4D for OCI

Introduction

Preface

4D for OCI is a set of 4D external routines that allows a 4D database to communicate with an Oracle® database. Using 4D for OCI, your 4D database can display, manipulate, and modify data stored in an Oracle database.

With 4D for OCI, 4D becomes a front-end to an Oracle database. This configuration allows you to:

- Present a fully customized 4D interface, complete with custom menus and forms.
- Use the data storage and disk access capabilities of Oracle to create a powerful data management system.
- Create multiple 4D databases that access the same Oracle database.

About this manual

This manual details implementing, using, and modifying systems that integrate an Oracle server with a 4D database.

The manual is written for users already familiar with the 4D language as well as with Oracle's SQL language. We recommend that new users familiarize themselves with both products before continuing with this manual.

Cross-Platform

Oracle OCIs are available for Mac as well as for Windows, and the 4D for OCI plug-in is available for both platforms. This plug-in is also available for both 32-bit and 64-bit applications.

Note: Prior to version v15, only 32-bit versions were available for OS X. To ensure complete compatibility with 4D for OCI v15, be sure to use version 11.2.0.4.0 of Oracle OCIs for your installation.

Conventions

In this manual, 4D for OCI commands are printed in camel case, for example: OCIRawPtr.

In addition, all table names are shown in brackets in the text to help distinguish them from the names of fields, forms, and other items. For instance, the "Companies" table is written as the [Companies] table.

Using OCI and supporting SQL*Net

4D for OCI uses Oracle Call Interfaces (OCI) to take full advantage of Oracle features. 4D for OCI requires the OCI driver file provided by Oracle Corporation. This driver file works with Oracle as well as SQL*Net.

Note: The version of Oracle OCI to use with 4D for OCI v15 is 11.2.0.4.0.

Using 4D for OCI with 4D

4D for OCI can be used with either 4D or 4D Server. When used with 4D for OCI, 4D enables you to create a database that can become a client of the Oracle database. Each user with a copy of the database can simultaneously connect to and use the Oracle database.

4D Server enables you to create a multi-developer database application. When used with 4D for OCI, 4D and 4D Server allow multiple developers to connect to an Oracle database. Although the clients are still connected to a server running 4D, they communicate directly with the Oracle server to display and modify data.

In this manual, 4D and 4D Server are both referred to as 4D except when there is a difference between the two products.

Using OCI parameters

The structure of dates is different between 4D and Oracle. Oracle includes the time in date fields while 4D sets a specific variable for time. As a result, when passing OCI date parameters, the Oracle dates were divided into date and time.

Example

The **OCIDateToText** command initially accepts a date (*date*) as the second parameter and then its format (*fmt*) as the third parameter.
As for 4D for OCI, the date (*date*) parameter becomes two different parameters for date and time and the *format* parameter becomes the fourth parameter.

Once a text parameter is passed, the OCIs expect a parameter specifying the length of the text. 4D for OCI handles the passage of this parameter, which means that the developer no longer has to pass it.

Example

The **OCILogon** command initially accepts the user name (*username*) as the third parameter and its length (*uname_len*) as the fourth parameter. In this case, 4D for OCI frees the developer from passing the *length* parameter. Thus, the fourth parameter becomes the password (*password*).

Installation

In order for the 4D for OCI plug-in to function properly, you first need to install the files (libraries) needed to run the Oracle Call Interface (OCI).

On Windows

In just a few simple steps, you can download the Oracle Instant Client and install it.

1. Download the appropriate *instantclient-basic-[...]* package for your platform here:
   http://www.oracle.com/technetwork/database/features/instant-client/index-097480.html
   Notes:
   - The version installed (32- or 64-bit) must match the version of 4D used for 4D for OCI.
   - For v15 of 4D for OCI, the files to download are: *instantclient-basic-nt-11.2.0.4.0.zip* or *instantclient-basic-windows.x64-11.2.0.4.0.zip* (for Windows 32-bit or 64-bit, respectively).
2. Unzip the packages into a single directory (e.g., C:¥oracle¥) and rename the subfolder as "instantclient" to get a path such as: C:¥oracle¥instantclient
3. In the environment variables:
   - Add the ORACLE_HOME variable and set it to the path defined in step 2 (e.g.: C:¥oracle¥instantclient)
   - Edit the PATH variable and add this same path at the beginning
4. Add a "network" subfolder to this path with an "admin" subfolder (e.g.: C:¥oracle¥instantclient¥network¥admin)
5. In this "admin" subfolder, create a "tnsnames.ora" file with the parameters to your Oracle server. For example:

```
XE =
 (DESCRIPTION =
   (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.10.10)(PORT = 1521))
  (CONNECT_DATA =
   (SERVER = DEDICATED)
   (SERVICE_NAME = XE)
  )
 )
```

On OS X

Here are the steps to install the Oracle Instant Client and make it work with 4D for OCI:

1. Download the appropriate *instantclient-basic-[...]* package for your platform here:
   http://www.oracle.com/technetwork/topics/intel-macsoft-096467.html
   Notes:
   - The version installed (32- or 64-bit) must match the version of 4D used for 4D for OCI.
   - For v15 of 4D for OCI, the files to download are: *instantclient-basic-macos.x32-11.2.0.4.0.zip* or *instantclient-basic-macos.x64-11.2.0.4.0.zip* (for 32- or 64-bit versions respectively).
2. Copy the required Oracle library to the proper location:
   - Copy all the dylib files from instantclient-basic-[…].zip archive to /usr/local/lib (for example). The dylib files can be copied to one of the following paths:
     - $(HOME)/lib
     - /usr/local/lib
     - /lib

        - /usr/lib
    - Execute the command 'cd /usr/local/lib && sudo ln -sf libclntsh.dylib.11.1 libclntsh.dylib'.
    - Execute the command 'chmod 777 /usr/local/lib/libclntsh.dylib'.

    Note: Since 4D for OCI was designed with version 11.2.0.4.0, in order to avoid problems, we link it to “libclntsh.dylib” instead of the versioned file name.

    Note for El Capitan version: Only the $(HOME)/lib and /usr/local/lib directories are not restricted under El Capitan.
3. Use a text editor to create a “/etc/tnsnames.ora” file and type entries like the example below:

```
oracle4d =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL =TCP)(HOST = 10.96.0.61)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = oracle4d)
    )
  )
```

Note: Oracle does not provide a universal binary download so if you want to use both versions (32- and 64-bits), you have to download them separately and build your own universal binary, for example using the shell ditto command.

Tip: If you experience connection failure, you may want to use SQLPlus from Oracle. You can download the appropriate version (instantclient-sqlplus-macos.x32-11.2.0.4.0.zip or instantclient-sqlplus-macos.x64-11.2.0.4.0.zip) here: http://www.oracle.com/technetwork/topics/intel-macsoft-096467.html

Summary of commands

The tables below list the 4D for OCI commands by theme. Syntax information is also provided for each command on a separate page. (Click on the command name to go directly to its corresponding page in this manual).

For more detailed information about the use of these commands, please refer to their equivalent OCI command described in the OCI documentation provided by the Oracle corporation.

Note: The **Extras** theme is composed of 4D commands that do not have equivalents in the OCI documentation.

Connection theme

Command name	Purpose
OCIBreak	Carries out an immediate asynchronous break
OCILogoff	Releases a session retrieved using **OCILogon**.
OCILogon	Simplified single-session logon
OCIParamGet	Gets parameter descriptor
OCIParamSet	Sets parameter descriptor in COR handle
OCIPasswordChange	Changes password
OCIReset	Resets asynchronous operation and protocol (after call to **OCIBreak**)
OCIServerAttach	Attaches to server; initializes server context handle
OCIServerDetach	Detaches from server; uninitializes server context handle
OCISessionBegin	Authenticates user
OCISessionEnd	Terminates user session

Relational commands theme

Command name	OCI command name (if different)	Purpose
OCIBindDateByName	OCIBindByName	Binds dates by name
OCIBindDateByPos	OCIBindByPos	Binds dates by position
OCICollAppend		Collection appends element
OCICollAssign		Assigns collection
OCICollAssignElem		Collection assigns element
OCICollGetElem		Gets pointer to an element
OCICollMax		Returns maximum number of elements in collection
OCICollSize		Gets current size of collection (in number of elements)
OCICollTrim		Trims elements from the collection
OCIDateAddDays		Adds or subtracts days
OCIDateAddMonths		Adds or subtracts months
OCIDateFromText		Converts string to date
OCIDateLastDay		Gets date of last day of month
OCIDateNextDay		Gets date of next day
OCIDateSysDate		Gets current system date and time
OCIDateToText		Converts date to string
OCIDateZoneToZone		Converts date from one time zone to another zone
OCIDefineDateByPos	OCIDefineByPos	Defines output variable association
OCIIterCreate		Creates iterator to scan the array elements
OCIIterDelete		Deletes iterator
OCIIterGetCurrent		Gets current collection element
OCIIterInit		Initializes iterator to scan the given collection
OCIIterNext		Gets next collection element
OCIIterPrev		Gets previous collection element
OCIRawAllocSize		Gets allocated size of raw memory in bytes
OCIRawAssignBytes		Assigns raw bytes to raw
OCIRawAssignRaw		Assigns raw to raw
OCIRawPtr		Gets raw data Pointer
OCIRawResize		Resizes memory of variable-length raw
OCIRawSize		Gets raw size
OCIRefAssign		Assigns one Ref to another
OCIRefClear		Clears or nullifies Ref
OCIRefFromHex		Converts hexadecimal string to Ref
OCIRefHexSize		Returns size of hexadecimal representation of Ref
OCIRefIsEqual		Compares two Refs for equality
OCIRefIsNull		Tests whether Ref is Null
OCIRefToHex		Converts Ref to hexadecimal string
OCITableDelete		Deletes element
OCITableExists		Tests whether element exists
OCITableFirst		Returns first index of table
OCITableLast		Returns last index of table
OCITableNext		Returns next available index of table
OCITablePrev		Returns previous available index of table
OCITableSize		Returns current size of table

Types theme

Command name	Purpose
OCICacheFlush	Flushes modified persistent objects in cache to server
OCICacheFree	Frees objects in the cache
OCICacheRefresh	Refreshes pinned persistent objects
OCICacheUnmark	Unmarks objects in the cache
OCICacheUnpin	Unpins persistent objects in cache or connection

Datatype theme

Command name	OCI command name (if different)	Purpose
OCIAttrGetText	OCIAttrGet	Gets value for attribute of a handle
OCIAttrGetVal	OCIAttrGet	Gets value for attribute of a handle
OCIAttrSetText	OCIAttrSet	Sets value for attribute of a handle or descriptor
OCIAttrSetVal	OCIAttrSet	Sets value for attribute of a handle or descriptor
OCIBindByName		Binds by name
OCIBindByPos		Binds by position
OCIDefineByPos		Defines output variable association
OCIDescribeAnyText	OCIDescribeAny	Describes existing schema objects
OCIDescribeAnyVal	OCIDescribeAny	Describes existing schema objects
OCIDescriptorAlloc		Allocates and initializes descriptor or LOB locator
OCIDescriptorFree		Frees previously-allocated descriptor
OCIEnvCreate		Creates and initializes an OCI environment
OCIErrorGet		Returns error message and Oracle error
OCIHandleAlloc		Allocates and initializes handle
OCIHandleFree		Frees previously-allocated handle
OCIServerVersion		Gets Oracle version string
OCIStmtExecute		Sends statements to server for execution
OCIStmtFetch		Fetches rows from a query
OCIStmtGetBindInfo		Gets bind and indicator variable names and handle
OCIStmtPrepare		Prepares SQL or PL/SQL statement for execution
OCITerminate		Detaches from shared memory subsystem

Transactions theme

Command name	Purpose
OCITransCommit	Commits transaction on service context
OCITransDetach	Detaches transaction from service context
OCITransForget	Forgets prepared global transaction
OCITransPrepare	Prepares global transaction for commit
OCITransRollback	Rolls back transaction
OCITransStart	Starts transaction on a service context

LOB theme

Command name	Purpose
OCIDurationBegin	Starts user duration for temporary LOB
OCIDurationEnd	Ends user duration for temporary LOB
OCILobAppend	Appends one LOB to another
OCILobAssign	Assigns one LOB locator to another
OCILobCharSetForm	Gets character set form from LOB locator
OCILobCharSetId	Gets character set ID from LOB locator
OCILobCopy	Copies all or part of one LOB to another
OCILobCreateTemporary	Creates temporary LOB
OCILobDisableBuffering	Turns off LOB buffering
OCILobEnableBuffering	Turns on LOB buffering
OCILobErase	Erases portion of LOB
OCILobFileClose	Closes previously-opened BFILE
OCILobFileCloseAll	Closes all previously-opened files
OCILobFileExists	Checks whether file exists on server
OCILobFileGetName	Gets directory object and file name from LOB locator
OCILobFileIsOpen	Checks whether LOB is open
OCILobFileOpen	Opens BFILE
OCILobFileSetName	Sets directory object and file name in LOB locator
OCILobFlushBuffer	Flushes LOB buffer
OCILobFreeTemporary	Frees temporary LOB
OCILobGetChunkSize	Gets chunk size of LOB
OCILobGetLength	Gets length of LOB
OCILobIsEqual	Compares two LOB locators for equality
OCILobIsTemporary	Determines if given LOB is temporary
OCILobLoadFromFile	Loads LOB from BFILE
OCILobLocatorIsInit	Checks to see if LOB locator is initialized
OCILobRead	Reads portion of LOB
OCILobTrim	Truncates LOB
OCILobWrite	Writes into LOB
OCILobWriteAppend	Writes data beginning at end of a LOB

Math theme

Command name	Purpose
OCINumberAdd	Adds numbers
OCINumberArcCos	Computes arc cosine
OCINumberArcSin	Computes arc sine
OCINumberArcTan	Computes arc tangent
OCINumberArcTan2	Computes arc tangent of two numbers
OCINumberCos	Computes cosine
OCINumberDiv	Divides two numbers
OCINumberExp	Raises e to specified Oracle number power
OCINumberFromText	Converts string to Oracle number
OCINumberHypCos	Computes hyperbolic cosine
OCINumberHypSin	Computes hyperbolic sine
OCINumberHypTan	Computes hyperbolic tangent
OCINumberIntPower	Raises given base to an integer power
OCINumberLn	Computes natural logarithm
OCINumberLog	Computes logarithm to arbitrary base
OCINumberMul	Multiplies numbers
OCINumberPower	Exponentiation to base e
OCINumberRound	Rounds Oracle number to specified decimal place
OCINumberSin	Computes sine
OCINumberSqrt	Computes square root of number
OCINumberSub	Subtracts numbers
OCINumberTan	Computes tangent
OCINumberToText	Converts Oracle number to string
OCINumberTrunc	Truncates Oracle number at specified decimal place

Extras theme

Command name	Purpose
_o_OCICleanUp	Frees up memory used by objects
OCIGetTnsnamesPath	Returns full path of Oracle home folder
OCIOnErrCall	Installs error-handling method

Note: These are 4D commands with no equivalent Oracle version.

