puede utilizar el siguiente comando:

```
SHOW ERRORS <type> <schema>.<name>
```

UTL_MAIL

El paquete UTL_MAIL no está instalado por debido a los requerimientos de configuración de SMTP_OUT_SERVER y la seguridad que implica. En la instalación de UTL_MAIL deberían tomar precauciones y asignar las operaciones adecuadas para que el puerto que se defina en SMTP_OUT_SERVER no sea utilizado por otras transmisiones.

Para instalar UTL_MAIL hay que ejecutar el siguiente código:

```
sqlplus sys/<password>
```

```
SQL> @$ORACLE_HOME/rdbms/admin/utlmail.sql
```

```
SQL> @$ORACLE_HOME/rdbms/admin/prvtmail.plb
```

Para definir los parámetros de SMTP_OUT_SERVER hay que abrir el fichero de inicialización de la base de datos: INIT.ORA.

Este paquete consta de los siguientes subprogramas:

Nombre Subprograma	Tipo	Descripción
SEND	Procedimiento	Empaqueta un mensaje de e-mail en el formato apropiado, localiza la información SMTP y envía el mensaje al servidor SMTP para que lo remita a los destinatarios.
SEND_ATTACH_RAW	Procedimiento	Representa al procedimiento SEND sobrecargado para adjuntar contenido RAW.
SEND_ATTACH_VARCHAR2	Procedimiento	Representa al procedimiento SEND sobrecargado para adjuntar contenido VARCHAR2.

Procedimiento SEND

Este procedimiento empaqueta un mensaje de e-mail en el formato apropiado, localiza la información SMTP, y remite el mensaje al servidor SMTP para que a su vez le de traslado a los receptores del mensaje.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_MAIL.SEND (
  sender      IN VARCHAR2 CHARACTER SET ANY_CS,
  recipients  IN VARCHAR2 CHARACTER SET ANY_CS,
  cc          IN VARCHAR2 CHARACTER SET ANY_CS
              DEFAULT NULL,
  bcc         IN VARCHAR2 CHARACTER SET ANY_CS
              DEFAULT NULL,
  subject     IN VARCHAR2 CHARACTER SET ANY_CS
              DEFAULT NULL,
  message     IN VARCHAR2 CHARACTER SET ANY_CS
              DEFAULT NULL,
  mime_type   IN VARCHAR2 DEFAULT CHARACTER SET
              ANY_CS DEFAULT 'text/plain;
              character=us-ascii',
  priority    IN PLS_INTEGER DEFAULT 3,
  replyto     IN VARCHAR2 CHARACTER SET ANY_CS
              DEFAULT NULL);
```

DESCRIPCIÓN DE LOS PARÁMETROS

A continuación se describen cada uno de los parámetros que se usan en el procedimiento SEND.

sender

Dirección email del que envía el mensaje.

recipients

Dirección email del receptor o receptores del mensaje (en caso de varios, han de ir separados por comas).

cc

Dirección email del receptor o receptores a los que se les envía una copia del mensaje. Al igual que antes, si son varios se separarán con comas. El valor por defecto es NULL.

bcc

Dirección email del receptor o receptores a los que se les envía una copia oculta del mensaje. Al igual que antes, si son varios se separarán con comas. El valor por defecto es NULL.

subject

Cadena de texto para identificar el asunto que se incluye en el mensaje.

message

Texto correspondiente al cuerpo del mensaje.

mime_type

Mensaje de tipo MIME. Por defecto el valor es 'text/plain; charset=us-ascii'.

priority

Prioridad del mensaje. 1 es la prioridad más alta y 5 la más baja. El valor por defecto es 3.

replyto

Define a quién se le reenviará el mensaje.

Procedimiento SEND_ATTACH_RAW

Este procedimiento es similar al anterior (SEND) pero sobrecargado para que admita adjuntar elementos RAW.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_MAIL.SEND_ATTACH_RAW(
sender      IN VARCHAR2 CHARACTER SET ANY_CS,
recipients  IN VARCHART2 CHARACTER SET ANY_CS,
cc          IN VARCHART2 CHARACTER SET ANY_CS
            DEFAULT NULL,
bcc         IN VARCHART2 CHARACTER SET ANY_CS
            DEFAULT NULL,
subject     IN VARCHART2 CHARACTER SET ANY_CS
            DEFAULT NULL,
message     IN VARCHART2 CHARACTER SET ANY_CS
            DEFAULT NULL,
mime_type   IN VARCHART2 DEFAULT CHARACTER SET
            ANY_CS DEFAULT 'text/plain;
            character=us-ascii',
priority    IN PLS_INTEGER DEFAULT 3,
attachment  IN RAW,
att_inline  IN BOOLEAN DEFAULT TRUE,
att_mime_type IN VARCHAR2 CHARACTER SET ANY_CS
            DEFAULT 'text/plain; character=us-
            ascii',
att_filename IN VARCHAR2 SET ANY_CS DEFAULT NULL,
replyto     IN VARCHART2 CHARACTER SET ANY_CS
            DEFAULT NULL);
```

DESCRIPCIÓN DE LOS PARÁMETROS

A continuación se describen cada uno de los parámetros que se usan en el procedimiento SEND_ATTACH_RAW.

sender

Dirección email del que envía el mensaje.

recipients

Dirección email del receptor o receptores del mensaje (en caso de varios, han de ir separados por comas).

cc

Dirección email del receptor o receptores a los que se les envía una copia del mensaje. Al igual que antes, si son varios se separarán con comas. El valor por defecto es NULL.

bcc

Dirección email del receptor o receptores a los que se les envía una copia oculta del mensaje. Al igual que antes, si son varios se separarán con comas. El valor por defecto es NULL.

subject

Cadena de texto para identificar el asunto que se incluye en el mensaje.

message

Texto correspondiente al cuerpo del mensaje.

mime_type

Mensaje de tipo MIME. Por defecto el valor es 'text/plain; charset=us-ascii'.

priority

Prioridad del mensaje. 1 es la prioridad más alta y 5 la más baja. El valor por defecto es 3.

attachment

Fichero de tipo RAW adjunto.

att_inline

Especifica si la información adjunta es visible dentro del propio cuerpo del mensaje. Por defecto el valor es TRUE.

att_mime_type

Tipo MIME del la información adjunta. Por defecto es 'application/octet'.

att_filename

Cadena de texto que especifica el nombre del fichero contenido en la información adjunta al mensaje. Por defecto el valor es NULL.

replyto

Define a quién se le reenviará el mensaje.

Procedimiento SEND_ATTACH_VARCHAR2

Este procedimiento es similar al anterior (SEND) pero sobrecargado para que admita adjuntar elementos VARCHAR2.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_MAIL.SEND_ATTACH_VARCHAR2 (
  sender      IN VARCHAR2 CHARACTER SET ANY_CS,
  recipients  IN VARCHAR2 CHARACTER SET ANY_CS,
  cc          IN VARCHAR2 CHARACTER SET ANY_CS
              DEFAULT NULL,
  bcc         IN VARCHAR2 CHARACTER SET ANY_CS
              DEFAULT NULL,
  subject     IN VARCHAR2 CHARACTER SET ANY_CS
              DEFAULT NULL,
```

Antolín Muñoz-Chaparro

Jefe de Proyecto-Sistemas Informáticos

División-III de Aplicaciones de Costes de Personal y Pensiones Públicas

Oficina de Informática-Presupuestaria (DIR3: EA002/952)

Intervención General de la Administración del Estado