OCIBreak

OCIBreak (hndlp ; errhp) -> Function result

Parameter	Type		Description
hndlp	Longint	⇒ ⇐	Server or service context handle
errhp	Longint	⇒ ⇐	Error handle
Function result	Longint	↩	Status

Description

Please refer to the **OCIBreak** command in the OCI documentation provided by Oracle.

Summary of constants

The following tables list 4D for OCI constants grouped by theme. For detailed information about these constants, please refer to the OCI documentation provided by the Oracle corporation.

OCI_ATTR theme

Constant	Value	Comment
OCI_ATTR_AGENT_ADDRESS	65	Protocol-specific address of recipient
OCI_ATTR_AGENT_NAME	64	Name of message producer or consumer
OCI_ATTR_AGENT_PROTOCOL	66	Protocol to interpret address and propagate message
OCI_ATTR_ALLOC_DURATION	37	Sets value of allocation duration
OCI_ATTR_APPCTX_ATTR	276	Specifies attribute name of externally initialized context
OCI_ATTR_APPCTX_LIST	274	Gets application context list descriptor
OCI_ATTR_APPCTX_NAME	275	Specifies namespace of externally initialized context
OCI_ATTR_APPCTX_SIZE	273	Initializes externally initialized context array size with number of attributes
OCI_ATTR_APPCTX_VALUE	277	Specifies value of externally initialized context
OCI_ATTR_ATTEMPTS	59	Specifies number of attempts made to dequeue message
OCI_ATTR_AUTOCOMMIT_DDL	271	Specifies if autocommit mode required for DDL statements
OCI_ATTR_BUF_ADDR	76	Buffer address of beginning of stream data
OCI_ATTR_BUF_SIZE	77	Size of stream data in bytes
OCI_ATTR_CACHE	115	Number of sequence numbers cached
OCI_ATTR_CACHE_CLIENT_CONTEXT	251	Cache for user constructed instances
OCI_ATTR_CACHE_MAX_SIZE	35	Sets maximum client-side object cache size as a percentage of optimal size
OCI_ATTR_CACHE_OPT_SIZE	34	Sets optimal client-side object cache size in bytes
OCI_ATTR_CATALOG_LOCATION	268	Position of catalog in table
OCI_ATTR_CHAR_COUNT	15	Sets number of characters in character type data
OCI_ATTR_CHARSET	20	Character set value
OCI_ATTR_CHARSET_FORM	32	Character set form of bind handle
OCI_ATTR_CHARSET_ID	31	Character set ID of bind handle
OCI_ATTR_CLUSTERED	105	Indicates table is clustered
OCI_ATTR_COL_COUNT	82	Last column of last row processed
OCI_ATTR_COLLECTION_ELEMENT	227	Handle to collection element
OCI_ATTR_COLLECTION_TYPECODE	217	Type code of collection
OCI_ATTR_COMPOBJ_COLL_OUTOFLIN	53	Indicates whether to fetch collection attributes in out-of-line type object
		(refer to OCI_ATTR_COMPLEXOBJECT_COLL_OUTOFLINE)
OCI_ATTR_COMPOBJ_LEVEL	52	Depth level for complex object retrieval
		(refer to OCI_ATTR_COMPLEXOBJECT_LEVEL)
OCI_ATTR_COMPOBJCOMP_TYP	50	Type of Ref to follow for complex object retrieval
		(refer to OCI_ATTR_COMPLEXOBJECTCOMP_TYPE)
OCI_ATTR_COMPOBJCOMP_TYP_LEVEL	51	Depth level for following Refs
		(refer to OCI_ATTR_COMPLEXOBJECTCOMP_TYPE_LEVEL)
OCI_ATTR_CONSUMER_NAME	50	Name of consumer
OCI_ATTR_CORRELATION	58	Specifies identifier for message
OCI_ATTR_CURSOR_COMMIT_BEHAVIOR	266	Specifies effect of Commit on cursors
OCI_ATTR_DATA_SIZE	1	Maximum size in bytes of external data for column
OCI_ATTR_DATA_TYPE	2	Gets or sets external data type of column
OCI_ATTR_DATEFORMAT	75	Date conversion mask for column
OCI_ATTR_DELAY	56	Specifies number of seconds to delay message

OCI_ATTR_DEQ_MODE	51	Specifies locking behavior associated with dequeue
OCI_ATTR_DEQ_MSGID	54	Specifies identifier of message to be dequeued
OCI_ATTR_DESC_PUBLIC	250	Specifies object name to be looked up as a public synonym
OCI_ATTR_DIRPATH_FILE	139	Database file to load into
OCI_ATTR_DIRPATH_INDEX_MAINT_METHOD	138	Performs index row insertion
OCI_ATTR_DIRPATH_MODE	78	Mode of direct path context
OCI_ATTR_DIRPATH_NOLOG	79	NOLOG attribute determines whether image redo or invalidation redo is generated
OCI_ATTR_DIRPATH_PARALLEL	80	Allows same direct path segment to be loaded concurrently
OCI_ATTR_DIRPATH_SORTED_INDEX	137	Index that data is sorted on
OCI_ATTR_DIRPATH_STORAGE_INITI	140	Initial extent size
OCI_ATTR_DIRPATH_STORAGE_NEXT	141	Next extent size
OCI_ATTR_DISP_NAME	100	Display name
OCI_ATTR_DISP_SIZE	3	Display size
OCI_ATTR_DML_ROW_OFFSET	74	Returns offset (into DML array) where error occurred
OCI_ATTR_DURATION	132	Duration of temporary table
OCI_ATTR_ENCAPSULATION	235	Encapsulation level of method
OCI_ATTR_ENQ_TIME	62	Specifies time message was enqueued
OCI_ATTR_ENV	5	Returns environment context associated with server context
OCI_ATTR_EXCEPTION_QUEUE	61	Specifies name of exception queue
OCI_ATTR_EXPIRATION	57	Specifies expiration of message
OCI_ATTR_EXTERNAL_NAME	26	Specifies external global database name
OCI_ATTR_FDO	39	Format Descriptor object attribute
OCI_ATTR_FNCODE	1	Returns OCI function code
OCI_ATTR_FOCBK	43	Sets failover callback
OCI_ATTR_FSPRECISION	0	Specifies fractional seconds precision of datetime or interval
OCI_ATTR_HAS_DEFAULT	212	Indicates whether argument has a default
OCI_ATTR_HAS_FILE	226	Type contains a BFILE attribute
OCI_ATTR_HAS_LOB	225	Type contains a LOB attribute
OCI_ATTR_HAS_NESTED_TABLE	224	Type contains a nested table attribute
OCI_ATTR_HAS_RESULT	240	Indicates whether argument has a result
OCI_ATTR_HEAPALLOC	30	Current size of memory allocated from environment handle
OCI_ATTR_HW_MARK	117	High-water mark (in Number format)
OCI_ATTR_IN_V8_MODE	44	Determines whether application has switched to Oracle release 7 mode
OCI_ATTR_INCR	114	Increment (in Oracle Number format)
OCI_ATTR_INDEX_ONLY	107	Indicates table is index-only
OCI_ATTR_INITIAL_CLIENT_ROLES	100	Specifies initial client role(s) when the application server connects
OCI_ATTR_INTERNAL_NAME	25	Sets client database name recorded when performing global transactions
OCI_ATTR_IOMODE	213	Indicates the argument mode (IN, OUT, IN/OUT)
OCI_ATTR_IS_CONSTANT	239	Indicates method is a constant
OCI_ATTR_IS_CONSTRUCTOR	241	Indicates method is a constructor
OCI_ATTR_IS_DESTRUCTOR	242	Indicates method is a destructor

OCI_ATTR_IS_INCOMPLETE_TYPE	219	Indicates an incomplete type
OCI_ATTR_IS_INLINE	238	Indicates method is inline
OCI_ATTR_IS_INVOKER_RIGHTS	133	Indicates procedure or function has invoker's rights
OCI_ATTR_IS_MAP	244	Indicates method is a map method
OCI_ATTR_IS_NULL	7	Indicates whether null values are permitted for column
OCI_ATTR_IS_OPERATOR	243	Indicates method is an operator
OCI_ATTR_IS_ORDER	245	Indicates method is an order method
OCI_ATTR_IS_PREDEFINED_TYPE	221	Indicates a predefined type
OCI_ATTR_IS_RNDS	246	Indicates "Read No Data State" is set for method
OCI_ATTR_IS_RNPS	247	Indicates "Read No Process State" is set for method
OCI_ATTR_IS_SELFISH	236	Indicates method is selfish
OCI_ATTR_IS_SUBTYPE	258	Indicates a subtype
OCI_ATTR_IS_SYSTEM_GENERATED_TYPE	223	Indicates a system-generated type
OCI_ATTR_IS_SYSTEM_TYPE	220	Indicates a system type
OCI_ATTR_IS_TEMPORARY	130	Indicates table is temporary
OCI_ATTR_IS_TRANSIENT_TYPE	222	Indicates a transient type
OCI_ATTR_IS_TYPED	131	Indicates table is typed
OCI_ATTR_IS_VIRTUAL	237	Indicates method is virtual
OCI_ATTR_IS_WNDS	248	Indicates "Write No Data State" is set for method
OCI_ATTR_IS_WNPS	249	Indicates "Write No Process State" is set for method
OCI_ATTR_LEVEL	211	Data type levels
OCI_ATTR_LFPRECISION	0	Leading field precision of interval
OCI_ATTR_LINK	111	Database link name of synonym translation
OCI_ATTR_LIST_ARGUMENTS	108	List of arguments at next level
OCI_ATTR_LIST_COLUMNS	103	Column list
OCI_ATTR_LIST_OBJECTS	261	List of objects in schema
OCI_ATTR_LIST_SCHEMAS	263	List of schemas in database
OCI_ATTR_LIST_SUBPROGRAMS	109	Subprogram list
OCI_ATTR_LIST_TYPE_ATTRS	229	List of type attributes
OCI_ATTR_LIST_TYPE_METHODS	231	List of type methods
OCI_ATTR_LOBEMPTY	45	Sets internal LOB locator to empty
OCI_ATTR_LOCKING_MODE	272	Locking mode for database
OCI_ATTR_LTYPE	128	Specifies list type
OCI_ATTR_MAP_METHOD	232	Map method of type
OCI_ATTR_MAX	113	Maximum value (in Oracle Number format)
OCI_ATTR_MAX_CATALOG_NAMELEN	267	Maximum length of catalog (database) name
OCI_ATTR_MAX_COLUMN_LEN	265	Maximum length of column name
OCI_ATTR_MAX_PROC_LEN	264	Maximum length of procedure name
OCI_ATTR_MAXDATA_SIZE	33	Sets maximum number of bytes allowed in buffer on server side
OCI_ATTR_MEMPOOL_APPNAME	90	Name or pathname of executable
OCI_ATTR_MEMPOOL_HOMENAME	91	Directory name where executables using same instance are located
OCI_ATTR_MEMPOOL_INSTNAME	89	User-defined name to identify an instance of shared subsystem
OCI_ATTR_MEMPOOL_MODEL	92	Model of shared pool
OCI_ATTR_MEMPOOL_SIZE	88	Size of shared pool in bytes
OCI_ATTR_MIGSESSION	86	Specifies session identified for session handle
OCI_ATTR_MIN	112	Minimum value (in Oracle Number format)

OCI_ATTR_MODES	93	Modes
OCI_ATTR_MSG_PROP	72	Message properties
OCI_ATTR_MSG_STATE	63	Specifies state of message at time of dequeue
OCI_ATTR_NAME	4	Name of table to be loaded into
OCI_ATTR_NAVIGATION	52	Specifies position of message retrieved
OCI_ATTR_NCHAR	21	NCHAR type
OCI_ATTR_NCHARSET_ID	262	Database national character set ID from server handle
OCI_ATTR_NESTED_PREFETCH_MEMOR	14	Memory limit for nested rows
OCI_ATTR_NESTED_PREFETCH_ROWS	12	Prefetch rows of nested table
OCI_ATTR_NFY_MSGID	71	Message ID
OCI_ATTR_NOCACHE	87	Specifies whether temporary LOB is read into buffer cache of server
OCI_ATTR_NONBLOCKING_MODE	3	Determines blocking mode
OCI_ATTR_NOWAIT_SUPPORT	270	Determines whether database supports NOWAIT clause
OCI_ATTR_NUM_ARGS	215	Total number of arguments
OCI_ATTR_NUM_ATTRS	120	Number of attributes
OCI_ATTR_NUM_COLS	102	Number of columns
OCI_ATTR_NUM_DML_ERRORS	73	Returns number of errors in DML operation
OCI_ATTR_NUM_ELEMS	234	Number of elements in an array
OCI_ATTR_NUM_PARAMS	121	Number of elements in list
OCI_ATTR_NUM_ROWS	81	Number of rows loaded/to be allocated
OCI_ATTR_NUM_TYPE_ATTRS	228	Number of type attributes
OCI_ATTR_NUM_TYPE_METHODS	230	Number of type methods
OCI_ATTR_OBJ_ID	136	Object or schema ID
OCI_ATTR_OBJ_NAME	134	Database name or object name in schema
OCI_ATTR_OBJ_SCHEMA	135	Schema name where object is located
OCI_ATTR_OBJECT	2	Returns True if environment initialized in object mode
OCI_ATTR_OBJID	122	Object ID
OCI_ATTR_ORDER	116	Specifies whether sequence is ordered
OCI_ATTR_ORDER_METHOD	233	Order method of type
OCI_ATTR_ORIGINAL_MSGID	69	ID of message in last queue that generated this message
OCI_ATTR_OVERLOAD	210	Specifies whether position is overloaded
OCI_ATTR_OVERLOAD_ID	125	Overloading ID number
OCI_ATTR_PARAM	124	Points to root of description
OCI_ATTR_PARAM_COUNT	18	Returns number of parameters in describe handle
OCI_ATTR_PARSE_ERROR_OFFSET	129	Returns parse error offset for statement
OCI_ATTR_PARTITIONED	106	Indicates table is partitioned
OCI_ATTR_PASSWORD	23	Specifies password to use for authentication
OCI_ATTR_PDPRC	17	Specifies packed decimal precision
OCI_ATTR_PDSCL	16	Specifies scale for packed decimal values
OCI_ATTR_PIN_DURATION	38	Sets default value for pin durations
OCI_ATTR_PINOPTION	36	Sets pin default value
OCI_ATTR_POSITION	11	Position of argument in list
OCI_ATTR_POSTPROCESSING_CALLBA	40	Callback to process outbind data
OCI_ATTR_POSTPROCESSING_CONTEX	41	Callback context to process outbind data
OCI_ATTR_PRECISION	5	Gets or sets precision
OCI_ATTR_PREFETCH_MEMORY	13	Sets memory level for top-level rows to be prefetched
OCI_ATTR_PREFETCH_ROWS	11	Sets number of top-level rows to be prefetched

OCI_ATTR_PRIORITY	55	Specifies message priority
OCI_ATTR_PROXY_CREDENTIALS	99	Specifies use of application server credentials for proxy authentication
OCI_ATTR_PTYPE	123	Type of information described by parameter
OCI_ATTR_QUEUE_NAME	70	Queue name of notification
OCI_ATTR_RADIX	214	Returns radix (if Number type)
OCI_ATTR_RDBA	104	Data block address of segment header
OCI_ATTR_RECIPIENT_LIST	60	List of recipients (queue subscribers by default)
OCI_ATTR_REF_TDO	110	Returns Ref of type descriptor object (TDO)
OCI_ATTR_RELATIVE_MSGID	48	Specifies identifier of message in sequence deviation operation
OCI_ATTR_ROW_COUNT	9	Returns number of rows successfully processed
OCI_ATTR_ROWID	19	Returns ROWID descriptor
OCI_ATTR_ROWS_RETURNED	42	Number of rows to be returned in current iteration
OCI_ATTR_SAVEPOINT_SUPPORT	269	Specifies whether database supports savepoints
OCI_ATTR_SCALE	6	Scale of numeric type attributes
OCI_ATTR_SCHEMA_NAME	9	Returns schema name
OCI_ATTR_SENDER_ID	68	Identifies original sender of message
OCI_ATTR_SEQUENCE_DEVIATION	49	Specifies whether message should be dequeued before other message(s) already in queue
OCI_ATTR_SERVER	6	Gets or sets server context attribute
OCI_ATTR_SERVER_GROUP	85	Specifies server group
OCI_ATTR_SERVER_STATUS	143	Returns current status of server handle
OCI_ATTR_SESSION	7	Gets or sets authentication context attribute
OCI_ATTR_SESSLANG	46	Session language handle
OCI_ATTR_SHARED_HEAPALLOC	84	Returns size of memory currently allocated from shared pool
OCI_ATTR_SQLCODE	4	Returns code of last SQL command processed
OCI_ATTR_SQLFNCODE	10	Returns function code of SQL command associated with statement
OCI_ATTR_STATEMENT	144	Returns text of SQL statement
OCI_ATTR_STMT_TYPE	24	Type of statement associated with handle
OCI_ATTR_STREAM_OFFSET	83	Offset into stream buffer of last processed row
OCI_ATTR_SUB_NAME	10	Name of partition or subpartition to be loaded
OCI_ATTR_SUBSCR_CALLBACK	95	Subscription callback
OCI_ATTR_SUBSCR_CTX	96	Context to pass to user callback
OCI_ATTR_SUBSCR_NAME	94	Subscription name
OCI_ATTR_SUBSCR_NAMESPACE	98	Namespace where subscription handle is used
OCI_ATTR_SUBSCR_PAYLOAD	97	Buffer that corresponds to payload sent with notification
OCI_ATTR_SUPERTYPE_NAME	260	Name of supertype
OCI_ATTR_SUPERTYPE_SCHEMA_NAME	259	Name of schema containing supertype
OCI_ATTR_TABLESPACE	126	Tablespace where table resides
OCI_ATTR_TDO	127	Sets object or collection attribute type
OCI_ATTR_TIMESTAMP	119	Timestamp of object
OCI_ATTR_TRANS	8	Gets or sets transaction context attribute
OCI_ATTR_TRANS_LOCK	28	Locks transaction
OCI_ATTR_TRANS_NAME	29	Gets or sets text string which identifies transaction
OCI_ATTR_TRANS_TIMEOUT	142	Gets or sets timeout interval value

OCI_ATTR_TYPE_NAME	8	Type name
OCI_ATTR_TYPE_SCHEMA	118	Schema name of type
OCI_ATTR_TYPECODE	216	Type code
OCI_ATTR_UCI_CONSTRUCT	252	Construct user constructed instance
OCI_ATTR_UCI_COPY	254	Copy user constructed instance
OCI_ATTR_UCI_DESTRUCT	253	Destroy user constructed instance
OCI_ATTR_UCI_PICKLE	255	Pickle user constructed instance
OCI_ATTR_UCI_REFRESH	257	Refresh user constructed instance
OCI_ATTR_UCI_UNPICKLE	256	Unpickle user constructed instance
OCI_ATTR_UNK	101	Unknown attribute
OCI_ATTR_USERNAME	22	Specifies user name for authentication
OCI_ATTR_VERSION	218	Database version
OCI_ATTR_VISIBILITY	47	Specifies transactional behavior for message
OCI_ATTR_WAIT	53	Specifies wait time if no message matching search criteria is currently available
OCI_ATTR_XID	27	Gets or sets XID which identifies transaction

OCI_DATA_TYPE theme

Constant	Value	Comment
SQLT_AFC	96	Specifies CHAR data type
SQLT_AVC	97	Specifies CHARZ data type
SQLT_BDOUBLE	22	Specifies BINARY_DOUBLE (native double) data type
SQLT_BFILE	114	Specifies binary FILE LOB data type
SQLT_BFILEE	114	Specifies binary file LOB data type
SQLT_BFLOAT	21	Specifies BINARY_FLOAT (native float) data type
SQLT_BIN	23	Specifies RAW data type
SQLT_BLOB	113	Specifies binary LOB descriptor data type
SQLT_CFILE	115	Specifies character file LOB data type
SQLT_CFILEE	115	Specifies character file LOB data type
SQLT_CHR	1	Specifies VARCHAR2 data type
SQLT_CLOB	112	Specifies character LOB data type
SQLT_CUR	102	Cursor type
SQLT_DAT	12	Specifies DATE data type
SQLT_DATE	184	Specifies ANSI DATE descriptor data type
SQLT_FILE	114	Specifies binary FILE descriptor data type
SQLT_FLT	4	Specifies FLOAT data type
SQLT_INT	3	Specifies INTEGER data type
SQLT_INTERVAL_DS	190	Specifies INTERVAL DAY TO SECOND descriptor data type
SQLT_INTERVAL_YM	189	Specifies INTERVAL YEAR TO MONTH data type
SQLT_LAB	105	MLSLABEL data type
SQLT_LBI	24	Specifies LONG RAW data type
SQLT_LNG	8	Specifies LONG data type
SQLT_LVB	95	Specifies LONG VARRAW data type
SQLT_LVC	94	Specifies LONG VARCHAR data type
SQLT_NCO	122	Specifies COLLECTION data type
SQLT_NON	10	Specifies unknown (none) data type
SQLT_NTY	108	Specifies named data type
SQLT_NUM	2	Specifies NUMBER data type
SQLT_ODT	156	Specifies OCI DATE data type
SQLT_OSL	106	Specific OS label type
SQLT_PDN	7	Specifies Packed Decimal Numeric (PDN) data type
SQLT_RDD	104	Specifies ROWID descriptor data type
SQLT_REF	110	Specifies REF data type
SQLT_RID	11	Specifies ROWID data type
SQLT_RSET	116	Specifies result set data type
SQLT_SLS	91	Specifies Sign Leading Separate (SLS) data type
SQLT_STR	5	Specifies NULL-terminated STRING data type
SQLT_TIME	185	Specifies TIME data type
SQLT_TIME_TZ	186	Specifies TIME WITH TIME ZONE data type
SQLT_TIMESTAMP	187	Specifies TIMESTAMP descriptor data type
SQLT_TIMESTAMP_LTZ	232	Specifies TIMESTAMP WITH LOCAL TIME ZONE descriptor data type
SQLT_TIMESTAMP_TZ	188	Specifies TIMESTAMP WITH TIME ZONE data type
SQLT_UIN	68	Specifies UNSIGNED data type
SQLT_VBI	15	Specifies VARRAW data type
SQLT_VCS	9	Specifies VARCHAR data type
SQLT_VNU	6	Specifies VARNUM data type

SQLT_VST	155	Specifies OCI STRING data type

OCI_DTYPE theme

Constant	Value	Comment
OCI_DTYPE_AQAGENT	60	Specifies generation of Advanced Queuing agent descriptor
OCI_DTYPE_AQDEQ_OPTIONS	58	Specifies generation of Advanced Queuing dequeue options descriptor
OCI_DTYPE_AQENQ_OPTIONS	57	Specifies generation of Advanced Queuing enqueue options descriptor
OCI_DTYPE_AQMSG_PROPERTIES	59	Specifies generation of Advanced Queuing message properties descriptor
OCI_DTYPE_AQNFY_DESCRIPTOR	64	Specifies generation of Advanced Queuing notification descriptor (refer to OCI_DTYPE_AQNFY)
OCI_DTYPE_COMPLEXOBJECTCOMP	55	Specifies generation of complex object retrieval descriptor
OCI_DTYPE_DATE	65	Specifies generation of ANSI DATE descriptor
OCI_DTYPE_FILE	56	Specifies generation of FILE value type locator
OCI_DTYPE_FIRST	50	Specifies first descriptor type
OCI_DTYPE_INTERVAL_DS	63	Specifies generation of INTERVAL DAY TO SECOND descriptor
OCI_DTYPE_LAST	71	Specifies last descriptor type
OCI_DTYPE_LOB	50	Specifies generation of LOB value type locator
OCI_DTYPE_LOCATOR	61	Specific descriptor type locator
OCI_DTYPE_PARAM	53	Specifies generation of read-only parameter descriptor
OCI_DTYPE_ROWID	54	Specifies generation of ROWID descriptor
OCI_DTYPE_RSET	52	Specifies result set descriptor
OCI_DTYPE_SNAP	51	Specifies generation of snapshot descriptor
OCI_DTYPE_TIME	66	Specifies generation of TIME descriptor
OCI_DTYPE_TIME_TZ	67	Specifies generation of TIME WITH TIME ZONE descriptor
OCI_DTYPE_TIMESTAMP	68	Specifies generation of TIMESTAMP descriptor
OCI_DTYPE_TIMESTAMP_LTZ	70	Specifies generation of TIMESTAMP WITH LOCAL TIME ZONE descriptor
OCI_DTYPE_TIMESTAMP_TZ	69	Specifies generation of TIMESTAMP WITH TIME ZONE descriptor
OCI_DTYPE_UCB	71	Specifies generation of user callback descriptor

OCI_EXEC_MODE theme

Constant	Value	Comment
OCI_BATCH_ERRORS	128	Collects information about any errors that occurred
OCI_BATCH_MODE	1	Batch OCI statement for execution
OCI_COMMIT_ON_SUCCESS	32	Selectively commits transactions at end of each statement
OCI_DESCRIBE_ONLY	16	Returns select-list description without executing statement
OCI_EXACT_FETCH	2	Used when application knows exact number of rows to be fetched in advance
OCI_KEEP_FETCH_STATE	4	Result set rows of statement are kept during transaction migration
OCI_NON_BLOCKING	64	Non-blocking statement execution mode
OCI_PARSE_ONLY	256	Allows user to parse query prior to execution
OCI_SCROLLABLE_CURSOR	8	Specifies cursor is scrollable
OCI_SHOW_DML_WARNINGS	1024	Returns diagnostic information

OCI_HTYPE theme

Constant	Value	Comment
OCI_HTYPE_BIND	5	OCI bind handle
OCI_HTYPE_COMPLEXOBJECT	11	OCI complex object retrieval (COR) handle
OCI_HTYPE_DEFINE	6	OCI define handle
OCI_HTYPE_DESCRIBE	7	OCI describe handle
OCI_HTYPE_DIRPATH_COLUMN_ARRAY	15	OCI direct path column array handle
OCI_HTYPE_DIRPATH_CTX	14	OCI direct path context handle
OCI_HTYPE_DIRPATH_STREAM	16	OCI direct path stream handle
OCI_HTYPE_ENV	1	OCI environment handle
OCI_HTYPE_ERROR	2	OCI error handle
OCI_HTYPE_FIRST	1	First OCI handle
OCI_HTYPE_LAST	17	Last OCI handle
OCI_HTYPE_PROC	17	OCI process handle
OCI_HTYPE_SECURITY	12	OCI security handle
OCI_HTYPE_SERVER	8	OCI server handle
OCI_HTYPE_SESSION	9	OCI user session handle
OCI_HTYPE_STMT	4	OCI statement handle
OCI_HTYPE_SUBSCRIPTION	13	OCI subscription handle
OCI_HTYPE_SVCCTX	3	OCI service context handle
OCI_HTYPE_TRANS	10	OCI transaction handle

OCI_MISC. theme

Constant	Value	Comment
BIND_IN	0	Direction for IN binds
BIND_IO	2	Direction for IN OUT binds
BIND_OUT	1	Direction for OUT binds
OCI_CRED_EXT	2	Authenticate using external credentials
OCI_CRED_RDBMS	1	Authenticate using database user name and password pair as credentials
OCI_STMT_ALTER	7	ALTER statement
OCI_STMT_BEGIN	8	BEGIN... (PL/SQL) statement
OCI_STMT_CREATE	5	CREATE statement
OCI_STMT_DECLARE	39	DECLARE... (PL/SQL) statement
OCI_STMT_DELETE	3	DELETE statement
OCI_STMT_DROP	6	DROP statement
OCI_STMT_INSERT	4	INSERT statement
OCI_STMT_SELECT	1	SELECT statement
OCI_STMT_UPDATE	2	UPDATE statement

OCI_MODE theme

Constant	Value	Comment
OCI_CACHE	512	Specifies use of cache
OCI_DATA_AT_EXEC	2	Defines total size of data that can be provided at runtime
OCI_DEFAULT	0	Default mode
OCI_DYNAMIC_FETCH	2	Fetches data dynamically at runtime
OCI_EVENTS	4	Utilizes publish-subscribe notifications
OCI_NO_MUTEX	128	No mutexing in this mode
OCI_NO_SHARING	1	Disables sharing mode for SQL statement
OCI_NO_UCB	64	Suppresses calling of dynamic callback routine
OCI_OBJECT	2	Uses object features
OCI_PIECEWISE	4	Operation on one of many pieces
OCI_SHARED	16	Uses shared data structures
OCI_SHARED_EXT	256	Used for shared forms
OCI_THREADED	1	Uses threaded environment

OCI_PTYPE theme

Constant	Value	Comment
OCI_PTYPE_ARG	10	Argument of function or procedure
OCI_PTYPE_COL	9	Column of table or view
OCI_PTYPE_DATABASE	18	Database
OCI_PTYPE_FUNC	4	Function
OCI_PTYPE_LIST	11	Column list, argument list, or subprogram list
OCI_PTYPE_PKG	5	Package
OCI_PTYPE_PROC	3	Procedure
OCI_PTYPE_SCHEMA	17	Schema
OCI_PTYPE_SEQ	8	Sequence
OCI_PTYPE_SYN	7	Synonym
OCI_PTYPE_TABLE	1	Table
OCI_PTYPE_TYPE	6	Type
OCI_PTYPE_TYPE_ARG	15	Argument of type method
OCI_PTYPE_TYPE_ATTR	12	Attribute of type
OCI_PTYPE_TYPE_COLL	13	Collection type information
OCI_PTYPE_TYPE_METHOD	14	Method of type
OCI_PTYPE_TYPE_RESULT	16	Results of method
OCI_PTYPE_UNK	0	Unknown schema object
OCI_PTYPE_VIEW	2	View

OCI_RETURN_VALUE theme

Constant	Value	Comment
OCI_CONTINUE	24200	Indicates callback function wants normal processing to resume
OCI_ERROR	-1	Returns additional information after function failure
OCI_INVALID_HANDLE	-2	Invalid handle was passed
OCI_NEED_DATA	99	Application needs to provide runtime data
OCI_NO_DATA	100	Indicates there is no further data
OCI_STILL_EXECUTING	-3123	Current operation could not be completed immediately
OCI_SUCCESS	0	Function completed successfully
OCI_SUCCESS_WITH_INFO	1	Indicates function completed successfully and returns additional diagnostic information

Examples of use

- Connecting to an Oracle database
- Reading from an Oracle BLOB column
- Writing in a LONG RAW column
- Reading from a LONG RAW column
- Working with Oracle Date types
- Executing an SQL SELECT request
- Executing an SQL INSERT request
- Executing an SQL UPDATE request
- Executing an SQL DELETE request
- Closing a connection

Connecting to an Oracle database

This 4D method allows a user to connect to an Oracle database. To keep the code simple, we have not included any error-handling.