```

message      IN VARCHART2 CHARACTER SET ANY_CS
              DEFAULT NULL,
mime_type    IN VARCHART2 DEFAULT CHARACTER SET
              ANY_CS DEFAULT 'text/plain;
              character=us-ascii',
priority     IN PLS_INTEGER DEFAULT 3
attachment   IN VARCHAR2 CHARACTER SET ANY_CS,
att_inline   IN BOOLEAN DEFAULT TRUE,
att_mime_type IN VARCHAR2 CHARACTER SET ANY_CS
              DEFAULT 'text/plain; character=us-
              ascii',
att_filename IN VARCHAR2 SET ANY_CS DEFAULT NULL,
replyto      IN VARCHAR2 CHARACTER SET ANY_CS DEFAULT
              NULL);

```

DESCRIPCIÓN DE LOS PARÁMETROS

A continuación se describen cada uno de los parámetros que se usan en el procedimiento `SEND_ATTACH_VARCHAR2`.

sender

Dirección email del que envía el mensaje.

recipients

Dirección email del receptor o receptores del mensaje (en caso de varios, han de ir separados por comas).

cc

Dirección email del receptor o receptores a los que se les envía una copia del mensaje. Al igual que antes, si son varios se separarán con comas. El valor por defecto es NULL.

bcc

Dirección email del receptor o receptores a los que se les envía una copia oculta del mensaje. Al igual que antes, si son varios se separarán con comas. El valor por defecto es NULL.

subject

Cadena de texto para identificar el asunto que se incluye en el mensaje.

message

Texto correspondiente al cuerpo del mensaje.

mime_type

Mensaje de tipo MIME. Por defecto el valor es 'text/plain; charset=us-ascii'.

priority

Prioridad del mensaje. 1 es la prioridad más alta y 5 la más baja. El valor por defecto es 3.

attachment

Fichero de tipo texto adjunto.

att_inline

Especifica si la información adjunta es visible dentro del propio cuerpo del mensaje. Por defecto el valor es TRUE.

att_mime_type

Tipo MIME de la información adjunta. Por defecto es 'text/plain; charset=us-ascii'.

att_filename

Cadena de texto que especifica el nombre del fichero contenido en la información adjunta al mensaje. Por defecto el valor es NULL.

replyto

Define a quién se le reenviará el mensaje.

UTL_HTTP

El paquete UTL_HTTP hace llamadas al protocolo HTTP (Hypertext Transfer Protocol) desde SQL y PL/SQL.

Cuando el paquete actúa contra datos de un sitio Web usando HTTPS, requiere Oracle Wallet Manager que puede ser creado por Oracle Wallet Manager o por la utilidad `orapki`.

Este paquete una serie de constantes agrupadas en las siguientes categorías:

- Constantes sobre versiones HTTP.
- Constantes sobre puertos por defecto.
- Constantes sobre códigos de estado HTTP 1.1.

Las constantes sobre versiones HTTP son:

Nombre Constante	Tipo	Valor	Descripción
HTTP_VERSION_1_0	VARCHAR2(10)	'HTTP/1.0'	Implica una versión 1.0 HTTP que puede ser usada en la función <code>BEGIN_REQUEST</code> .
Nombre Constante	Tipo	Valor	Descripción
HTTP_VERSION_1_1	VARCHAR2(10)	'HTTP/1.1'	Implica una versión 1.1 HTTP que puede ser usada en la función <code>BEGIN_REQUEST</code> .

Las constantes sobre puertos por defecto son:

Nombre Constante	Tipo	Valor	Descripción
DEFAULT_HTTP_PORT	PLS_INTEGER	80	El puerto por defecto TCP/IP 80 en el que escucha un servidor Web o un servidor proxy.
DEFAULT_HTTPS_PORT	PLS_INTEGER	443	El puerto por defecto TCP/IP 443 en el que

		escucha un servidor HTTPS.
--	--	----------------------------

Las constantes sobre códigos de estado HTTP 1.1 son numerosas y quedan fuera de la redacción de este curso. Puede consultarlas en la documentación oficial de Oracle: *PL_SQL Package and Types Reference* (<http://docs.oracle.com/database/121/ARPLS/toc.htm>).

Este paquete también presenta la definición de tipos de datos personalizados y nominados de la siguiente forma:

- Tipo REQ.
- Tipo RESP.
- Tipos COOKIE y COOKIE_TABLE.
- Tipo CONNECTION.
- Tipo REQUEST_CONTEXT_KEY.

Por último, consta de una serie de subprogramas agrupados en las siguientes categorías:

- Subprogramas de llamadas HTTP simples.
- Subprogramas de configuración de sesión.
- Subprogramas de petición HTTP.
- Subprogramas de peticiones contextuales HTTP.
- Subprogramas de respuesta HTTP.
- Subprogramas de COOKIES HTTP.
- Subprogramas de conexiones HTTP persistentes.
- Subprogramas para el tratamiento de las condiciones de error.
- Constantes sobre códigos de estado HTTP 1.1.

Los subprogramas de llamadas HTTP simples son los siguientes:

Nombre Subprograma	Tipo	Descripción
REQUEST	Función	Devuelve por encima de los primeros 2000 bytes de los datos devueltos de la URL indicada. Esta

		función se puede utilizar directamente en consultas SQL.
REQUEST_PIECES	Función	Devuelve una tabla PL/SQL con celdas de 2000 bytes cargadas con los datos devueltos por la URL indicada.

Los subprogramas de configuración de sesión son los siguientes:

Nombre Subprograma	Tipo	Descripción
GET_BODY_CHARSET	Procedimiento	Devuelve el juego de caracteres por defecto del cuerpo de todas las peticiones HTTP futuras.
GET_COOKIE_SUPPORT	Procedimiento	Devuelve la configuración de la COOKIE actual soportada.
GET_DETAILED_EXCP_SUPPORT	Procedimiento	Comprueba si el paquete UTL_HTTP abortará por una excepción detallada o no.
GET_FOLLOW_REDIRECT	Procedimiento	Devuelve la configuración del siguiente redireccionamiento en la sesión actual.
GET_PERSISTENT_CONN_SUPPORT	Procedimiento	Comprueba si el soporte de la conexión persistente está habilitado y obtiene el máximo de conexiones persistentes en la sesión actual.
GET_PROXY	Procedimiento	Devuelve la configuración del proxy actual.
GET_RESPONSE_ERROR_CHECK	Procedimiento	Comprueba si está activado el control de errores de respuesta o no.
GET_TRANSFER_TIMEOUT	Procedimiento	Devuelve el valor actual del TIMEOUT de transferencia de red.

SET_TRANSFER_TIMEOUT	Procedimiento	Establece el conjunto de caracteres por defecto del cuerpo para las peticiones http futuras cuando el MEDIA TYPE es texto y no se ha especificado un conjunto de caracteres para la cabecera CONTENT_TYPE.
SET_COOKIE_SUPPORT	Procedimiento	Establece si una futura petición HTTP soportará o no cookies. Establece el número máximo de cookies mantenidas en la sesión actual de base de datos del usuario.
SET_DETAILED_EXCP_SUPPORT	Procedimiento	Establece que el paquete UTL_HTTP abortará por una excepción detallada.
SET_FOLLOW_REDIRECT	Procedimiento	Establece el número máximo de veces que UTL_HTTP seguirán las instrucciones de redireccionamiento HTTP en la respuesta a las peticiones futuras en la función GET_RESPONSE
SET_PERSISTENT_CONN_SUPPORT	Procedimiento	Establece si la petición http soportará o no conexiones persistentes http 1.1. Establece el número máximo de conexiones persistentes mantenidas en la sesión actual de base de datos del usuario.
SET_PROXY	Procedimiento	Establece el proxy que se utilizará para las peticiones de HTTP u otros protocolos.

SET_RESPONSE_ERROR_CHECK	Procedimiento	Establece si GET_RESPONSE abortará o no por una excepción cuando el servidor Web devuelva un código de estado que indique un error (códigos en los rangos 4xx o 5xx).
SET_TRANSFER_TIMEOUT	Procedimiento	Establece el valor del TIMEOUT para las lecturas de las respuestas HTTP desde el servidor Web o servidor proxy.
SET_WALLET	Procedimiento	Establece el Oracle Wallet utilizado para todas las peticiones HTTP sobre SSL, que sean HTTPS.

Los subprogramas de peticiones HTTP son los siguientes:

Nombre Subprograma	Tipo	Descripción
BEGIN_REQUEST	Función	Comienza una nueva petición HTTP.
SET_HEADER	Procedimiento	Establece una cabecera de petición HTTP.
SET_AUTHENTICATION	Procedimiento	Establece la información de autenticación en la cabecera de petición HTTP.
SET_AUTHENTICATION_FROM_WALLET	Procedimiento	Establece la información de autenticación en la cabecera de petición HTTP, necesaria para la petición de autorización al servidor Web, usando las credenciales (usuario y password) almacenadas en Oracle Wallet.

SET_BODY_CHARSET	Procedimiento	Establece el conjunto de caracteres del cuerpo de la petición cuando el MEDIA TYPE es texto y no se ha especificado un conjunto de caracteres en la cabecera CONTENT-TYPE.
SET_COOKIE_SUPPORT	Procedimiento	Establece si una futura petición HTTP soportará o no cookies. Establece el número máximo de cookies mantenidas en la sesión actual de base de datos del usuario.
SET_FOLLOW_REDIRECT	Procedimiento	Establece el número máximo de veces que UTL_HTTP seguirán las instrucciones de redireccionamiento HTTP en la respuesta a las peticiones futuras en la función GET_RESPONSE
SET_PERSISTENT_CONN_SUPPORT	Procedimiento	Establece si la petición http soportará o no conexiones persistentes http 1.1.
SET_PROXY	Procedimiento	Escribe una línea de texto en el cuerpo de la petición http y finaliza la línea con un carácter de nueva línea (CRLF – retorno de carro y salto de línea).
WRITE_RAW	Procedimiento	Escribe algunos datos binarios en el cuerpo de la petición HTTP.
WRITE_TEXT	Procedimiento	Escribe algunos datos de texto en el cuerpo de la petición HTTP.

Los subprogramas de peticiones contextuales HTTP son los siguientes:

Nombre Subprograma	Tipo	Descripción
CREATE_REQUEST_CONTEXT	Función	Crea una petición contextual in UTL_HTTP para una cartera(WALLET) y una tabla de cookies.
DESTROY_REQUEST_CONTEXT	Procedimiento	Destruye la petición contextual en UTL_HTTP.

Los subprogramas de COOKIES HTTP son los siguientes:

Nombre Subprograma	Tipo	Descripción
END_RESPONSE	Procedimiento	Finaliza la respuesta HTTP.
GET_AUTHENTICATION	Procedimiento	Recupera la información de autenticación http necesaria para que la petición sea aceptada por el servidor Web como se indica en la cabecera de respuesta HTTP.
GET_HEADER	Procedimiento	Devuelve los "n" nombres de la cabecera de respuesta HTTP y el valor devuelto en la respuesta.
GET_HEADER_BY_NAME	Procedimiento	Devuelve el valor de la cabecera de respuesta devuelto al acceder por el nombre de la cabecera.
GET_HEADER_COUNT	Función	Devuelve el número de cabeceras de respuesta HTTP devueltas en la respuesta.
GET_RESPONSE	Función	Lee la respuesta HTTP.
READ_LINE	Procedimiento	Lee el cuerpo de la respuesta HTTP en formato de texto hasta

		que se alcance el final de línea y devuelve la salida en un buffer.
READ_RAW	Procedimiento	Lee el cuerpo de la respuesta HTTP en formato binario y devuelve la salida en un buffer.
READ_TEXT	Procedimiento	Lee el cuerpo de la respuesta HTTP en formato texto y devuelve la salida en un buffer.
SET_BODY_CHARSET	Procedimiento	Establece el conjunto de caracteres del cuerpo de la petición cuando el MEDIA TYPE es texto y no se ha especificado un conjunto de caracteres en la cabecera CONTENT-TYPE.

Los subprogramas de respuestas HTTP son los siguientes:

Nombre Subprograma	Tipo	Descripción
ADD_COOKIES	Procedimiento	Añade la cookie a la petición contextual o al estado de la sesión del paquete UTL_HTTP.
CLEAR_COOKIES	Procedimiento	Limpia todas las cookies que se mantienen en la petición contextual o en el estado de la sesión del paquete UTL_HTTP.
GET_COOKIE_COUNT	Función	Devuelve el número de cookies que se mantienen en la petición contextual o en el estado de la sesión del paquete UTL_HTTP.
GET_COOKIES	Función	Devuelve todas las cookies que se mantienen en la petición contextual o en el estado de la sesión del paquete UTL_HTTP.

Los subprogramas de las conexiones persistentes HTTP son los siguientes:

Nombre Subprograma	Tipo	Descripción
CLOSE_PERSISTENT_CONN	Procedimiento	Cierra una conexión persistente HTTP que se mantiene por el paquete UTL_HTTP en la sesión de base de datos actual.
CLOSE_PERSISTENT_CONNS	Procedimiento	Cierra un grupo de conexiones persistentes http que se mantienen por el paquete UTL_HTTP en la sesión de base de datos actual.
GET_PERSISTENT_CONN_COUNT	Función	Devuelve el número de conexiones de red que actualmente se mantienen persistente por el paquete UTL_HTTP a los servidores Web.
GET_PERSISTENT_CONNS	Procedimiento	Devuelve todas las conexiones de red que actualmente se mantienen persistentes por el paquete UTL_HTTP a los servidores Web.

Los subprogramas para el control de las condiciones de error son los siguientes:

Nombre Subprograma	Tipo	Descripción
GET_DETAILED_SQLCODE	Función	Devuelve el valor de SQLCODE del último error que se ha producido.
GET_DETAILED_SQLERRM	Función	Devuelve el valor de SQLERRM del último error que se ha producido.

EJEMPLO

A continuación se muestra un ejemplo de uso del paquete UTL_HTTP de manera general:

```
SET SERVEROUTPUT ON SIZE 40000

DECLARE
    req          UTL_HTTP.REQ
    resp         UTL_HTTP.RESP;
    value        VARCHAR2(1024);

BEGIN
    UTL_HTTP.SET_PROXY('proxy.my-company.com',
                      'corp.my-company.com');
    req := UTL_HTTP.BEGIN_REQUEST('http://www-
hr.corp.my-company.com');
    UTL_HTTP.SET_HEADER(req, 'User-Agent',
                        'Mozilla/4.0');
    resp := UTL_HTTP.GET_RESPONSE(req);
    LOOP
        UTL_HTTP.READ_LINE(resp, value, TRUE);
        DBMS_OUTPUT.PUT_LINE(value);
    END LOOP;
    UTL_HTTP.END_RESPONSE(resp);
EXCEPTION
    WHEN UTL_HTTP.END_OF_BODY THEN
        UTL_HTTP.END_RESPONSE(resp);
END;
```

Errores de la sección EXCEPTION

El paquete UTL_HTTP puede devolver los siguientes errores que se describen continuación:

Excepción	Cód.Error	Descripción
BAD_ARGUMENT	29261	Son incorrectos los argumentos que se han pasado a la interface.

BAD_URL	29262	La petición URL está incorrectamente formulada.
END_OF_BODY	29266	Se alcanzó el final del cuerpo de respuesta HTTP.
HEADER_NOT_FOUND	29265	No se encuentra la cabecera.
HTTP_CLIENT_ERROR	29268	Desde GET_RESPONSE, el parámetro STATUS_CODE indica que ha ocurrido un error en el cliente (código de error en el rango 4xx). Desde BEGIN_REQUEST, indica que el proxy ha devuelto un código de estado en el rango 4xx cuando hizo una petición HTTPS a través del proxy.
HTTP_SERVER_ERROR	29269	Desde GET_RESPONSE, el código de estado de respuesta indica que ha ocurrido un error en el cliente (código de error en el rango 5xx). O desde BEGIN_REQUEST, el proxy HTTP devuelve un código de estado en el rango 4xx cuando una petición HTTPS a través del proxy.
NETWORK_ACCESS_DENIED	24247	El acceso al equipo de red remoto o las credenciales de Oracle se han denegado.
ILLEGAL_CALL	29267	La llamada a UTL_HTTP es ilegal en el estado actual de la petición HTTP.
PARTIAL_MULTIBYTE_EXCEPTION	29275	No se ha leído ningún carácter completo y se encontró un carácter parcial multibyte al final del cuerpo de respuesta.
PROTOCOL_ERROR	29263	Un error de protocolo HTTP ocurrió cuando se comunicaba con el serv. Web.
REQUEST_FAILED	29273	La petición falló al ejecutarse.
TOO_MANY_REQUESTS	29270	Demasiadas peticiones o respuestas están abiertas.
TRANSFER_TIMEOUT	29276	No se ha leído ningún dato y ocurrió un timeout (finalización por exceso de tiempo) en la lectura.
UNKNOWN_SCHEME	29264	El esquema de la petición URL es desconocido.

Tipo REQ

Se usa este tipo de registro PL/SQL para representar una petición HTTP.

La información que se devuelve sobre este tipo desde la interface `BEGIN_REQUEST` es sólo para lectura. Los cambios que se hagan en los valores del registro no tienen efecto sobre la petición.

SINTAXIS

La sintaxis de este tipo de dato es la siguiente:

```
TYPE req IS RECORD(   url           VARCHAR2 (32767) ,
                      method        VARCHAR2 (64) ,
                      http_version  VARCHAR2 (64) );
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL TIPO DE DATO

A continuación se describen cada uno de los parámetros que se usan en la definición del tipo de dato REQ.

url

La URL de la petición HTTP.

method

Es el método a llevar a cabo en el recurso identificado por la URL.

http_version

La versión del protocolo HTTP que se usa para enviar la petición.

Tipo REQUEST_CONTEXT_TYPE

Este tipo se usa para representar la clave del contexto de una petición. El contexto de una petición engloba una parte privada y una tabla de cookies para hacer la petición HTTP.

SINTAXIS

La sintaxis de este tipo de dato es la siguiente:

```
TYPE request_context_key IS PLS_INTEGER;
```

Tipo RESP

Se usa este tipo de registro PL/SQL para representar una respuesta HTTP.

La información que se devuelve sobre este tipo desde la interface GET_RESPONSE es sólo para lectura.