Although this code may seem a bit long for a simple connection, you need to keep in mind that 4D for OCI also includes a simpler alternative by means of the **OCILogon** command, which lets you avoid many of the handle assignments and helps to simplify and shorten your code. The example shown below is primarily for didactic purposes.

This method starts by assigning the various handles, in hierarchical order. All handles are assigned with respect to the environment handle.

OCIServerAttach creates an access to the Oracle server by associating a connection string to the server handle. **OCISessionBegin** starts the session itself, by establishing the connection.

Example

Source code for ***OCI_CONNECT*** project method:

```
 //Method: CONNECT
 //Example of method call: CONNECT ("SCOTT";"TIGER";"ORAQA")
 //Use: connects a user to an Oracle database
 //$1: user name
 //$2: password
 //$3: connection string or name of Oracle service specified in 'tnsnames.ora' file

C_TEXT($1)
C_LONGINT(envhp) //environment handle
C_LONGINT(svchp) //context handle
C_LONGINT(authp) //session handle
C_LONGINT(srvhp) //server handle
C_LONGINT($status) //return of OCI commands

 //Allocation of handles
$status:=OCIEnvCreate(envhp;OCI_DEFAULT) //environment handle. 'Default' environment

 //(No processing of objects...)
$status:=OCIHandleAlloc(envhp;errhp;OCI_HTYPE_ERROR) //handle of errors (if any)
$status:=OCIHandleAlloc(envhp;svchp;OCI_HTYPE_SVCCTX) //context handle
$status:=OCIHandleAlloc(envhp;authp;OCI_HTYPE_SESSION) //session handle
$status:=OCIHandleAlloc(envhp;srvhp;OCI_HTYPE_SERVER) //server handle

 //create access to server by assigning connection string to the server handle
$status:=OCIServerAttach(srvhp;errhp;$3)

 //assign server handle to context handle
$status:=OCIAttrSetVal(svchp;srvhp;OCI_ATTR_SERVER;errhp)

 //update name and password attributes of session handle
 //with parameters supplied to this method
$status:=OCIAttrSetText(authp;$1;OCI_ATTR_USERNAME;errhp)
$status:=OCIAttrSetText(authp;$2;OCI_ATTR_PASSWORD;errhp)

 //assign session handle to context handle
$status:=OCIAttrSetVal(svchp;authp;OCI_ATTR_SESSION;errhp)

 //start of user session
$status:=OCISessionBegin(svchp;errhp;authp;OCI_CRED_RDBMS;OCI_DEFAULT)
```

Reading from an Oracle BLOB column

This sample code retrieves data from a BLOB type Oracle column using commands of the 4D for OCI plug-in.

Example

Method: ***OCI_GET_BLOB***

```
 //connection parameters to modify
$user:="xxx"
$password:="xxx"
$server:="xxx"
 //complex login
status:=OCIEnvCreate(envhp;OCI_DEFAULT)
status:=OCIHandleAlloc(envhp;errhp;OCI_HTYPE_ERROR)
status:=OCIHandleAlloc(envhp;svchp;OCI_HTYPE_SVCCTX)
status:=OCIHandleAlloc(envhp;authp;OCI_HTYPE_SESSION)
status:=OCIHandleAlloc(envhp;srvhp;OCI_HTYPE_SERVER)
status:=OCIServerAttach(srvhp;errhp;$server)
status:=OCIAttrSetVal(svchp;srvhp;OCI_ATTR_SERVER;errhp)
status:=OCIAttrSetText(authp;$user;OCI_ATTR_USERNAME;errhp)
status:=OCIAttrSetText(authp;$password;OCI_ATTR_PASSWORD;errhp)
status:=OCISessionBegin(svchp;errhp;authp;OCI_CRED_RDBMS;OCI_DEFAULT)
status:=OCIAttrSetVal(svchp;authp;OCI_ATTR_SESSION;errhp)

 //SQL request
$sql_statement:="SELECT t_blob FROM test_lob WHERE t_id = 1"

C_LONGINT(bloblocator)
C_LONGINT($define;$position;$sqlt)
C_POINTER($p_define)
C_LONGINT(null_ind01;rlen01;rcode01)

bloblocator:=0
$define:=0
$position:=1
$p_define:=->bloblocator
$sqlt:=SQLT_BLOB
C_BLOB($blob)

 //preparation of request
status:=OCIHandleAlloc(envhp;stmthp;OCI_HTYPE_STMT)
status:=OCIDescriptorAlloc(envhp;bloblocator;OCI_DTYPE_LOB)
status:=OCIStmtPrepare(stmthp;errhp;$sql_statement;OCI_DEFAULT)
status:=OCIDefineByPos(stmthp;$define;errhp;$position;$p_define;$sqlt;->null_ind01;->rlen01;-
>rcode01;OCI_DEFAULT)

status:=OCIStmtExecute(svchp;stmthp;errhp;1;0;0;0;OCI_DEFAULT)

status:=OCILobRead(svchp;errhp;bloblocator;1;$blob)

 //release
status:=OCIDescriptorFree(bloblocator)
status:=OCIHandleFree(stmthp)
status:=OCISessionEnd(svchp;errhp;authp)
status:=OCIServerDetach(srvhp;errhp)
status:=OCIHandleFree(envhp)

 //retrieval from Blob into a file
```

```
If(BLOB size($blob)>0)

 //if the Oracle BLOB column was supplied by 4D for Oracle,
 //we have to remove the first 8 bytes from the BLOB received
 //DELETE FROM BLOB($blob;0;8)

 //we have hard-coded the file name but we can create the name using programming
 //and retrieve the file type in the Blob for the extension
   $DocRef:=Create document("image_1.jpg")
   If(OK=1)
      CLOSE DOCUMENT($DocRef)
      BLOB TO DOCUMENT(Document;$blob)
   End if

Else
   ALERT("No contents in Blob!")
End if
 //we empty the BLOB
SET BLOB SIZE($blob;0)
```

Description of Oracle table used in this example:

```
CREATE TABLE TEST_LOB (
T_ID NUMBER(5,0),
T_NAME VARCHAR2(80),
T_BLOB BLOB
);
```

Writing in a LONG RAW column

You can use 4D for OCI to write the contents of a LONG RAW column in an Oracle database.

Example

```
C_LONGINT(envhp;errhp;svchp)
C_LONGINT(Define;stmthp;stmtSelecthp)
C_LONGINT(RealLength)
C_BLOB(TheBlob)
SET BLOB SIZE(TheBlob;0)
C_LONGINT(vp_Null1)
C_POINTER(vp_Null3)

C_TIME($DocRef)
C_TEXT($Path)
$DocRef:=Open document("";"";Get Pathname)
If(OK=1)
   $Path:=document
   CLOSE DOCUMENT($DocRef)
End if
DOCUMENT TO BLOB($Path;TheBlob)
WriteLength:=BLOB size(TheBlob)

C_TEXT($UserName;$Password;$DbService;$SQLStatement)
$UserName:="xxx"
$Password:="xxx"
$DbService:="xxx"

$Status:=OCIEnvCreate(envhp;OCI_OBJECT)
$Status:=OCIHandleAlloc(envhp;errhp;OCI_HTYPE_ERROR)
$Status:=OCILogon(envhp;errhp;svchp;$UserName;$Password;$DbService)
$Status:=OCIHandleAlloc(envhp;stmthp;OCI_HTYPE_STMT)

$SQLStatement:="INSERT INTO test_LGRAW (key, lgraw) "
$SQLStatement:=$SQLStatement+"VALUES (3, EMPTY_BLOB())"

$Status:=OCIStmtPrepare(stmthp;errhp;$SQLStatement;OCI_DEFAULT)
$Status:=OCIStmtExecute(svchp;stmthp;errhp;1;0;0;0;OCI_DEFAULT)

$SQLStatement:="UPDATE test_LGRAW SET lgraw=:1 WHERE key=3"

$Status:=OCIHandleAlloc(envhp;stmtSelecthp;OCI_HTYPE_STMT)
$Status:=OCIStmtPrepare(stmtSelecthp;errhp;$SQLStatement;OCI_DEFAULT)

vp_Null1:=1
WriteLength:=BLOB size(TheBlob)
$Status:=OCIBindByPos(stmtSelecthp;Define;errhp;1;->TheBlob;SQLT_LBI;->vp_Null1;
->WriteLength;->vp_Null3;OCI_DATA_AT_EXEC)
$status:=OCIErrorGet(errhp;1;$errornum;$errorexplain)
If($errornum#0)
   ALERT(String($errornum)+Char(13)+$errorexplain)
End if

$Status:=OCIStmtExecute(svchp;stmtSelecthp;errhp;1;0;0;0;OCI_DEFAULT)
$Status:=OCIHandleFree(stmtSelecthp)
$status:=OCIErrorGet(errhp;1;$errornum;$errorexplain)
If($errornum#0)
   ALERT(String($errornum)+Char(13)+$errorexplain)
```

```
End if

$Status:=OCIHandleFree(stmthp)
$Status:=OCILogoff(svchp;errhp)
$Status:=OCIHandleFree(errhp)
$Status:=OCIHandleFree(envhp)
```

Reading from a LONG RAW column

You can use 4D for OCI to read the contents of a LONG RAW column in an Oracle database under the following conditions:

1. the receiving variable must be of the BLOB type
2. the datatype must be SQLT_LBI
3. the mode must be OCI_DYNAMIC_FETCH (and not OCI_DEFAULT)

Example

```
 //Method: Read_long_raw
 //reads data from a long raw column
 //and creates the picture file (jpg) in the database folder

C_LONGINT(envhp;errhp;svchp;vDefine;stmthp;stmtSelecthp)
C_LONGINT(RealLength)
C_BLOB(TheBlob)

 //-------------------------------------------
 //CONDITION 1: Blob
SET BLOB SIZE(TheBlob;0)
 //-------------------------------------------

C_POINTER(vp_Null1;vp_Null3)
ReadLength:=0

C_TEXT($UserName;$Password;$ServiceName;$SQL_statement)

$UserName:="xxx"
$Password:="xxx"
$ServiceName:="xxx"

Status:=OCIEnvCreate(envhp;OCI_OBJECT)
Status:=OCIHandleAlloc(envhp;errhp;OCI_HTYPE_ERROR)
Status:=OCILogon(envhp;errhp;svchp;$UserName;$Password;$ServiceName)
Status:=OCIHandleAlloc(envhp;stmthp;OCI_HTYPE_STMT)

$SQL_statement:="SELECT lgraw FROM test_LGRAW WHERE key = 20"
Status:=OCIHandleAlloc(envhp;stmtSelecthp;OCI_HTYPE_STMT)
If(Status=OCI_SUCCESS)
   Status:=OCIStmtPrepare(stmtSelecthp;errhp;$SQL_statement;OCI_DEFAULT)

 //-------------------------------------------
 //CONDITION 2: SQLT_LBI for LONG RAW
 //CONDITION 3: OCI_DYNAMIC_FETCH instead of OCI_DEFAULT

   Status:=OCIDefineByPos(stmtSelecthp;vDefine;errhp;1;->TheBlob;SQLT_LBI;vp_Null1;
   ->ReadLength;vp_Null3;OCI_DYNAMIC_FETCH)
 //-------------------------------------------

   $status:=OCIErrorGet(errhp;1;$errornum;$errorexplain)
   If($errornum#0)
      ALERT(String($errornum)+Char(13)+$errorexplain)
   End if
   Repeat
      Status:=OCIStmtExecute(svchp;stmtSelecthp;errhp;1;0;0;0;OCI_DEFAULT)
```

```
   Until(Status#OCI_STILL_EXECUTING)
   Status:=OCIHandleFree(stmtSelecthp)
End if

 //remove the first 8 characters if the picture was inserted with 4D for Oracle
DELETE FROM BLOB(TheBlob;0;8)

If(ReadLength>0)
   $DocRef:=Create document("longraw_image.jpg")
   If(OK=1)
      CLOSE DOCUMENT($DocRef)
      BLOB TO DOCUMENT(Document;TheBlob)
   End if
Else
   ALERT("Blob size: zero!")
End if

Status:=OCIHandleFree(stmthp)
Status:=OCILogoff(svchp;errhp)
Status:=OCIHandleFree(errhp)
Status:=OCIHandleFree(envhp)

SET BLOB SIZE(TheBlob;0) //empty Blob

 //End of method
```

Working with Oracle Date types

Unlike the Oracle Object Date Time (ODT) data type, in 4D dates and times are stored separately. If you encounter difficulties returning values from DATETIME or TIMESTAMP type Oracle columns with 4D for OCI, you should try using the one of the following specific functions:

- **OCIDefineDateByPos**
- **OCIBindDateByName**
- **OCIBindDateByPos**

Since Oracle date fields store the time as well, you must use the SQLT_ODT type and pass two pointers instead of just one: one for the date and a second for the time, even if you do not need to use it.

The functions listed above work the same way as the Oracle commands they are based on (i.e.: **OCIDefineByPos**, **OCIBindByName** and **OCIBindByPos**), except that we have added a second pointer parameter to adapt them for working with Oracle Date fields.

Example

Using **OCIDefineDateByPos** with a SELECT request:

```
ARRAY DATE(date_results;3) //date array for first pointer
ARRAY LONGINT(time_results;3) //time array for second pointer
ARRAY LONGINT($arrNull;1)
ARRAY LONGINT($arrSizes;1)
ARRAY LONGINT($arrCodes;1)

$arrSizes{1}:=255

$UserName:="xxx"
$Password:="xxx"
$OraServ:="xxx"

$sql:="SELECT inv_date FROM invoices"

$Status:=OCIEnvCreate($DescrEnv;OCI_DEFAULT)
$Status:=OCIHandleAlloc($DescrEnv;$DescrErr;OCI_HTYPE_ERROR)
$Status:=OCIHandleAlloc($DescrEnv;$DescrService;OCI_HTYPE_SVCCTX)
$Status:=OCIHandleAlloc($DescrEnv;$DescrAuth;OCI_HTYPE_SESSION)
$Status:=OCIHandleAlloc($DescrEnv;$DescrServer;OCI_HTYPE_SERVER)

$Status:=OCIServerAttach($DescrServer;$DescrErr;$OraServ)
$Status:=OCIAttrSetVal($DescrService;$DescrServer;OCI_ATTR_SERVER;$DescrErr)
$Status:=OCIAttrSetText($DescrAuth;$UserName;OCI_ATTR_USERNAME;$DescrErr)
$Status:=OCIAttrSetText($DescrAuth;$Password;OCI_ATTR_PASSWORD;$DescrErr)
$Status:=OCISessionBegin($DescrService;$DescrErr;$DescrAuth;OCI_CRED_RDBMS;OCI_DEFAULT)
$Status:=OCIAttrSetVal($DescrService;$DescrAuth;OCI_ATTR_SESSION;$DescrErr)

$Status:=OCIHandleAlloc($DescrEnv;$DescrStmt;OCI_HTYPE_STMT)
$Status:=OCIStmtPrepare($DescrStmt;$DescrErr;$sql;Length($sql))

$sqlt:=SQLT_ODT
$Status:=OCIDefineDateByPos($DescrStmt;$DescrDefine;$DescrErr;1;->date_results;
->time_results;$sqlt;->$arrNull;->$arrSizes;->$arrCodes;OCI_DEFAULT)

$Status:=OCIStmtExecute($DescrService;$DescrStmt;$DescrErr;0;0;0;0;OCI_DEFAULT)

$Status:=OCIStmtFetch($DescrStmt;$DescrErr;10)
```

```
$text:=String(date_results{1})+Char(13)
For($i;2;Size of array(date_results))
   $text:=$text+String(date_results{$i})+Char(13)
End for

ALERT("ARRAY:"+Char(13)+$text)

$Status:=OCISessionEnd($DescrService;$DescrErr;$DescrAuth)
$Status:=OCIServerDetach($DescrServer;$DescrErr)
$Status:=OCIHandleFree($DescrService)
$Status:=OCIHandleFree($DescrServer)
$Status:=OCIHandleFree($DescrErr)
$Status:=OCIHandleFree($DescrEnv)

ARRAY DATE(date_results;0)
ARRAY DATE(time_results;0)
```

In addition, here is a short SQL script to create the invoices table:

```
drop table invoices;
create table invoices (inv_ID varchar(3), date_fact date);
insert into invoices (inv_ID,inv_date) values ('F01','10/05/2014');
insert into invoices (inv_ID,inv_date) values ('F02','11/05/2014');
insert into invoices (inv_ID,inv_date) values ('F03','12/05/2014');
select * from invoices;
commit;
```

Executing an SQL SELECT request

The method in the example below executes an SQL **SELECT** request to retrieve values from columns in the Oracle "emp" table, which is a table of employees.

This method starts by allocating a handle for the SQL request (request handle). This type of allocation is typical for methods that work with SQL requests (**INSERT**, **UPDATE**, **DELETE**).

The method selects all the employees but only certain columns (fields). Here we select the employee number (empno), their name (ename), job title (job) and hire date (hiredate).
Once the label of the SQL request and the request handle have been defined and initialized, they are linked using the **OCIStmtPrepare** command.

For each column implicated in the SQL request, we associate an array in 4D to retrieve the values. To link the 4D array with the targeted Oracle column, we use the **OCIDefineByPos** command, which expects the number of the column mentioned in the SQL request.

Since Oracle's Date type is particular in that it consists of both a date and a time part, we have used the same command here as well (**OCIDefineByPos** command). We did this because the time component is of no interest to us here; otherwise we would have needed to use the **OCIDefineDateByPos** command.

Example

Source code of the ***OCI_SELECT*** project method:

```
C_TEXT($sql_request) //label of SQL request

C_LONGINT($status) //return code of OCI commands
C_LONGINT($errhp) //error handle
C_LONGINT($stmthp) //request handle
C_LONGINT($define) //define handle

 //information to retrieve from the Oracle table named 'emp'
C_LONGINT($column_number) //number of column to retrieve after execution of the SQL request
ARRAY LONGINT(tls_empno;20) //employee numbers ('empno' column)
ARRAY TEXT(tas_ename;20) //employee names ('ename' column of 'emp' table)
ARRAY TEXT(tas_job;20) //employee jobs ('job' column of 'emp' table)
ARRAY DATE(tds_hiredate;20) //hire dates ('hiredate' column of 'emp' table)
C_LONGINT($max_emp) //maximum number of employees to retrieve
C_LONGINT(null_ind1;null_ind2;null_ind3) //indicator variables

 //selection request of list of numbers, names, jobs and hire dates of all employees
 //in the Oracle 'emp' table
$requete_sql:="SELECT empno, ename, job, hiredate FROM emp"

 //allocation of request handle. The envhp environment handle
 //has been allocated in the OCI_CONNECT connection method
$status:=OCIHandleAlloc(envhp;$stmthp;OCI_HTYPE_STMT)

 //allocation of error handle
$status:=OCIHandleAlloc(envhp;$errhp;OCI_HTYPE_ERROR)

 //assigning of SQL request label to request handle
$status:=OCIStmtPrepare($stmthp;$errhp;$requete_sql;OCI_DEFAULT)

 //indicator variables for OCIDefineByPos() command
 //The information returned by these variables is useless in our example
 //These variables can be used to find out whether there are any NULL or truncated values.
null_ind1:=0
```

```
null_ind2:=0
null_ind3:=0

 //implementation of the Define for each column specified in the SQL request
 //Note that $define, which specifies the define handle, did not need to be explicitly
allocated.

$column_number:=1 //number of column pointed to, recovered by executing the (empno) request
$status:=OCIDefineByPos($stmthp;$define;$errhp;$column_number;->tls_empno;SQLT_INT;
->null_ind1;->null_ind2;->null_ind3;OCI_DEFAULT)

$column_number:=2 //number of column pointed to, recovered by executing the (ename) request
$status:=OCIDefineByPos($stmthp;$define;$errhp;$column_number;->tas_ename;SQLT_STR;
->null_ind1;->null_ind2;->null_ind3;OCI_DEFAULT)

$column_number:=3 //number of column pointed to, recovered by executing the (job) request
$status:=OCIDefineByPos($stmthp;$define;$errhp;$column_number;->tas_job;SQLT_STR;
->null_ind1;->null_ind2;->null_ind3;OCI_DEFAULT)

$column_number:=4 //number of column pointed to, recovered by executing the (hiredate) request
$status:=OCIDefineByPos($stmthp;$define;$errhp;$column_number;->tds_hiredate;SQLT_ODT;
->null_ind1;->null_ind2;->null_ind3;OCI_DEFAULT)

 //retrieval of up to 20 employee records
$max_emp:=20

 //execution of SQL request
$status:=OCIStmtExecute(svchp;$stmthp;$errhp;$max_emp;0;0;0;OCI_DEFAULT)

 //release of request handle$status:=OCIHandleFree ($stmthp)

 //release of error handle
$status:=OCIHandleFree($errhp)
```

Executing an SQL INSERT request

The method in the example below executes an SQL INSERT request to add records to the Oracle "emp" table.

The values to be added are stored in 4D arrays. We associate the data to be added with the Oracle columns by using the ":" character in the SQL request. The association of 4D data with Oracle columns can be done either by position (**OCIBindByPos**) or by name (**OCIBindByName**).

Note that for inserting values of the Date type, we use the **OCIBindDateByPos** command, which expects both date and time variables to be passed. This way it is able to represent the entirety of the Oracle Object Date Time (ODT) type, which contains 2 parts: date and time. Since the time part does not concern us, we used a 4D time variable with a null value.

Example

Source code for ***OCI_INSERT*** project method:

```
C_TEXT($sql_request) //label of SQL request

C_LONGINT($status) //return code of OCI commands
C_LONGINT($errhp) //error handle
C_LONGINT($stmthp) //request handle
C_LONGINT($bind) //bind handle

C_POINTER(pnull_ind1;pnull_ind2;pnull_ind3) //indicator variables (see the OCI_SELECT method)

C_LONGINT($nb_emp) //number of employees to insert

 //declaration of data to insert
ARRAY LONGINT(tli_empno;3) //empno column
ARRAY TEXT(tai_ename;3) //ename column
ARRAY TEXT(tai_job;3) //job column
ARRAY LONGINT(tli_mgr;3) //mgr column
ARRAY DATE(tdi_hiredate;3) //hiredate column
C_TIME(null_time) //time type variable, in addition to the date column, to represent
 //the whole Oracle ODT ODT type, which includes the date and time
ARRAY LONGINT(tli_sal;3) //sal column
ARRAY LONGINT(tli_comm;3) //comm column
ARRAY LONGINT(tli_deptno;3) //deptno column

 //filling in data to be inserted
 //empno column
tli_empno{1}:=1111
tli_empno{2}:=2222
tli_empno{3}:=3333
 //ename column
tai_ename{1}:="Joel"
tai_ename{2}:="Catherine"
tai_ename{3}:="Marianne"
 //job column
tai_job{1}:="analyst"
tai_job{2}:="salesperson"
tai_job{3}:="manager"
 //mgr column
tli_mgr{1}:=7902
tli_mgr{2}:=7698
tli_mgr{3}:=7788
 //hiredate column
tdi_hiredate{1}:=!01/01/03!
tdi_hiredate{2}:=!02/01/03!
```

```
 tdi_hiredate{3}:=!03/01/03!
 null_time:=?00:00:00?
  //sal column
 tli_sal{1}:=1915
 tli_sal{2}:=2012
 tli_sal{3}:=1713
  //comm column
 tli_comm{1}:=100
 tli_comm{2}:=200
 tli_comm{3}:=150
  //deptno column
 tli_deptno{1}:=20
 tli_deptno{2}:=30
 tli_deptno{3}:=20

  //SQL request for data insertion. We specify the columns of the 'emp' table which will be
filled
  //Note the use of the ":" character to indicate that a variable will provide the data to
insert
  //This indication will either be by name (name of the 4D variable) or by position (number of
  //position in the request).
 $sql_request:="INSERT INTO emp (empno, ename, job, mgr, hiredate, sal, comm, deptno)"
 $sql_request:=$sql_request+" VALUES (:tli_empno, :tai_ename, :tai_job, :tli_mgr,"
 $sql_request:=$sql_request+" :tdi_hiredate, :tli_sal,:tli_comm,:tli_deptno)"

  //allocation of request handle
 $status:=OCIHandleAlloc(envhp;$stmthp;OCI_HTYPE_STMT)

  //allocation of error handle
 $status:=OCIHandleAlloc(envhp;$errhp;OCI_HTYPE_ERROR)

  //assigning the label of the SQL request to the request handle
 $status:=OCIStmtPrepare($stmthp;$errhp;$sql_request;OCI_DEFAULT)

  //performing the bind for each column of the request. Each bind is carried out by position
  //note that the $bind bind handle does not have to be explicitly allocated
 $status:=OCIBindByPos($stmthp;$bind;$errhp;1;->
 tli_empno;SQLT_INT;pnull_ind1;pnull_ind2;pnull_ind3;OCI_DEFAULT;BIND_IN)
 $status:=OCIBindByPos($stmthp;$bind;$errhp;2;->
 tai_ename;SQLT_STR;pnull_ind1;pnull_ind2;pnull_ind3;OCI_DEFAULT;BIND_IN)
 $status:=OCIBindByPos($stmthp;$bind;$errhp;3;->
 tai_job;SQLT_STR;pnull_ind1;pnull_ind2;pnull_ind3;OCI_DEFAULT;BIND_IN)
 $status:=OCIBindByPos($stmthp;$bind;$errhp;4;->
 tli_mgr;SQLT_INT;pnull_ind1;pnull_ind2;pnull_ind3;OCI_DEFAULT;BIND_IN)

  //the Oracle ODT type recovers both the date and time
  //since the time value does not interest us here, it is passed as null
 $status:=OCIBindDateByPos($stmthp;$bind;$errhp;5;->null_time;->
 tdi_hiredate;SQLT_ODT;pnull_ind1;pnull_ind2;pnull_ind3;OCI_DEFAULT;BIND_IN)
 $status:=OCIBindByPos($stmthp;$bind;$errhp;6;->
 tli_sal;SQLT_INT;pnull_ind1;pnull_ind2;pnull_ind3;OCI_DEFAULT;BIND_IN)
 $status:=OCIBindByPos($stmthp;$bind;$errhp;7;->
 tli_comm;SQLT_INT;pnull_ind1;pnull_ind2;pnull_ind3;OCI_DEFAULT;BIND_IN)
 $status:=OCIBindByPos($stmthp;$bind;$errhp;8;->
 tli_deptno;SQLT_INT;pnull_ind1;pnull_ind2;pnull_ind3;OCI_DEFAULT;BIND_IN)

  //number of employees to insert
 $nb_emp:=3

  //execution of SQL request
 $status:=OCIStmtExecute(svchp;$stmthp;$errhp;$nb_emp;0;0;0;OCI_DEFAULT)

  //confirmation of insertion by immediate validation of the transaction
  //otherwise, the insertion is done when the session is closed
 status:=OCITransCommit(svchp;$errhp;0)

  //release of request handle
 $status:=OCIHandleFree($stmthp)
```

```
 //release of error handle
$status:=OCIHandleFree($errhp)
```

Executing an SQL UPDATE request

The method in the example below executes an SQL UPDATE request to update records that were added previously to the Oracle "emp" table. In this method, we update 3 records that were added by the INSERT request. We are only modifying the names (ename column).

Note that here we have done a bind (association of a 4D variable with an Oracle column) by name. In the previous INSERT request, we did the bind by position.