SINTAXIS

La sintaxis de este tipo de dato es la siguiente:

```
TYPE resp IS RECORD( status_code    VARCHAR2(32767),
                     reason_phrase  VARCHAR2(64),
                     http_version   VARCHAR2(64));
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL TIPO DE DATO

A continuación se describen cada uno de los parámetros que se usan en la definición del tipo de dato RESP.

status_code

El código de estado devuelto por el servidor Web. Es un entero de 3 dígitos que indica el resultado de la petición http como manejador del servidor Web.

reason_phrase

El mensaje corto de texto devuelto por el servidor Web que describe el código de estado. Ofrece una breve descripción del resultado de la petición HTTP.

http_version

La versión del protocolo HTTP que se usa para enviar la respuesta.

Tipos COOKIE y COOKIE_TABLE

El tipo COOKIE es un tipo registro PL/SQL que representa una cookie HTTP. El tipo COOKIE_TABLE es un tipo tabla PL/SQL que almacena elementos COOKIE en su colección de datos.

SINTAXIS

La sintaxis del tipo COOKIE es la siguiente:

```
TYPE cookie IS RECORD(
    name          VARCHAR2(256),
    value         VARCHAR2(1024),
    domain        VARCHAR2(256),
    expire        TIMESTAMP WITH TIME ZONE,
    path          VARCHAR2(1024),
    secure        BOOLEAN,
    version       PLS_INTEGER,
    comment       VARCHAR2(1024));
```

La sintaxis del tipo COOKIE_TABLE es la siguiente:

```
TYPE cookie_table IS TABLE OF cookie INDEX BY
    BINARY_INTEGER;
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL TIPO DE DATO COOKIE

A continuación se describen cada uno de los parámetros que se usan en la definición del tipo de dato COOKIE.

name

El nombre de la cookie HTTP.

value

El valor de la cookie.

domain

El dominio por el que la cookie se valida.

expire

El tiempo máximo en el que la cookie expirará.

path

El subconjunto de URLs que aplica la cookie.

secure

Indica que la cookie debería ser devuelta al servidor Web usando sólo mecanismos de seguridad.

version

La versión de la especificación del HTTP de la cookie. Este campo estará a NULL para cookies de Netscape.

comment

El comentario que describe la intención de uso de la cookie. Este campo estará a NULL para cookies de Netscape.

Tipo CONNECTION

Se usa este tipo de registro PL/SQL para representar el equipo remoto y puertos TCP/IP de una conexión de red que se mantiene persistente después de que se complete una petición HTTP de acuerdo a la especificación del protocolo HTTP 1.1.

Una conexión de red persistente puede ser reutilizada por una subsecuencia de petición HTTP en el mismo equipo y puerto.

Para una conexión HTTP persistente a un servidor Web, los parámetros HOST y PORT contienen en el nombre del equipo y el puerto TCP/IP (respectivamente), del servidor Web.

El parámetro SSL indica si se está utilizando SSL (Secured Socked Layer) en una conexión HTTP persistente. Una petición HTTPS es una petición HTTP realizada sobre SSL.

SINTAXIS

La sintaxis de este tipo de dato es la siguiente:

```
TYPE connection IS RECORD(
                                host          VARCHAR2 (256) ,
                                port         PLS_INTEGER,
                                proxy_host   VARCH2 (256) ,
                                proxy_port   PLS_INTEGER,
                                ssl          BOOLEAN) ;
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL TIPO DE DATO

A continuación se describen cada uno de los parámetros que se usan en la definición del tipo de dato CONNECTION.

host

El nombre del equipo al que se quiere conectar.

port

El puerto por el que se va a conectar con el equipo.

proxy_host

El nombre del proxy que se utiliza en la conexión.

proxy_port

El puerto del proxy que se utiliza en la conexión.

ssl

TRUE para indicar que se va a utilizar HTTPS.

Procedimiento ADD_COOKIES

Añade la cookie a la petición contextual o al estado de la sesión del paquete UTL_HTTP.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.ADD_COOKIES (
    cookies           IN cookie_table,
    request_context   IN resquest_context_key
                      DEFAULT NULL);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento ADD_COOKIES.

cookies

Las cookies que se añadirán.

request_context

La petición contextual para añadir las cookies. Si es NULL, las cookies serán añadidas al estado de la sesión del paquete UTL_HTTP en vez de a la petición contextual.

Función BEGIN_REQUEST

Comienza una nueva petición HTTP. UTL_HTTP establece la conexión de red al servidor Web de origen o al servidor proxy y envía la petición HTTP. El programa PL/SQL continúa la petición utilizando llamadas a otros interfaces para poder completarla. La URL puede contener el nombre del usuario y el password necesarios para autenticar la petición en el servidor. El formato es:

```
scheme://[user[:password]@host[:port]]/[...]
```

SINTAXIS

La sintaxis de esta función es la siguiente:

```
UTL_HTTP.BEGIN_REQUEST (url IN VARCHAR2,  
method                IN VARCHAR2 DEFAULT 'GET',  
http_version          IN VARCHAR2 DEFAULT NULL,  
request_context IN request_context_key DEFAULT NULL)  
RETURN req;
```

DESCRIPCIÓN DE LOS PARÁMETROS DE LA FUNCIÓN

A continuación se describen cada uno de los parámetros que se usan en la definición de la función `BEGIN_REQUEST`.

url

La URL de la petición HTTP.

method

El método implementado en el recurso identificado por la URL. Por defecto si no se especifica nada, se asume el método GET.

http_version

La versión del protocolo HTTP que se envía en la petición. El formato de la versión del protocolo es: `HTTP/major-version.minor-version`, donde `major-version` y `minor-version` son números positivos. Si el parámetro tiene un valor NULL, `UTL_HTTP` utiliza la versión del último protocolo HTTP que se soporta para el envío de la petición. La última versión que el paquete soporta es 1.1 y puede ser actualizada a una versión posterior. El valor por defecto del parámetro es NULL.

request_context

La petición contextual que soporta tanto la cartera (WALLET) privada como la tabla de cookies que se usa en la petición HTTP. Si el parámetro es NULL, se usará la cartera (WALLET) y la tabla de cookies compartidas en la sesión de base de datos actual.

RETURN REQ

Esta función devuelve un tipo REQ en la respuesta a la llamada que se haga sobre la misma.

Procedimiento CLEAR_COOKIE

Limpia todas las cookies que se mantienen en la petición contextual o en el estado de la sesión del paquete UTL_HTTP.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.CLEAR_COOKIE (
    request_context IN request_context_key
                    DEFAULT NULL);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento CLEAR_COOKIE.

request_context

La petición contextual para eliminar las cookies. Si es NULL, se limpiarán las cookies que se mantienen en el estado de la sesión del paquete UTL_HTTP.

Procedimiento CLOSE_PERSISTENT_CONN

Cierra una conexión persistente HTTP que se mantiene por el paquete UTL_HTTP en la sesión de base de datos actual.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.CLOSE_PERSISTENT_CONN (conn IN connection);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento CLOSE_PERSISTENT_CONN.

conn

La conexión HTTP persistente que se va a cerrar.

Procedimiento CLOSE_PERSISTENT_CONNS

Cierra un grupo de conexiones persistentes http que se mantienen por el paquete UTL_HTTP en la sesión de base de datos actual.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.CLOSE_PERSISTENT_CONNS (
    host          IN VARCHAR2 DEFAULT NULL,
    port          IN PLS_INTEGER DEFAULT NULL,
    proxy_host    IN VARCHAR2 DEFAULT NULL,
    proxy_port    IN PLS_INTEGER DEFAULT NULL,
    ssl           IN BOOLEAN DEFAULT NULL);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento CLOSE_PERSISTENT_CONNS.

host

El equipo para el que se cerrarán las conexiones persistentes.

port

El número del puerto para el que se cerrarán las conexiones persistentes.

proxy_host

El equipo proxy para el que se cerrarán las conexiones persistentes.

proxy_port

El número del puerto del proxy para el que se cerrarán las conexiones persistentes.

ssl

Cierra las conexiones persistentes SSL.

EJEMPLOS

En el siguiente ejemplo se cierran todas las conexiones permanentes a un equipo denominado SERVIDOR1:

```
UTL_HTTP.CLOSE_PERSISTENT_CONNS(host => 'servidor1');
```

En el siguiente ejemplo se cierran todas las conexiones a través del proxy PROXY1, por el puerto 80:

```
UTL_HTTP.CLOSE_PERSISTENT_CONNS(proxy_host => 'proxy1',  
                                proxy_port => 80);
```

En este último ejemplo se cierran todas las conexiones permanentes:

```
UTL_HTTP.CLOSE_PERSISTENT_CONNS;
```

Función CREATE_REQUEST_CONTEXT

Crea una petición contextual. Una petición contextual es un contexto que soporta una cartera (WALLET) y una cookie para uso privado en la petición HTTP. Esto permite a la petición HTTP use una cartera (WALLET) y una tabla de cookies que no será compartida con otras aplicaciones que hacen peticiones HTTP en la misma sesión de base de datos.

SINTAXIS

La sintaxis de esta función es la siguiente:

Antolín-Muñoz-Chaparro
Jefe-de-Proyecto-Sistemas-Informáticos
División-III-de-Aplicaciones-de-Costos-de-Personal-y-Pensiones-Públicas
Oficina-de-Informática-Presupuestaria-(DIR3-EA0027952)
Intervención-General-de-la-Administración-del-Estado