Example

Source code for ***OCI_UPDATE*** project method:

```
C_TEXT($sql_request) //label of SQL request

C_LONGINT($status) //return code of OCI commands
C_LONGINT($errhp) //error handle
C_LONGINT($stmthp) //request handle
C_LONGINT($bind) //bind handle

C_POINTER(pnull_ind1;pnull_ind2;pnull_ind3) //indicator variables (see the OCI_SELECT method)

C_LONGINT($nb_emp) //number of employees to update

 //declaration of data to modify
ARRAY LONGINT(tlu_empno;3) //empno column
ARRAY TEXT(tau_ename;3) //ename column

 //search criteria in SQL request for the update
tlu_empno{1}:=1111
tlu_empno{2}:=2222
tlu_empno{3}:=3333
 //modified data
tau_ename{1}:="JJ"
tau_ename{2}:="CC"
tau_ename{3}:="MM"

 //update of employee names in the Oracle "emp" table whose numbers are 1111, 2222, and 3333
 //reminder: it is the data that was inserted that is modified
$sql_request:="UPDATE emp SET ename=:the_names WHERE empno=:the_numbers"

 //allocation of request handle
$status:=OCIHandleAlloc(envhp;$stmthp;OCI_HTYPE_STMT)

 //allocation of error handle
$status:=OCIHandleAlloc(envhp;$errhp;OCI_HTYPE_ERROR)

 //assignment SQL request label to request handle
$status:=OCIStmtPrepare($stmthp;$errhp;$sql_request;OCI_DEFAULT)

 //we do a bind by name. On the INSERT, we did it by position
$status:=OCIBindByName($stmthp;$bind;$errhp;":the_names";->
tau_ename;SQLT_STR;pnull_ind1;pnull_ind2;pnull_ind3;OCI_DEFAULT;BIND_IN)
$status:=OCIBindByName($stmthp;$bind;$errhp;":the_numbers";->
tlu_empno;SQLT_INT;pnull_ind1;pnull_ind2;pnull_ind3;OCI_DEFAULT;BIND_IN)

 //execution of SQL request
$nb_emp:=3
$status:=OCIStmtExecute(svchp;$stmthp;$errhp;$nb_emp;0;0;0;OCI_DEFAULT)
```

```
 //confirmation of modification by immediate validation of the transaction
 //otherwise, the update is done when the session is closed
status:=OCITransCommit(svchp;$errhp;0)

 //release of request handle
$status:=OCIHandleFree($stmthp)

 //release of error handle
$status:=OCIHandleFree($errhp)
```

Executing an SQL DELETE request

The method in the example below executes an SQL DELETE request in order to remove records added previously in the Oracle "emp" table.

We delete the records that were inserted and updated previously. The code of this method is similar to that of the UPDATE request in that we have also done a bind (association of a 4D variable with an Oracle column) by name.

Example

Source code of the ***OCI_DELETE*** project method:

```
C_TEXT($sql_request) //label of SQL request

C_LONGINT($status) //return code of OCI commands
C_LONGINT($errhp) //error handle
C_LONGINT($stmthp) //request handle
C_LONGINT($bind) //bind handle

C_POINTER(pnull_ind1;pnull_ind2;pnull_ind3) //indicator variables (see the OCI_SELECT method)

C_LONGINT($nb_emp) //number of employees to delete from the Oracle database
ARRAY LONGINT(tld_empno;3) //declaration of data to search for deletion

 //numbers of employee records to delete from Oracle database
tld_empno{1}:=1111
tld_empno{2}:=2222
tld_empno{3}:=3333

 //deletion from Oracle emp table of employees whose numbers are 1111, 2222, and 3333
 //these employees were added using the OCI_INSERT method
$sql_request:="DELETE FROM emp WHERE empno=:numemp"

 //allocation of request handle
$status:=OCIHandleAlloc(envhp;$stmthp;OCI_HTYPE_STMT)

 //allocation of error handle
$status:=OCIHandleAlloc(envhp;$errhp;OCI_HTYPE_ERROR)

 //assigning of SQL request label to request handle
$status:=OCIStmtPrepare($stmthp;$errhp;$sql_request;OCI_DEFAULT)

 //we do a bind by name
 //note that the $bind bind handle did not need to be allocated explicitly
$status:=OCIBindByName($stmthp;$bind;$errhp;":numemp";->
tld_empno;SQLT_INT;pnull_ind1;pnull_ind2;pnull_ind3;OCI_DEFAULT;BIND_IN)

 //execution of SQL request
$nb_emp:=3
$status:=OCIStmtExecute(svchp;$stmthp;$errhp;$nb_emp;0;0;0;OCI_DEFAULT)

 //confirmation of deletion by immediate validation of the transaction
 //otherwise, the deletion is done when the session is closed
$status:=OCITransCommit(svchp;$errhp;0)

 //release of request handle
$status:=OCIHandleFree($stmthp)

 //release of error handle
$status:=OCIHandleFree($errhp)
```

Closing a connection

Here we use the **OCIHandleFree** command to free the handles, and then close the session. When handles are structured hierarchically, closing the parent handle also closes its related handles.

The code below is used to disconnect from the Oracle server. We could also use the **OCILogoff** command instead.

OCISessionEnd deletes the session.
OCIServerDetach deletes the path to the Oracle server.

After this, we can free the handles one by one, or just free the environment handles, which in turn will release the other handles.

Example

Source code for ***OCI_DISCONNECT*** project method:

```
 //Method: DISCONNECT
 //Method call: DISCONNECT
 //Use: disconnects the currently-connected user

C_LONGINT($status) //return of OCI commands

$status:=OCISessionEnd(svchp;errhp;authp) //deletion of user session
$status:=OCIServerDetach(srvhp;errhp) //deletion of access to data source
$status:=OCIHandleFree(envhp) //frees the environment handle
 //which in turn frees up all the other handles
```

Connection

- OCIBreak
- OCILogoff
- OCILogon
- OCIParamGet
- OCIParamSet
- OCIPasswordChange
- OCIReset
- OCIServerAttach
- OCIServerDetach
- OCISessionBegin
- OCISessionEnd

OCILogoff

OCILogoff (svchp ; errhp) -> Function result

Parameter	Type		Description
svchp	Longint	⇒	Service context handle
errhp	Longint	⇒	Error handle
		⇐	
Function result	Longint	↩	Status

Description

Please refer to the **OCILogoff** command in the OCI documentation provided by Oracle.

OCILogon

OCILogon (envhp ; errhp ; svchp ; username ; password ; dbname) -> Function result

Parameter	Type		Description
envhp	Longint	⇒	OCI environment handle
errhp	Longint	⇒	Error handle
		⇐	
svchp	Longint	⇒	Service context handle
		⇐	
username	String	⇒	Name of user
password	String	⇒	User's password
dbname	String	⇒	Name of database to connect to
Function result	Longint	⇐	Status

Description

Please refer to the **OCILogon** command in the OCI documentation provided by Oracle.

Example

See the example for **Connecting to an Oracle database**.

OCIParamGet

OCIParamGet (hndlp ; errhp ; parmdpp ; pos) -> Function result

Parameter	Type		Description
hndlp	Longint	⇒	Handle pointer
errhp	Longint	⇒	Error handle
		⇐	
parmdpp	Longint	⇐	Descriptor of parameter found at position passed in pos
pos	Longint	⇒	Position number
Function result	Longint	↩	Status

Description

Please refer to the **OCIParamGet** command in the OCI documentation provided by Oracle.

OCIParamSet

OCIParamSet (hndlp ; errhp ; dscp ; pos) -> Function result

Parameter	Type		Description
hndlp	Longint	⇒ ⇐	Handle pointer
errhp	Longint	⇒ ⇐	Error handle
dscp	Longint	⇒	Descriptor pointer
pos	Longint	⇒	Position number
Function result	Longint	↩	Status

Description

Please refer to the **OCIParamSet** command in the OCI documentation provided by Oracle.

OCIPasswordChange

OCIPasswordChange (svchp ; errhp ; user_name ; opasswd ; npasswd ; mode) -> Function result

Parameter	Type		Description
svchp	Longint	⇒	Service context handle
		⇐	
errhp	Longint	⇒	Error handle
user_name	String	⇒	Name of user (can be in UTF-16 encoding)
opasswd	String	⇒	User's old password
npasswd	String	⇒	User's new password (can be in UTF-16 encoding)
mode	Longint	⇒	Mode of operation
Function result	Longint	⇐	Status

Description

Please refer to the **OCIPasswordChange** command in the OCI documentation provided by Oracle.

OCIReset

OCIReset (hndlp ; errhp) -> Function result

Parameter	Type		Description
hndlp	Longint	⇒	Server or service context handle
errhp	Longint	⇒	Error handle
Function result	Longint	⇐	Status

Description

Please refer to the **OCIReset** command in the OCI documentation provided by Oracle.

OCIServerAttach

OCIServerAttach (svchp ; errhp ; mode) -> Function result

Parameter	Type		Description
svchp	Longint	⇨ ⇦	Server handle
errhp	Longint	⇨ ⇦	Error handle
mode	Longint	⇨	Mode of operation
Function result	Longint	⇦	Status

Description

Please refer to the **OCIServerAttach** command in the OCI documentation provided by Oracle.

Example

See the examples for **Connecting to an Oracle database** and **Reading from an Oracle BLOB column**.

OCIServerDetach

OCIServerDetach (svchp ; errhp) -> Function result

Parameter	Type		Description
svchp	Longint	⇒	Server context handle
errhp	Longint	⇒	Error handle
		⇐	
Function result	Longint	↩	Status

Description

Please refer to the **OCIServerDetach** command in the OCI documentation provided by Oracle

Example

See the example for **Closing a connection**.

OCISessionBegin

OCISessionBegin (svchp ; errhp ; usrhp ; credt ; mode) -> Function result

Parameter	Type		Description
svchp	Longint	⇒	Service context handle
errhp	Longint	⇒	Error handle
usrhp	Longint	⇒	User session context handle
		⇐	
credt	Longint	⇒	Credentials to use for establishing user session
mode	Longint	⇒	Mode of operation
Function result	Longint	⇒	Status

Description

Please refer to the **OCISessionBegin** command in the OCI documentation provided by Oracle.

Example

See the example for **Connecting to an Oracle database**.

OCISessionEnd

OCISessionEnd (svchp ; errhp ; mode) -> Function result

Parameter	Type		Description
svchp	Longint	⇒ ⇐	Service context handle
errhp	Longint	⇒ ⇐	Error handle
mode	Longint	⇒	Only default mode is valid
Function result	Longint	↩	Status

Description

Please refer to the **OCISessionEnd** command in the OCI documentation provided by Oracle.

Example

See the example for **Closing a connection**.

Datatype

- OCIAttrGetText
- OCIAttrGetVal
- OCIAttrSetText
- OCIAttrSetVal
- OCIBindByName
- OCIBindByPos
- OCIDefineByPos
- OCIDescribeAnyText
- OCIDescribeAnyVal
- OCIDescriptorAlloc
- OCIDescriptorFree
- OCIEnvCreate
- OCIErrorGet
- OCIHandleAlloc
- OCIHandleFree
- OCIServerVersion
- OCIStmtExecute
- OCIStmtFetch
- OCIStmtGetBindInfo
- OCIStmtPrepare
- OCITerminate

OCIAttrGetText

OCIAttrGetText (trgthndlp ; attributep ; attrtype ; errhp) -> Function result

Parameter	Type		Description
trgthndlp	Longint	⇒	Handle type whose attribute are retrieved
attributep	String	⇐	Storage for attribute value
attrtype	Longint	⇒	Type of attribute being retrieved
errhp	Longint	⇒	Error handle
		⇐	
Function result	Longint	↩	Status

Description

Please refer to the **OCIAttrGet** command in the OCI documentation provided by Oracle.

OCIAttrGetVal

OCIAttrGetVal (trgthndlp ; attributep ; attrtype ; errhp) -> Function result

Parameter	Type		Description
trgthndlp	Longint	⇒	Handle type whose attributes are retrieved
attributep	Longint	⇐	Storage for attribute value
attrtype	Longint	⇒	Type of attribute being retrieved
errhp	Longint	⇒	Error handle
		⇐	
Function result	Longint	↩	Status

Description

Please refer to the **OCIAttrGet** command in the OCI documentation provided by Oracle.

OCIAttrSetText

OCIAttrSetText (trgthndlp ; attributep ; attrtype ; errhp) -> Function result

Parameter	Type		Description
trgthndlp	Longint	⇒	Handle type whose attribute gets modified
		⇐	
attributep	String	⇒	Attribute value
attrtype	Longint	⇒	Type of attribute being set
errhp	Longint	⇒	Error handle
		⇐	
Function result	Longint	⇐	Status

Description

Please refer to the **OCIAttrSet** command in the OCI documentation provided by Oracle.

OCIAttrSetVal

OCIAttrSetVal (trgthndlp ; attributep ; attrtype ; errhp) -> Function result

Parameter	Type		Description
trgthndlp	Longint	⇒	Handle type whose attribute gets modified
		⇐	
attributep	Longint	⇒	Attribute value
attrtype	Longint	⇒	Type of attribute being set
errhp	Longint	⇒	Error handle
		⇐	
Function result	Longint	↩	Status

Description

Please refer to the **OCIAttrSet** command in the OCI documentation provided by Oracle.

OCIBindByName

OCIBindByName (stmtp ; bindpp ; errhp ; placeholder ; valuep ; dty ; indp ; alenp ; rcodep ; maxarr_len ; mode) -> Function result

Parameter	Type		Description
stmtp	Longint	⇒	Statement handle
		⇐	
bindpp	Longint	⇒	Address of bind handle
		⇐	
errhp	Longint	⇒	Error handle
		⇐	
placeholder	String	⇒	Name of placeholder
valuep	Pointer	⇒	Address of data value(s) of type specified in dty parameter
		⇐	
dty	Longint	⇒	Datatype of value(s) being bound
indp	Pointer	⇒	Indicator variable or array
		⇐	
alenp	Pointer	⇒	Array of actual lengths of array elements
		⇐	
rcodep	Pointer	⇐	Array of column-level return codes
maxarr_len	Longint	⇒	Maximum array length parameter
mode	Longint	⇒	Specifies mode of operation
Function result	Longint	⇐	Status

Description

Please refer to the **OCIBindByName** command in the OCI documentation provided by Oracle.

Example

See the examples for **Executing an SQL INSERT request**, **Executing an SQL UPDATE request** and **Executing an SQL DELETE request**.

OCIBindByPos

OCIBindByPos (stmtp ; bindpp ; errhp ; position ; valuep ; dty ; indp ; alenp ; rcodep ; maxarr_len ; mode) -> Function result

Parameter	Type		Description
stmtp	Longint	⇒ ⇐	Statement handle
bindpp	Longint	⇒ ⇐	Address of bind handle
errhp	Longint	⇒ ⇐	Error handle
position	Longint	⇒	Specifies placeholder attributes
valuep	Pointer	⇒ ⇐	Address of data value(s) of type specified in dty parameter
dty	Longint	⇒	Datatype of value(s) being bound
indp	Pointer	⇒ ⇐	Indicator variable or array
alenp	Pointer	⇒ ⇐	Array of actual lengths of array elements
rcodep	Pointer	⇐	Array of column-level return codes
maxarr_len	Longint	⇒	Maximum array length parameter
mode	Longint	⇒	Specifies mode of operation
Function result	Longint	↩	Status

Description

Please refer to the **OCIBindByPos** command in the OCI documentation provided by Oracle.