```

UTL_HTTP.CREATE_REQUEST_CONTEXT (
wallet_path           IN VARCHAR2 DEFAULT NULL,
wallet_password       IN VARCHAR2 DEFAULT NULL,
enable_cookies        IN BOOLEAN  DEFAULT TRUE,
max_cookies           IN PLS_INTEGER DEFAULT 300,
max_cookies_per_site IN PLS_INTEGER DEFAULT 20)
RETURN request_context_key;

```

DESCRIPCIÓN DE LOS PARÁMETROS DE LA FUNCIÓN

A continuación se describen cada uno de los parámetros que se usan en la definición de la función CREATE_REQUEST_CONTEXT.

wallet_path

La ruta del directorio que contiene la cartera (WALLET) Oracle. El formato es file:directory-path.

wallet_password

El password necesario para abrir la cartera (WALLET). Si la cartera tiene habilitado el sistema de auto-conexión, el password se puede omitir y debería establecerse este parámetro a NULL.

enable_cookies

Establece si la petición HTTP debería soportar cookies HTTP o no. TRUE para habilitar que las soporte y FALSE para deshabilitarlo.

max_cookies

Establece el número máximo de cookies que se mantendrán en la petición contextual.

max_cookies_per_site

Establece el número máximo de cookies por cada sitio Web que se mantendrán en la petición contextual.

RETURN REQUEST_CONTEXT_KEY

Esta función devuelve la petición contextual creada, mediante un tipo REQUEST_CONTEXT_KEY.

EJEMPLO

```
DECLARE
    request_context      UTL_HTTP.REQUEST_CONTEXT_KEY;
    req                  UTL_HTTP.req;
BEGIN
    request_context := UTL_HTTP.CREATE_REQUEST_CONTEXT(
        wallet_path => 'file:/oracle/wallets/test_wallets',
        wallet_password => NULL,
        enable_cookies => TRUE,
        max_cookies => 300,
        max_cookies_per_site => 20);
    req := UTL_HTTP.BEGIN_REQUEST(
        url => 'http://www.example.com/',
        request_context => request_context);
END;
```

Procedimiento DESTROY_REQUEST_CONTEXT

Destruye una petición contextual en UTL_HTTP. Una petición contextual no puede ser destruida cuando está siendo usada por una petición o respuesta HTTP.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.DESTROY_REQUEST_CONTEXT(
    request_context IN request_context_key);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento DESTROY_REQUEST_CONTEXT.

request_context

Petición contextual a destruir.

EJEMPLO

```

DECLARE
    request_context          UTL_HTTP.REQUEST_CONTEXT_KEY;
BEGIN
    request_context := UTL_HTTP.CREATE_REQUEST_CONTEXT (...);
    UTL_HTTP.DESTROY_REQUEST_CONTEXT(request_context);
END;
```

Procedimiento END_REQUEST

Finaliza una petición HTTP. Para terminarla sin completar la petición y esperar una respuesta, el programa puede llamar a este procedimiento. La otra manera de operar debería seguir la secuencia normal que consiste en comenzar una petición, obtener la respuesta y cerrar la respuesta. La conexión de red siempre será cerrada y no será reusada.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.END_REQUEST (r      IN OUT NOCOPY req);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento END_REQUEST.

r

La petición HTTP.

Antolín Muñoz Chaparro

Jefe de Proyecto-Sistemas Informáticos

División-III de Aplicaciones de Costes de Personal y Pensiones Públicas

Oficina de Informática Presupuestaria (DIR3: EAO027952)

Intervención General de la Administración del Estado



Procedimiento END_RESPONSE

Finaliza una respuesta HTTP. Completa la petición y la respuesta HTTP.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.END_RESPONSE(r      IN OUT NOCOPY resp);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento END_RESPONSE.

r

La respuesta HTTP.

Procedimiento GET_AUTHENTICATION

Devuelve la información de autenticación HTTP necesaria para que la petición sea aceptada por el servidor Web como se indicó en la cabecera de respuesta HTTP.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.GET_AUTHENTICATION(
    r              IN OUT NOCOPY resp,
    scheme         OUT VARCHAR2,
    realm          OUT VARCHAR2,
    for_proxy      IN BOOLEAN DEFAULT FALSE);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento GET_AUTHENTICATION.

r

La respuesta HTTP.

scheme

El esquema para la autenticación HTTP requerida.

realm

El realm para la autenticación HTTP requerida.

for_proxy

Devuelve la información de autenticación HTTP requerida para el acceso al servidor proxy HTTP en vez de la del servidor Web. El valor por defecto es FALSE.

Procedimiento GET_BODY_CHARSET

Devuelve el juego de caracteres por defecto para el cuerpo de todas las próximas peticiones HTTP.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.GET_BODY_CHARSET(charset OUT VARCHAR2);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento GET_BODY_CHARSET.

charset

El juego de caracteres por defecto.

Función GET_COOKIE_COUNT

Devuelve el número de cookies que se mantienen tanto en la petición contextual como en la sesión de estado del paquete UTL_HTTP.

SINTAXIS

La sintaxis de esta función es la siguiente:

```
UTL_HTTP.GET_COOKIE_COUNT (
    request_context IN request_context_key DEFAULT NULL)
RETURN PLS_INTEGER;
```

DESCRIPCIÓN DE LOS PARÁMETROS DE LA FUNCIÓN

A continuación se describen cada uno de los parámetros que se usan en la definición de la función GET_COOKIE_COUNT.

request_context

La petición contextual para la que devuelve el conteo de cookies. Si es NULL el conteo que se devuelve corresponde a la sesión de estado del paquete UTL_HTTP.

RETURN PLS_INTEGER

Esta función devuelve en un tipo PLS_INTEGER el conteo de cookies.

Procedimiento GET_COOKIE_SUPPORT

Devuelve la configuración de soporte de la cookie actual.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.GET_COOKIE_SUPPORT (
    enable                OUT BOOLEAN,
    max_cookies           OUT PLS_INTEGER,
    max_cookies_per_site OUT PLS_INTEGER);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento GET_COOKIE_SUPPORT.

enable

Indica si la futura petición HTTP debería soportar cookies HTTP (en caso de que el valor sea TRUE) o no las debería soportar (en caso de que el valor sea FALSE).

max_cookies

Indica el número total máximo de cookies que se mantienen en la sesión actual.

max_cookies_per_site

Indica el número total máximo de cookies que se mantienen en la sesión actual de cada sitio Web.

Función GET_COOKIES

Devuelve todas las cookies que se mantienen tanto en la petición contextual como en el estado de la sesión del paquete UTL_HTTP.

SINTAXIS

La sintaxis de esta función es la siguiente:

```
UTL_HTTP.GET_COOKIES (
  cookies           IN OUT NOCOPY cookie_table,
  request_context IN request_context_key DEFAULT
  NULL) ;
```

DESCRIPCIÓN DE LOS PARÁMETROS DE LA FUNCIÓN

A continuación se describen cada uno de los parámetros que se usan en la definición de la función GET_COOKIES.

cookies

Las cookies devueltas.

request_context

La petición contextual para la que devuelve las cookies. Si el valor es `NULL`, se devuelven las cookies del estado de la sesión del paquete `UTL_HTTP`.

Procedimiento GET_DETAILED_EXCP_SUPPORT

Chequea si el paquete `UTL_HTTP` alcanzará una excepción detallada o no.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.GET_DETAILED_EXCP_SUPPORT(  
    enable          OUT BOOLEAN);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento `GET_DETAILED_EXCP_SUPPORT`.

enable

`TRUE` si `UTL_HTTP` alcanza una excepción detallada; en caso contrario `FALSE`.

Función GET_DETAILED_SQLCODE

Devuelve el parámetro SQLCODE detallado de una excepción alcanzada.

SINTAXIS

La sintaxis de esta función es la siguiente:

```
UTL_HTTP.GET_DETAILED_SQLCODE
RETURN PLS_INTEGER;
```

Función GET_DETAILED_SQLERRM

Devuelve el parámetro SQLERRM de la última excepción alcanzada.

SINTAXIS

La sintaxis de esta función es la siguiente:

```
UTL_HTTP.GET_DETAILED_SQLERRM
RETURN VARCHAR2;
```

Procedimiento GET_FOLLOW_REDIRECT

Devuelve la configuración del siguiente redireccionamiento en la sesión actual.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.GET_FOLLOW_REDIRECT (
    Max_redirects OUT PLS_INTEGER);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento GET_FOLLOW_REDIRECT.

Antolín Muñoz Chaparro

Jefe de Proyecto-Sistemas Informáticos

División-III de Aplicaciones de Costes de Personal y Pensiones Públicas

Oficina de Informática Presupuestaria (DIR3: EAO027952)

Intervención General de la Administración del Estado



max_redirects

El número máximo de redireccionamientos para todas las peticiones http futuras.

Procedimiento GET_HEADER

Devuelve el *n*ésimo nombre de cabecera de respuesta HTTP y el valor devuelto en la respuesta.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.GET_HEADER (
  r          IN OUT NOCOPY resp,
  n          IN PLS_INTEGER,
  name       OUT NOCOPY VARCHAR2,
  value      OUT NOCOPY VARCHAR2);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento GET_HEADER.

r

La respuesta HTTP.

n

La *n*ésima cabecera a devolver.

name

El nombre de la cabecera de respuesta HTTP.

value

El valor de la cabecera de respuesta HTTP.

Procedimiento GET_HEADER_BY_NAME

Devuelve el valor de la cabecera de respuesta, devuelto en la respuesta dada por el nombre de la cabecera.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.GET_HEADER_BY_NAME (  
  r          IN OUT NOCOPY resp,  
  name       IN VARCHAR2,  
  value      OUT NOCOPY VARCHAR2,  
  n          IN PLS_INTEGER DEFAULT 1);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento GET_HEADER_BY_NAME.

r

La respuesta HTTP.

name

El nombre de la cabecera de respuesta HTTP para la que el valor es devuelto.

value

El valor de la cabecera de respuesta HTTP.

n

La enésima ocurrencia de una cabecera de respuesta HTTP por el nombre especificado para devolver. El valor por defecto es 1.

Función GET_HEADER_COUNT

Devuelve el número de cabeceras de respuesta HTTP devuelta en la respuesta.

SINTAXIS

La sintaxis de esta función es la siguiente:

```
UTL_HTTP.GET_HEADER_COUNT (
    r                               IN OUT NOCOPY resp)
RETURN PLS_INTEGER;
```

DESCRIPCIÓN DE LOS PARÁMETROS DE LA FUNCIÓN

A continuación se describen cada uno de los parámetros que se usan en la definición de la función GET_HEADER_COUNT.

r

La respuesta HTTP.

RETURN PLS_INTEGER

Esta función devuelve en un tipo PLS_INTEGER el número de cabeceras.

Procedimiento GET_PERSISTENT_CONN_SUPPORT

Este procedimiento chequea:

- Si está habilitado el soporte para las conexiones persistentes.
- Obtención del número máximo de conexiones persistente en la sesión actual.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.GET_PERSISTENT_CONN_SUPPORT (
    enable          OUT BOOLEAN,
```

```
max_conns OUT PLS_INTEGER);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento GET_PERSISTENT_CONN_SUPPORT.

enable

TRUE si se ha habilitado el soporte para conexiones persistentes; en caso contrario FALSE.

max_conns

El número máximo de conexiones persistentes que se mantienen en la sesión actual.

Procedimiento GET_PERSISTENT_CONNS

Devuelve todas las conexiones de red que se mantienen persistentes por el paquete UTL_HTTP en los servidores web.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.GET_PERSISTENT_CONNS (
    connections IN OUT NOCOPY connection_table);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento GET_PERSISTENT_CONNS.

connections

Las conexiones de red que se mantienen persistentes.

Función GET_PERSISTENT_CONN_COUNT

Devuelve el número de conexiones de red que actualmente se mantienen persistentes por el paquete UTL_HTTP del servidor Web.

SINTAXIS

La sintaxis de esta función es la siguiente:

```
UTL_HTTP.GET_PERSISTENT_CONN_COUNT
RETURN PLS_INTEGER;
```

Procedimiento GET_PROXY

Devuelve la configuración actual del proxy.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.GET_PROXY (
proxy          OUT NOCOPY VARCHAR2,
no_proxy_domains OUT NOCOPY VARCHAR2);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento GET_PROXY.

proxy

El proxy (equipo y número de puerto opcional) que está siendo usado actualmente por el paquete UTL_HTTP.

no_proxy_domains

La lista de los equipos y dominios en los que no se utiliza ningún proxy para todas las peticiones.

Función GET_RESPONSE

Esta función devuelve la respuesta HTTP. Cuando la función devuelve la información, entonces la línea de estado y las cabeceras de respuesta HTTP han sido leídas y procesadas. Esta función completa la sección de cabeceras HTTP.

SINTAXIS

La sintaxis de esta función es la siguiente:

```
UTL_HTTP.GET_RESPONSE (
    r          IN OUT NOCOPY req)
RETURN resp;
```

DESCRIPCIÓN DE LOS PARÁMETROS DE LA FUNCIÓN

A continuación se describen cada uno de los parámetros que se usan en la definición de la función GET_RESPONSE.

r

La respuesta HTTP.

RETURN resp

Esta función devuelve en un tipo RESP la respuesta HTTP.

Procedimiento GET_RESPONSE_ERROR_CHECK

Chequea si se ha habilitado o no la comprobación de errores en la respuesta.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.GET_RESPONSE_ERROR_CHECK (
    enable          OUT BOOLEAN);
```

Antolín Muñoz-Chaparro

Jefe de Proyecto-Sistemas Informáticos

División III de Aplicaciones de Costes de Personal y Pensiones Públicas

Oficina de Informática Presupuestaria (DIR3: EAO027952)

Intervención General de la Administración del Estado



DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento GET_RESPONSE_ERROR_CHECK.

enable

TRUE si se ha habilitado la comprobación de errores en la respuesta. En otro caso FALSE.

Procedimiento GET_TRANSFER_TIMEOUT

Retorna el valor por defecto del TIMEOUT (tiempo máximo de espera) para todas las futuras respuestas HTTP.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.GET_TRANSFER_TIMEOUT (
    timeout      OUT PLS_INTEGER);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento GET_TRANSFER_TIMEOUT.

timeout

El valor del TIMEOUT de transferencia de la red, medido en segundos.

Procedimiento READ_LINE

Lee el cuerpo de la respuesta HTTP en formato de texto hasta que se alcance el final de línea y devuelve la salida al buffer. El final de línea es el que se haya definido en la función READ_LINE de UTL_TCP. La excepción END_OF_BODY se alcanza cuando se llega al final del cuerpo de respuesta HTTP. La información que se obtiene del cuerpo

de la respuesta es convertida automáticamente desde el juego de caracteres del propio cuerpo al juego de caracteres de la base de datos.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.READ_LINE (  
  r           IN OUT NOCOPY resp,  
  data        OUT NOCOPY VARCHAR2 CHARACTER SET ANY_CS,  
  remove_crlf IN BOOLEAN DEFAULT FALSE);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento READ_LINE.

r

La respuesta HTTP.

data

El cuerpo de respuesta HTTP en formato de texto.

remove_crlf

Borra los caracteres de línea nueva si se le indica el valor TRUE.

Procedimiento READ_RAW

Lee el cuerpo de la respuesta HTTP en formato binario y devuelve la salida al buffer. La excepción END_OF_BODY se alcanza cuando se llega al final del cuerpo de respuesta HTTP.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.READ_RAW (
    r                IN OUT NOCOPY resp,
    data             OUT NOCOPY RAW,
    len              IN PLS_INTEGER DEFAULT NULL);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento READ_RAW.

r

La respuesta HTTP.

data

El cuerpo de respuesta HTTP en formato binario.

len

El número de bytes de los datos a leer. Si este parámetro es NULL, lee tantas entradas como sean posibles hasta rellenar el buffer reservado para el parámetro DATA. La cantidad actual de datos devueltos puede ser menor que la especificada si no hay suficientes datos disponibles antes de que se alcance el final del cuerpo de la respuesta o bien se ha superado el tiempo de espera especificado en TRANSFER_TIMEOUT. El valor por defecto del parámetro es NULL.

Procedimiento READ_TEXT

Lee el cuerpo de la respuesta HTTP en formato texto y devuelve la salida al buffer. La excepción END_OF_BODY se alcanza cuando se llega al final del cuerpo de respuesta HTTP. La información que se obtiene del cuerpo de la respuesta es convertida automáticamente desde el juego de caracteres del propio cuerpo al juego de caracteres de la base de datos.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.READ_TEXT(
  r          IN OUT NOCOPY resp,
  data       OUT NOCOPY VARCHAR2 CHARACTER SET ANY_CS,
  len        IN PLS_INTEGER DEFAULT NULL);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento READ_TEXT.

r

La respuesta HTTP.

data

El cuerpo de respuesta HTTP en formato texto.

len

El número de bytes de los datos a leer. Si este parámetro es NULL, lee tantas entradas como sean posibles hasta rellenar el buffer reservado para el parámetro DATA. La cantidad actual de datos devueltos puede ser menor que la especificada si no hay suficientes datos disponibles antes de que se alcance el final del cuerpo de la respuesta o bien se ha superado el tiempo de espera especificado en TRANSFER_TIMEOUT. El valor por defecto del parámetro es NULL.

Función REQUEST

Devuelve los primeros 2000 bytes de datos recuperados de la URL proporcionada. Esta función puede ser usada directamente en consultas SQL. La URL puede contener el nombre de usuario y la password necesarias para autenticarse contra el servidor. El formato es:

```
scheme://[user[:password]@]host[:port]/[...]
```

Se puede especificar un usuario y password para el proxy indicándolo con el siguiente formato:

```
[http://][user[:password]@]host[:port]/
```

SINTAXIS

La sintaxis de esta función es la siguiente:

```
UTL_HTTP.REQUEST (
    url                IN VARCHAR2,
    proxy              IN VARCHAR2 DEFAULT NULL,
    wallet_path        IN VARCHAR2 DEFAULT NULL,
    wallet_password    IN VARCHAR2 DEFAULT NULL)
RETURN VARCHAR2;
```

DESCRIPCIÓN DE LOS PARÁMETROS DE LA FUNCIÓN

A continuación se describen cada uno de los parámetros que se usan en la definición de la función REQUEST.

url

La URL especificada (Uniform Resource Locator).

proxy

Parámetro opcional. Especifica un servidor proxy que se usa cuando se hace la petición http.

wallet_path

Parámetro opcional. Especifica una cartera (wallet) del lado del cliente.

wallet_password

Parámetro opcional. Especifica la password necesaria para abrir la cartera (wallet).

RETURN VARCHAR2

El tipo devuelto es una cadena de 2000 o menos bytes que contiene los primeros 2000 bytes del resultado HTML devuelto desde la petición http para la URL especificada.

EXCEPCIONES QUE PUEDE DEVOLVER LA FUNCIÓN

Esta función puede llegar a devolver las siguientes excepciones como errores:

- INIT_FAILED
- REQUEST_FAILED

EJEMPLO

A continuación se muestra un ejemplo de uso de esta función:

```
SQL> SELECT UTL_HTTP.REQUEST ('http://www.company.com/')
        FROM DUAL;
```

```
UTL_HTTP.REQUEST ('HTTP://WWW.COMPANY.COM/')
<html>
<head><title>My Company Home Page</title>
<!changed Jan. 16, 19
1 row selected.
```

Si se está utilizando un FIREWALL, se puede incluir el parámetro del proxy de la siguiente forma:

```
SQL> SELECT UTL_HTTP.REQUEST ('http://www.company.com/', )
        'www-proxy.es.company.com') FROM DUAL;
```

Función REQUEST_PIECES

Devuelve una tabla PL/SQL con elementos de 2000 bytes de datos cada uno, recuperados de la URL proporcionada. Se puede especificar un usuario y password para el proxy indicándolo con el siguiente formato:

```
[http://][user[:password]@]host[:port]/
```

SINTAXIS

La sintaxis de esta función es la siguiente:

```
UTL_HTTP.REQUEST_PIECES (
    url                IN VARCHAR2,
    max_pieces         IN NATURAL DEFAULT 32767,
    proxy              IN VARCHAR2 DEFAULT NULL,
    wallet_path        IN VARCHAR2 DEFAULT NULL,
    wallet_password    IN VARCHAR2 DEFAULT NULL)
RETURN html_pieces;
```

DESCRIPCIÓN DE LOS PARÁMETROS DE LA FUNCIÓN

A continuación se describen cada uno de los parámetros que se usan en la definición de la función REQUEST_PIECES.

url

La URL especificada (Uniform Resource Locator).

max_pieces

Parámetro opcional. El máximo número de elementos de la tabla PL/SQL (cada 2000 caracteres en longitud, excepto para el último elemento que puede ser más corto), que deberían devolverse. Si se indica, el valor deberá ser un número de tipo INTEGER positivo.

proxy

Parámetro opcional. Especifica un servidor proxy que se usa cuando se hace la petición http.

wallet_path

Parámetro opcional. Especifica una cartera (wallet) del lado del cliente.

wallet_password

Parámetro opcional. Especifica la password necesaria para abrir la cartera (wallet).

RETURN html_pieces

Devuelve una tabla PL/SQL de tipo UTL_HTTP.HTML_PIECES. Cada elemento de la tabla es una cadena de un máximo de 2000 caracteres. Los elementos devueltos son porciones de caracteres (de 2000 caracteres) sucesivos de la información obtenida de la petición HTTP para la URL indicada.

EXCEPCIONES QUE PUEDE DEVOLVER LA FUNCIÓN

Esta función puede llegar a devolver las siguientes excepciones como errores:

- INIT_FAILED
- REQUEST_FAILED

Procedimiento SET_AUTHENTICATION

Establece la información de autenticación HTTP en la cabecera de la petición HTTP. El servidor Web necesita esta información para autorizar la petición.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.SET_AUTHENTICATION (
    r                IN OUT NOCOPY req,
    username         IN VARCHAR2,
    password         IN VARCHAR2,
    scheme           IN VARCHAR2 DEFAULT 'Basic',
    for_proxy        IN BOOLEAN DEFAULT FALSE);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento SET_AUTHENTICATION.

r

La petición HTTP.

username

El usuario para la autenticación del HTTP.

password

El password para la autenticación del HTTP.

scheme

El esquema de autenticación HTTP. El valor por defecto es 'Basic'.

for_proxy

Identifica si la información de autenticación http es para acceso a un servidor proxy en vez de a un servidor Web. El valor por defecto es FALSE.

Procedimiento SET_AUTHENTICATION_FROM_WALLET

Establece la información de autenticación HTTP en la cabecera de la petición http necesaria para que la petición sea autorizada por el servidor Web usando las credenciales de usuario y password almacenadas en Oracle wallet.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.SET_AUTHENTICATION_FROM_WALLET(  
  r           IN OUT NOCOPY req,  
  alias       IN VARCHAR2,  
  scheme      IN VARCHAR2 DEFAULT 'Basic',  
  for_proxy   IN BOOLEAN DEFAULT FALSE);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento SET_AUTHENTICATION_FROM_WALLET.

r

La petición HTTP.

alias

Alias para identificar y devolver las credenciales del usuario y password almacenadas en Oracle wallet.

scheme

El esquema de autenticación HTTP. El valor por defecto es 'Basic'.

for_proxy

Identifica si la información de autenticación http es para acceso a un servidor proxy en vez de a un servidor Web. El valor por defecto es FALSE.

Procedimiento SET_BODY_CHARSET

Este procedimiento está sobrecargado, lo que supone que permite realizar diferentes funcionalidades dependiendo de la sintaxis que se utilice.

SINTAXIS

La primera sintaxis del procedimiento establece el juego de caracteres por defecto del cuerpo de todas las futuras peticiones HTTP.

```
UTL_HTTP.SET_BODY_CHARSET (
    charset      IN VARCHAR2 DEFAULT NULL);
```

La segunda sintaxis del procedimiento establece el juego de caracteres por defecto del cuerpo de una petición concreta HTTP.

```
UTL_HTTP.SET_BODY_CHARSET (
    r            IN OUT NOCOPY req,
    charset      IN VARCHAR2 DEFAULT NULL);
```

La última sintaxis del procedimiento establece el juego de caracteres por defecto del cuerpo de una respuesta concreta HTTP.

Antolín Muñoz-Chaparro

Jefe de Proyecto-Sistemas Informáticos

División III de Aplicaciones de Costes de Personal y Pensiones Públicas

Oficina de Informática Presupuestaria (DIR3: EA002/952)

Intervención General de la Administración del Estado




```
UTL_HTTP.SET_BODY_CHARSET (
  r           IN OUT NOCOPY resp,
  charset     IN VARCHAR2 DEFAULT NULL);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento SET_BODY_CHARSET.

r

La petición/respuesta HTTP.

charset

El juego de caracteres por defecto del cuerpo de la petición/respuesta. Puede establecer en notación Oracle o IANA (Internet Assigned Numbers Authority). Si este parámetro tiene un valor NULL, se asumirá el juego de caracteres que esté establecido en la base de datos.

Procedimiento SET_COOKIE_SUPPORT

Este procedimiento está sobrecargado, lo que supone que permite realizar diferentes funcionalidades dependiendo de la sintaxis que se utilice.

SINTAXIS

La primera sintaxis del procedimiento habilita o deshabilita el soporte para las cookies HTTP en las peticiones. Use este procedimiento para cambiar la configuración de soporte de las cookies en una petición inherente desde la sesión que por defecto se haya configurado:

```
UTL_HTTP.SET_COOKIE_SUPPORT (
  r           IN OUT NOCOPY REQ,
  enable      IN BOOLEAN DEFAULT TRUE);
```

La segunda sintaxis del procedimiento establece si las peticiones HTTP futuras soportarán cookies, y el máximo número de cookies que serán mantenidas en la sesión actual del usuario establecida contra la base de datos.