Example

See the examples for **Writing in a LONG RAW column** and **Executing an SQL INSERT request**.

OCIDefineByPos

OCIDefineByPos (stmtp ; defnnpp ; errhp ; position ; valuep ; dty ; indp ; rlenp ; rcodep ; mode) -> Function result

Parameter	Type		Description
stmtp	Longint	⇒	Handle to requested SQL query operation
		⇐	
defnnpp	Longint	⇒	Define handle
		⇐	
errhp	Longint	⇒	Error handle
		⇐	
position	Longint	⇒	Position of value in list
valuep	Pointer	⇒	Buffer(s) of type specified in dty parameter
		⇐	
dty	Longint	⇒	Datatype
indp	Pointer	⇒	Indicator variable or array
rlenp	Pointer	⇒	Array of length of data fetched
		⇐	
rcodep	Pointer	⇐	Array of column-level return codes
mode	Longint	⇒	Specifies mode of operation
Function result	Longint	⇐	Status

Description

Please refer to the **OCIDefineByPos** command in the OCI documentation provided by Oracle.

Example

See the examples for **Reading from an Oracle BLOB column**, **Working with Oracle Date types**, and **Reading from a LONG RAW column**.

OCIDescribeAnyText

OCIDescribeAnyText (svchp ; errhp ; objname ; objptr_typ ; info_level ; objtyp ; dschp) -> Function result

Parameter	Type		Description
svchp	Longint	➡	Service context handle
errhp	Longint	➡	Error handle
objname	String	➡	Object to be described
objptr_typ	Longint	➡	Type of object
info_level	Longint	➡	Reserved for future extensions; pass OCI_DEFAULT
objtyp	Longint	➡	Type of schema object being described
dschp	Longint	➡	Handle describing object
Function result	Longint	↩	Status

Description

Please refer to the **OCIDescribeAny** command in the OCI documentation provided by Oracle.

OCIDescribeAnyVal

OCIDescribeAnyVal (svchp ; errhp ; objptr ; objptr_typ ; info_level ; objtyp ; dschp) -> Function result

Parameter	Type		Description
svchp	Longint	➡	Service context handle
errhp	Longint	➡	Error handle
objptr	Longint	➡	Object to be described
objptr_typ	Longint	➡	Type of object
info_level	Longint	➡	Reserved for future extensions; pass OCI_DEFAULT
objtyp	Longint	➡	Type of schema object being described
dschp	Longint	➡	Handle describing object
Function result	Longint	↩	Status

Description

Please refer to the **OCIDescribeAny** command in the OCI documentation provided by Oracle.

OCIDescriptorAlloc

OCIDescriptorAlloc (parenth ; descpp ; type) -> Function result

Parameter	Type		Description
parenth	Longint	⇒	Environment handle
descpp	Longint	⇐	Descriptor of desired type
type	Longint	⇒	Specifies type of descriptor to be allocated
Function result	Longint	⇒	Status

Description

Please refer to the **OCIDescriptorAlloc** command in the OCI documentation provided by Oracle.

OCIDescriptorFree

OCIDescriptorFree (descp) -> Function result

Parameter	Type		Description
descp	Longint	⇒	Descriptor to be deallocated
Function result	Longint	⇐	Status

Description

Please refer to the **OCIDescriptorFree** command in the OCI documentation provided by Oracle.

OCIEnvCreate

OCIEnvCreate (envhpp ; mode) -> Function result

Parameter	Type		Description
envhpp	Longint	⇦	Pointer to environment handle
mode	Longint	⇨	Specifies encoding mode
Function result	Longint	⇨	Status

Description

Please refer to the **OCIEnvCreate** command in the OCI documentation provided by Oracle.

Example

See the examples for **Connecting to an Oracle database** and **Reading from an Oracle BLOB column**.

OCIErrorGet

OCIErrorGet (hndlp ; recordno ; errcodep ; bufp) -> Function result

Parameter	Type		Description
hndlp	Longint	⇒	Error or environment handle
recordno	Longint	⇒	Record whose error you want to get
errcodep	Longint	⇐	Error code
bufp	String	⇐	Error message text
Function result	Longint	↩	Status

Description

Please refer to the **OCIErrorGet** command in the OCI documentation provided by Oracle.

OCIHandleAlloc

OCIHandleAlloc (parenth ; hndlpp ; type) -> Function result

Parameter	Type		Description
parenth	Longint	⇒	Environment handle
hndlpp	Longint	⇐	Returns a handle
type	Longint	⇒	Type of handle to be allocated
Function result	Longint	⇒	Status

Description

Please refer to the **OCIHandleAlloc** command in the OCI documentation provided by Oracle.

OCIHandleFree

OCIHandleFree (hndlp) -> Function result

Parameter	Type		Description
hndlp	Longint	⇨	Handle to be deallocated
Function result	Longint	⮌	Status

Description

Please refer to the **OCIHandleFree** command in the OCI documentation provided by Oracle.

Example

See the example for **Closing a connection**.

OCIServerVersion

OCIServerVersion (hndlp ; errhp ; bufp) -> Function result

Parameter	Type		Description
hndlp	Longint	➡	Service or server context handle
errhp	Longint	➡	Error handle
bufp	Longint	➡	Buffer in which version information is returned
Function result	Longint	↩	Status

Description

Please refer to the **OCIServerVersion** command in the OCI documentation provided by Oracle.

OCIStmtExecute

OCIStmtExecute (svchp ; stmtp ; errhp ; iters ; rowoff ; snap_in ; snap_out ; mode) -> Function result

Parameter	Type		Description
svchp	Longint	⇒ ⇐	Service context handle
stmtp	Longint	⇒ ⇐	Statement handle
errhp	Longint	⇒ ⇐	Error handle
iters	Longint	⇒	Number of times statement is executed
rowoff	Longint	⇒	Starting index from which data in array bind is relevant
snap_in	Longint	⇒	Database snapshot where statement is being executed
snap_out	Longint	⇐	Database snapshot suitable as a snap_in input to a subsequent call
mode	Longint	⇒	Mode of execution
Function result	Longint	↩	Status

Description

Please refer to the **OCIStmtExecute** command in the OCI documentation provided by Oracle.

Example

See the examples for **Executing an SQL SELECT request**, **Executing an SQL INSERT request**, **Executing an SQL UPDATE request** and **Executing an SQL DELETE request**.

OCIStmtFetch

OCIStmtFetch (stmtp ; errhp ; nrows) -> Function result

Parameter	Type		Description
stmtp	Longint	⇒	Statement handle
errhp	Longint	⇒	Error handle
nrows	Longint	⇒	Number of rows to be fetched from current position
Function result	Longint	⇐	Status

Description

Please refer to the **OCIStmtFetch** command in the OCI documentation provided by Oracle.

OCIStmtGetBindInfo

OCIStmtGetBindInfo (stmtp ; errhp ; stmt ; mode) -> Function result

Parameter	Type		Description
stmtp	Longint	⇒	Statement handle
errhp	Longint	⇒	Error handle
stmt	Text	⇒	Statement to be executed
mode	Longint	⇒	Specifies default encoding mode
Function result	Longint	⇐	Status

Description

Please refer to the **OCIStmtGetBindInfo** command in the OCI documentation provided by Oracle.

OCIStmtPrepare

OCIStmtPrepare (stmtp ; errhp ; stmt ; mode) -> Function result

Parameter	Type		Description
stmtp	Longint	⇒	Statement handle
errhp	Longint	⇒	Error handle
stmt	Text	⇒	Statement to be executed
mode	Longint	⇒	Specifies default encoding mode
Function result	Longint	⇐	Status

Description

Please refer to the **OCIStmtPrepare** command in the OCI documentation provided by Oracle.

Example

See the examples for **Executing an SQL SELECT request**, **Executing an SQL INSERT request**, **Executing an SQL UPDATE request** and **Executing an SQL DELETE request**.

OCITerminate

OCITerminate -> Function result

Parameter	Type	Description
Function result	Longint	Status

Description

Please refer to the **OCITerminate** command in the OCI documentation provided by Oracle.

Extras

- OCIGetTnsnamesPath
- OCIOnErrCall
- *_o_OCICleanUp*

OCIGetTnsnamesPath

OCIGetTnsnamesPath -> Function result

Parameter	Type	Description
Function result	String	Pathname of Oracle home folder

Description

The **OCIGetTnsnamesPath** command returns the full path of the Oracle home folder.

This function is especially useful if you need to access certain files from the Oracle home folder.

Example

Get the list of entry points contained in the "tnsnames.ora" file.

```
$TnsPath:=OCIGetTnsnamesPath
oci_tool_GetEntryName($TnsPath+"Network"\Admin\tnsnames";->oci_tTnsNameEntry)
```

oci_tool_GetEntryName is a 4D method that allows analyzing the "tnsnames.ora" file to extract entry points and save them in a text array.

OCIOnErrCall

OCIOnErrCall (methodName ; $1 ; $2)

Parameter	Type		Description
methodName	String	⇒	Name of error-handling method
$1	Longint	⇒	First argument to pass to method
$2	Longint	⇒	Second argument to pass to method

Description

The **OCIOnErrCall** command installs an error-handling method that will be executed every time an error occurs. This allows you to manage possible execution errors and override default error handling.

methodName is the name of the method to install. To return to default behavior, pass an empty string such as **OCIOnErrCall**("").

4D for OCI passes two arguments, *$1* and *$2*, to your method. If you want to compile your database, you must declare the *$1* and *$2* variables using the **C_LONGINT** commands.

_o_OCICleanUp

_o_OCICleanUp -> Function result

Parameter	Type	Description
Function result	Longint	Count of handles removed

Description

The primary task of the **_o_OCICleanUp** command is to free up memory used by objects within the plug-in, particularly potentially large objects created during Binds and Defines.

Note: This command is obsolete and should not be used. Instead you can use **OCIHandleFree** and pass the OCI environment handle (*envhp*).

LOB

- OCIDurationBegin
- OCIDurationEnd
- OCILobAppend
- OCILobAssign
- OCILobCharSetForm
- OCILobCharSetId
- OCILobCopy
- OCILobCreateTemporary
- OCILobDisableBuffering
- OCILobEnableBuffering
- OCILobErase
- OCILobFileClose
- OCILobFileCloseAll
- OCILobFileExists
- OCILobFileGetName
- OCILobFileIsOpen
- OCILobFileOpen
- OCILobFileSetName
- OCILobFlushBuffer
- OCILobFreeTemporary
- OCILobGetChunkSize
- OCILobGetLength
- OCILobIsEqual
- OCILobIsTemporary
- OCILobLoadFromFile
- OCILobLocatorIsInit
- OCILobRead
- OCILobTrim
- OCILobWrite
- OCILobWriteAppend

OCIDurationBegin

OCIDurationBegin (env ; err ; svc ; parent) -> Function result

Parameter	Type		Description
env	Longint	⇒ ⇐	Environment handle
err	Longint	⇒ ⇐	Error handle
svc	Longint	⇒	Service context handle
parent	Longint	⇐	Duration number of parent duration
Function result	Longint	↩	Status

Description

Please refer to the **OCIDurationBegin** command in the OCI documentation provided by Oracle.

OCIDurationEnd

OCIDurationEnd (env ; err ; duration ; svc) -> Function result

Parameter	Type		Description
env	Longint	⇒ ⇐	Environment handle
err	Longint	⇒ ⇐	Error handle
duration	Longint	⇒	User duration
svc	Longint	⇒	Service context handle
Function result	Longint	⇐	Status

Description

Please refer to the **OCIDurationEnd** command in the OCI documentation provided by Oracle.

OCILobAppend

OCILobAppend (svchp ; errhp ; dst_locp ; src_locp) -> Function result

Parameter	Type		Description
svchp	Longint	⇒	Service context handle
errhp	Longint	⇒	Error handle
		⇐	
dst_locp	Longint	⇒	Destination LOB
		⇐	
src_locp	Longint	⇒	Source LOB
Function result	Longint	⇐	Status

Description

Please refer to the **OCILobAppend** command in the OCI documentation provided by Oracle.

OCILobAssign

OCILobAssign (envhp ; errhp ; src_locp ; dst_locp) -> Function result

Parameter	Type		Description
envhp	Longint	⇒	Environment handle
		⇐	
errhp	Longint	⇒	Error handle
		⇐	
src_locp	Longint	⇒	LOB to copy from
dst_locp	Longint	⇒	LOB to copy to
		⇐	
Function result	Longint	↩	Status

Description

Please refer to the **OCILobAssign** command in the OCI documentation provided by Oracle.

OCILobCharSetForm

OCILobCharSetForm (envhp ; errhp ; locp ; csfrm) -> Function result

Parameter	Type		Description
envhp	Longint	⇨ ⇦	Environment handle
errhp	Longint	⇨ ⇦	Error handle
locp	Longint	⇨	LOB for which to get character set form
csfrm	Longint	⇦	Character set form of input LOB
Function result	Longint	⇩	Status

Description

Please refer to the **OCILobCharSetForm** command in the OCI documentation provided by Oracle.

OCILobCharSetId

OCILobCharSetId (envhp ; errhp ; locp ; csid) -> Function result

Parameter	Type		Description
envhp	Longint	⇒ ⇐	Environment handle
errhp	Longint	⇒ ⇐	Error handle
locp	Longint	⇒	LOB for which to get character set ID
csid	Longint	⇐	LOB character set ID
Function result	Longint	↩	Status

Description

Please refer to the **OCILobCharSetId** command in the OCI documentation provided by Oracle.

OCILobCopy

OCILobCopy (svchp ; errhp ; dst_locp ; src_locp ; amount ; dst_offset ; src_offset) -> Function result

Parameter	Type		Description
svchp	Longint	➡	Service context handle
errhp	Longint	➡	Error handle
		⬅	
dst_locp	Longint	➡	Destination LOB
		⬅	
src_locp	Longint	➡	Source LOB
amount	Longint	➡	Number of characters to be copied from source LOB to destination LOB
dst_offset	Longint	➡	Absolute offset for destination LOB
src_offset	Longint	➡	Absolute offset for source LOB
Function result	Longint	↩	Status

Description

Please refer to the **OCILobCopy** command in the OCI documentation provided by Oracle.

OCILobCreateTemporary

OCILobCreateTemporary (svchp ; errhp ; locp ; csid ; csfrm ; lobtype ; cache ; duration) -> Function result

Parameter	Type		Description
svchp	Longint	⇒	Service context handle
errhp	Longint	⇒	Error handle
		⇐	
locp	Longint	⇒	Points to temporary LOB
		⇐	
csid	Longint	⇒	LOB character set ID
csfrm	Longint	⇒	LOB character set form
lobtype	Longint	⇒	Type of LOB to create
cache	Longint	⇒	True if temporary LOB should be read into cache; otherwise, False
duration	Longint	⇒	Duration of temporary LOB
Function result	Longint	⇐	Status

Description

Please refer to the **OCILobCreateTemporary** command in the OCI documentation provided by Oracle.