```
UTL_HTTP.SET_COOKIE_SUPPORT (
  enable           IN BOOLEAN,
  max_cookies      IN PLS_INTEGER DEFAULT 300,
  max_cookies_per_site IN PLS_INTEGER DEFAULT 20);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento SET_COOKIE_SUPPORT.

r

La petición HTTP.

enable

Para habilitar el soporte de las cookies HTTP se establecerá a valor TRUE. En caso de querer deshabilitarlas se indicará FALSE.

max_cookies

Establece el número total máximo de cookies que se mantendrán en la sesión actual.

max_cookies_per_site

Establece el número máximo de cookies que se mantendrán en la sesión actual para cada sitio Web.

Procedimiento SET_DETAILED_EXCP_SUPPORT

Establece que el paquete UTL_HTTP alcanzará excepciones detalladas. Por defecto UTL_HTTP alcanza la excepción REQUEST_FAILED cuando falla una petición HTTP. Use GET_DETAILED_SQLCODE y GET_DETAILED_SQLERRM para obtener más información sobre el error (excepción) obtenido.

Antolín Muñoz-Chaparro

Jefe de Proyecto-Sistemas Informáticos

División-III de Aplicaciones de Costes de Personal y Pensiones Públicas

Oficina de Informática Presupuestaria (DIR3: EAO027952)

Intervención General de la Administración del Estado



SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.SET_DETAILED_EXCP_SUPPORT (
enable      IN BOOLEAN DEFAULT FALSE);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento SET_DETAILED_EXCP_SUPPORT.

enable

Si el valor es TRUE, pide a UTL_HTTP que alcance una excepción detallada, directamente. En caso contrario establezca el valor a FALSE.

Procedimiento SET_FOLLOW_REDIRECT

Establece el número máximo de veces que el paquete UTL_HTTP seguirá redireccionando la instrucción HTTP en la respuesta a esta petición, o a futuras peticiones. Este procedimiento está sobrecargado.

SINTAXIS

La primera sintaxis del procedimiento se utiliza para establecer el número máximo de redireccionamientos.

```
UTL_HTTP.SET_FOLLOW_REDIRECT (
Max_redirects  IN PLS_INTEGER DEFAULT 3);
```

La segunda sintaxis del procedimiento cambia el número máximo de redireccionamientos de una petición inherente a la sesión por defecto.

```
UTL_HTTP.SET_FOLLOW_REDIRECT (
r              IN OUT NOCOPY req,
max_redirects  IN PLS_INTEGER DEFAULT 3);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento SET_FOLLOW_REDIRECT.

r

La petición HTTP.

max_redirects

El número máximo de redireccionamientos. Establézcalo a valor cero para deshabilitar los redireccionamientos.

Procedimiento SET_HEADER

Establece la cabecera de una petición HTTP. La cabecera de la petición es enviada al servidor Web en cuanto se establece.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.SET_HEADER (
  r           IN OUT NOCOPY req,
  name        IN VARCHAR2,
  value       IN VARCHAR2);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento SET_HEADER.

r

La petición HTTP.

name

El nombre de la cabecera de la petición HTTP.

value

El valor de la cabecera de la petición HTTP.

Procedimiento SET_PERSISTENT_CONN_SUPPORT

Habilita o deshabilita el soporte para conexiones persistentes HTTP 1.1 en las peticiones.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.SET_PERSISTENT_CONN_SUPPORT (
    r           IN OUT NOCOPY req,
    enable      IN BOOLEAN DEFAULT FALSE);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento SET_PERSISTENT_CONN_SUPPORT.

r

La petición HTTP.

enable

Cuando tiene valor TRUE, mantiene las conexiones de red persistentes. En otro caso habrá que indicar valor FALSE.

Procedimiento SET_PROXY

Establece el proxy que se usará en las peticiones del protocolo HTTP o de cualquier otro, excluyendo del uso del proxy, aquellos hosts y dominios que se hayan indicado en el parámetro NO_PROXY_DOMAINS.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.SET_PROXY (
  proxy          IN VARCHAR2,
  no_proxy_domains IN VARCHAR2);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento SET_PROXY.

proxy

El proxy (host y número de puerto opcional) que se usa por el paquete UTL_HTTP.

no_proxy_domains

La lista de los hosts y dominios en los que no se usará ningún proxy para todas las peticiones que vayan dirigidas a los mismos.

Procedimiento SET_RESPONSE_ERROR_CHECK

Establece si se habilita o no la posibilidad de que GET_RESPONSE alcance una excepción cuando el servidor Web devuelva un código de estado que indique un error. El código de estado se encontrará en el rango de valores de 4xx a 5xx. Por ejemplo, cuando la petición URL no encuentra el servidor Web destino, se devuelve un código de estado 404 (documento no encontrado).

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.SET_RESPONSE_ERROR_CHECK (
enable                               IN BOOLEAN DEFAULT FALSE);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento SET_RESPONSE_ERROR_CHECK.

enable

TRUE para habilitar el servicio de respuesta de errores; en otro caso se indicará FALSE.

Procedimiento SET_TRANSFER_TIMEOUT

Establece el tiempo límite de espera por defecto antes de terminar todas las peticiones futuras que el paquete UTL_HTTP debería llevar a cabo mientras lee las respuestas HTTP del servidor Web o del servidor proxy. El valor de tiempo límite de espera puede ser usado para evitar que los programas PL/SQL se queden bloqueados porque el servidor Web esté ocupado o tenga un tráfico de red congestionado mientras recupera las páginas Web de los servidores Web.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.SET_TRANSFER_TIMEOUT (
timeout                             IN PLS_INTEGER DEFAULT 60);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento SET_TRANSFER_TIMEOUT.

timeout

El tiempo límite de espera en las transferencias efectuadas en la red, medido en segundos.

Procedimiento SET_WALLET

Establece la cartera (wallet) Oracle utilizada para todas las peticiones HTTP sobre SSL (Secured Socket Layer), denominadas HTTPS.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.SET_WALLET (
  path    IN VARCHAR2,
  password IN VARCHAR2 DEFAULT NULL);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento SET_WALLET.

path

La ruta que contiene la cartera (wallet) Oracle. El formato es `file:directory-path`.

El formato de WALLET_PATH en un PC es por ejemplo, `file:c:\WINNT\Profiles\username\WALLETS`, y en Unix es por ejemplo, `file:/home/username/wallets`.

Cuando el paquete UTL_HTTP se ejecuta en un servidor de base de datos Oracle, la cartera (wallet) es accesible desde el servidor de base de datos. Por tanto, la ruta de la cartera (wallet) debe ser accesible desde el servidor de base de datos.

password

El password necesario para abrir la cartera (wallet). Si la cartera (wallet) tiene habilitado el sistema auto-login, se debe omitir este parámetro y pasarle un valor NULL.

Antolín-Muñoz-Chaparro

Jefe de Proyecto-Sistemas Informáticos

División-III de Aplicaciones de Costes de Personal y Pensiones Públicas

Oficina de Informática Presupuestaria (DIR3: EAO027952)

Intervención General de la Administración del Estado



Procedimiento WRITE_LINE

Escribe una línea de texto en el cuerpo de la petición HTTP y finaliza la línea con un carácter de nueva línea (CRLF como se definió en UTL_TCP). Los datos del texto son automáticamente convertidos del juego de caracteres de la base de datos al juego de caracteres del cuerpo de la petición.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.WRITE_LINE (  
  r           IN OUT NOCOPY req,  
  data        IN VARCHAR2 CHARACTER SET ANY_CS);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento WRITE_LINE.

r

La petición HTTP.

data

La línea de texto para enviar en el cuerpo de petición HTTP.

Procedimiento WRITE_RAW

Escribe datos binarios en el cuerpo de petición HTTP.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.WRITE_RAW (  
  r           IN OUT NOCOPY req,  
  data        IN RAW);
```

Antolín-Muñoz-Chaparro

Jefe de Proyecto-Sistemas-Informáticos

División-III de Aplicaciones de Costes de Personal y Pensiones Públicas

Oficina de Informática-Presupuestaria (DIR3-EA0027952)

Intervención General de la Administración del Estado



DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento `WRITE_RAW`.

r

La petición HTTP.

data

Los datos binarios a enviar en el cuerpo de petición HTTP.

Procedimiento `WRITE_TEXT`

Escribe datos de texto en el cuerpo de petición HTTP. El texto es convertido automáticamente del juego de caracteres de la base de datos al juego de caracteres del cuerpo de la petición.

SINTAXIS

La sintaxis de este procedimiento es la siguiente:

```
UTL_HTTP.WRITE_TEXT (
  r           IN OUT NOCOPY req,
  data        IN VARCHAR2 CHARACTER SET ANY_CS);
```

DESCRIPCIÓN DE LOS PARÁMETROS DEL PROCEDIMIENTO

A continuación se describen cada uno de los parámetros que se usan en la definición del procedimiento `WRITE_TEXT`.

r

La petición HTTP.

data

Los datos de texto a enviar en el cuerpo de petición HTTP.