OCILobDisableBuffering

OCILobDisableBuffering (svchp ; errhp ; locp) -> Function result

Parameter	Type		Description
svchp	Longint	⇒	Service context handle
errhp	Longint	⇒	Error handle
		⇐	
locp	Longint	⇒	LOB for which buffering is disabled
		⇐	
Function result	Longint	⇐	Status

Description

Please refer to the **OCILobDisableBuffering** command in the OCI documentation provided by Oracle.

OCILobEnableBuffering

OCILobEnableBuffering (svchp ; errhp ; locp) -> Function result

Parameter	Type		Description
svchp	Longint	⇒	Service context handle
errhp	Longint	⇒	Error handle
		⇐	
locp	Longint	⇒	LOB for which buffering is enabled
		⇐	
Function result	Longint	⇐	Status

Description

Please refer to the **OCILobEnableBuffering** command in the OCI documentation provided by Oracle.

OCILobErase

OCILobErase (svchp ; errhp ; locp ; amount ; offset) -> Function result

Parameter	Type		Description
svchp	Longint	⇒	Service context handle
errhp	Longint	⇒	Error handle
		⇐	
locp	Longint	⇒	LOB where data is erased
		⇐	
amount	Longint	⇒	Number of characters to erase
		⇐	
offset	Longint	⇒	Absolute offset in characters from beginning of LOB value from which to start erasing data
Function result	Longint	⇐	Status

Description

Please refer to the **OCILobErase** command in the OCI documentation provided by Oracle.

OCILobFileClose

OCILobFileClose (svchp ; errhp ; filep) -> Function result

Parameter	Type		Description
svchp	Longint	⇒	Service context handle
errhp	Longint	⇒	Error handle
		⇐	
filep	Longint	⇒	File to be closed
		⇐	
Function result	Longint	↩	Status

Description

Please refer to the **OCILobFileClose** command in the OCI documentation provided by Oracle.

OCILobFileCloseAll

OCILobFileCloseAll (svchp ; errhp) -> Function result

Parameter	Type		Description
svchp	Longint	⇨	Service context handle
errhp	Longint	⇨	Error handle
		⇦	
Function result	Longint	⇦	Status

Description

Please refer to the **OCILobFileCloseAll** command in the OCI documentation provided by Oracle.

OCILobFileExists

OCILobFileExists (svchp ; errhp ; filep ; flag) -> Function result

Parameter	Type		Description
svchp	Longint	⇨	Service context handle
errhp	Longint	⇨	Error handle
		⇦	
filep	Longint	⇨	File to be tested
flag	Longint	⇦	True if file exists on server; otherwise, False
Function result	Longint	⇩	Status

Description

Please refer to the **OCILobFileExists** command in the OCI documentation provided by Oracle.

OCILobFileGetName

OCILobFileGetName (envhp ; errhp ; filep ; dir_alias ; filename) -> Function result

Parameter	Type		Description
envhp	Longint	⇒	Environment handle
		⇐	
errhp	Longint	⇒	Error handle
		⇐	
filep	Longint	⇒	File for which to get directory object and file name
dir_alias	String	⇐	Buffer where directory object name is placed
filename	String	⇐	Buffer where file name is placed
Function result	Longint	⇐	Status

Description

Please refer to the **OCILobFileGetName** command in the OCI documentation provided by Oracle.

OCILobFileIsOpen

OCILobFileIsOpen (svchp ; errhp ; filep ; flag) -> Function result

Parameter	Type		Description
svchp	Longint	⇒	Service context handle
errhp	Longint	⇒	Error handle
		⇐	
filep	Longint	⇒	File being examined
flag	Longint	⇐	True if file was opened; otherwise, False
Function result	Longint	⇐	Status

Description

Please refer to the **OCILobFileIsOpen** command in the OCI documentation provided by Oracle.

OCILobFileOpen

OCILobFileOpen (svchp ; errhp ; filep ; mode) -> Function result

Parameter	Type		Description
svchp	Longint	⇒	Service context handle
errhp	Longint	⇒	Error handle
		⇐	
filep	Longint	⇒	File to open
		⇐	
mode	Longint	⇒	Mode in which to open file
Function result	Longint	⇐	Status

Description

Please refer to the **OCILobFileOpen** command in the OCI documentation provided by Oracle.

OCILobFileSetName

OCILobFileSetName (envhp ; errhp ; filepp ; dir_alias ; filename) -> Function result

Parameter	Type		Description
envhp	Longint	⇒ ⇐	Environment handle
errhp	Longint	⇒ ⇐	Error handle
filepp	Longint	⇒ ⇐	File for which to set directory object and file name
dir_alias	String	⇒	Directory object name to set
filename	String	⇒	File name to set
Function result	Longint	⇐	Status

Description

Please refer to the **OCILobFileSetName** command in the OCI documentation provided by Oracle.

OCILobFlushBuffer

OCILobFlushBuffer (svchp ; errhp ; locp ; flag) -> Function result

Parameter	Type		Description
svchp	Longint	⇒ ⇐	Service context handle
errhp	Longint	⇒ ⇐	Error handle
locp	Longint	⇒ ⇐	LOB whose buffers will be flushed
flag	Longint	⇒	Set to free buffer resources for LOB after flush
Function result	Longint	↩	Status

Description

Please refer to the **OCILobFlushBuffer** command in the OCI documentation provided by Oracle.

OCILobFreeTemporary

OCILobFreeTemporary (svchp ; errhp ; locp) -> Function result

Parameter	Type		Description
svchp	Longint		Service context handle
errhp	Longint		Error handle
locp	Longint		LOB to be freed
Function result	Longint		Status

Description

Please refer to the **OCILobFreeTemporary** command in the OCI documentation provided by Oracle.

OCILobGetChunkSize

OCILobGetChunkSize (svchp ; errhp ; locp ; chunk_size) -> Function result

Parameter	Type		Description
svchp	Longint	⇒	Service context handle
errhp	Longint	⇒	Error handle
		⇐	
locp	Longint	⇒	LOB for which to get chunk size
		⇐	
chunk_size	Longint	⇐	Size to be used when reading or writing the LOB value
Function result	Longint	↩	Status

Description

Please refer to the **OCILobGetChunkSize** command in the OCI documentation provided by Oracle.

OCILobGetLength

OCILobGetLength (svchp ; errhp ; locp ; lenp) -> Function result

Parameter	Type		Description
svchp	Longint	⇨	Service context handle
errhp	Longint	⇨	Error handle
		⇦	
locp	Longint	⇨	LOB whose length you want to get
lenp	Longint	⇦	Length of LOB
Function result	Longint	⇩	Status

Description

Please refer to the **OCILobGetLength** command in the OCI documentation provided by Oracle.

OCILobIsEqual

OCILobIsEqual (envhp ; x ; y ; is_equal) -> Function result

Parameter	Type		Description
envhp	Longint	⇒	Environment handle
x	Longint	⇒	LOB locator to compare
y	Longint	⇒	LOB locator to compare
is_equal	Longint	⇐	True if LOB locators are equal; otherwise, False
Function result	Longint	⇐	Status

Description

Please refer to the **OCILobIsEqual** command in the OCI documentation provided by Oracle.

OCILobIsTemporary

OCILobIsTemporary (envhp ; errhp ; locp ; is_temporary) -> Function result

Parameter	Type		Description
envhp	Longint	⇒	Environment handle
errhp	Longint	⇒	Error handle
		⇐	
locp	Longint	⇒	LOB to test
is_temporary	Longint	⇐	True if LOB is temporary; otherwise, False
Function result	Longint	↩	Status

Description

Please refer to the **OCILobIsTemporary** command in the OCI documentation provided by Oracle.

OCILobLoadFromFile

OCILobLoadFromFile (svchp ; errhp ; dst_locp ; src_locp ; amount ; dst_offset ; src_offset) -> Function result

Parameter	Type		Description
svchp	Longint	⇒	Service context handle
errhp	Longint	⇒	Error handle
		⇐	
dst_locp	Longint	⇒	Destination LOB
		⇐	
src_locp	Longint	⇒	Source file
		⇐	
amount	Longint	⇒	Number of bytes to be loaded
dst_offset	Longint	⇒	Absolute offset for destination LOB
src_offset	Longint	⇒	Absolute offset for source file
Function result	Longint	⇐	Status

Description

Please refer to the **OCILobLoadFromFile** command in the OCI documentation provided by Oracle.

OCILobLocatorIsInit

OCILobLocatorIsInit (envhp ; errhp ; locp ; is_initialized) -> Function result

Parameter	Type		Description
envhp	Longint	⇒ ⇐	Environment handle
errhp	Longint	⇒ ⇐	Error handle
locp	Longint	⇒	LOB being tested
is_initialized	Longint	⇐	True if LOB is initialized; otherwise, False
Function result	Longint	↩	Status

Description

Please refer to the **OCILobLocatorIsInit** command in the OCI documentation provided by Oracle.

OCILobRead

OCILobRead (svchp ; errhp ; locp ; offset ; bufp) -> Function result

Parameter	Type		Description
svchp	Longint	⇒ ⇐	Service context handle
errhp	Longint	⇒ ⇐	Error handle
locp	Longint	⇒	LOB to be read
offset	Longint	⇒	Absolute offset from beginning of LOB value
bufp	BLOB	⇒ ⇐	Buffer into which LOB will be read
Function result	Longint	⇐	Status

Description

Please refer to the **OCILobRead** command in the OCI documentation provided by Oracle.

Example

See the example for **Reading from an Oracle BLOB column**.

OCILobTrim

OCILobTrim (svchp ; errhp ; locp ; newlen) -> Function result

Parameter	Type		Description
svchp	Longint	⇒	Service context handle
errhp	Longint	⇒	Error handle
		⇐	
locp	Longint	⇒	LOB to be truncated
		⇐	
newlen	Longint	⇒	New length of LOB
Function result	Longint	⇐	Status

Description

Please refer to the **OCILobTrim** command in the OCI documentation provided by Oracle.

OCILobWrite

OCILobWrite (svchp ; errhp ; locp ; offset ; bufp) -> Function result

Parameter	Type		Description
svchp	Longint	⇨ ⇦	Service context handle
errhp	Longint	⇨ ⇦	Error handle
locp	Longint	⇨ ⇦	LOB to be written
offset	Longint	⇨	Absolute offset from beginning of LOB value
bufp	BLOB	⇨	Buffer from which LOB will be written
Function result	Longint	⇦	Status

Description

Please refer to the **OCILobWrite** command in the OCI documentation provided by Oracle.

OCILobWriteAppend

OCILobWriteAppend (svchp ; errhp ; locp ; bufp) -> Function result

Parameter	Type		Description
svchp	Longint	⇒	Service context handle
errhp	Longint	⇒	Error handle
		⇐	
locp	Longint	⇒	Unique reference for LOB to be modified
		⇐	
bufp	Longint	⇒	Pointer to buffer
Function result	Longint	⇐	Status

Description

Please refer to the **OCILobWriteAppend** command in the OCI documentation provided by Oracle.

Math

- OCINumberAdd
- OCINumberArcCos
- OCINumberArcSin
- OCINumberArcTan
- OCINumberArcTan2
- OCINumberCos
- OCINumberDiv
- OCINumberExp
- OCINumberFromText
- OCINumberHypCos
- OCINumberHypSin
- OCINumberHypTan
- OCINumberIntPower
- OCINumberLn
- OCINumberLog
- OCINumberMul
- OCINumberPower
- OCINumberRound
- OCINumberSin
- OCINumberSqrt
- OCINumberSub
- OCINumberTan
- OCINumberToText
- OCINumberTrunc

OCINumberAdd

OCINumberAdd (err ; number1 ; number2 ; result) -> Function result

Parameter	Type		Description
err	Longint	⇒ ⇐	Error handle
number1	Real	⇒	Number to be added to number2
number2	Real	⇒	Number to be added to number1
result	Real	⇐	Result of adding number 1 and number 2
Function result	Longint	⇐	Status

Description

Please refer to the **OCINumberAdd** command in the OCI documentation provided by Oracle.

OCINumberArcCos

OCINumberArcCos (err ; number ; result) -> Function result

Parameter	Type		Description
err	Longint	⇒ ⇐	Error handle
number	Real	⇒	Argument of arc cosine
result	Real	⇐	Result of arc cosine in radians
Function result	Longint	↩	Status

Description

Please refer to the **OCINumberArcCos** command in the OCI documentation provided by Oracle.

OCINumberArcSin

OCINumberArcSin (err ; number ; result) -> Function result

Parameter	Type		Description
err	Longint	⇒	Error handle
		⇐	
number	Real	⇒	Argument of arc sine
result	Real	⇐	Result of arc sine in radians
Function result	Longint	⇐	Status

Description

Please refer to the **OCINumberArcSin** command in the OCI documentation provided by Oracle.

OCINumberArcTan

OCINumberArcTan (err ; number ; result) -> Function result

Parameter	Type		Description
err	Longint	⇨ ⇦	Error handle
number	Real	⇨	Argument of arc tangent
result	Real	⇦	Result of arc tangent in radians
Function result	Longint	⇩	Status

Description

Please refer to the **OCINumberArcTan** command in the OCI documentation provided by Oracle.

OCINumberArcTan2

OCINumberArcTan2 (err ; number1 ; number2 ; result) -> Function result

Parameter	Type		Description
err	Longint	⇒	Error handle
		⇐	
number1	Real	⇒	Argument 1 of arc tangent
number2	Real	⇒	Argument 2 of arc tangent
result	Real	⇐	Result of arc tangent in radians
Function result	Longint	↩	Status

Description

Please refer to the **OCINumberArcTan2** command in the OCI documentation provided by Oracle.

OCINumberCos

OCINumberCos (err ; number ; result) -> Function result

Parameter	Type		Description
err	Longint	⇒ ⇐	Error handle
number	Real	⇒	Argument of cosine in radians
result	Real	⇐	Result of cosine in radians
Function result	Longint	↩	Status

Description

Please refer to the **OCINumberCos** command in the OCI documentation provided by Oracle.

OCINumberDiv

OCINumberDiv (err ; number1 ; number2 ; result) -> Function result

Parameter	Type		Description
err	Longint	⇨ ⇦	Error handle
number1	Real	⇨	Numerator
number2	Real	⇨	Denominator
result	Real	⇦	Division result
Function result	Longint	⇦	Status

Description

Please refer to the **OCINumberDiv** command in the OCI documentation provided by Oracle.

OCINumberExp

OCINumberExp (err ; number ; result) -> Function result

Parameter	Type		Description
err	Longint	⇒ ⇐	Error handle
number	Real	⇒	Power to which to raise e
result	Real	⇐	Result of exponentiation
Function result	Longint	↩	Status

Description

Please refer to the **OCINumberExp** command in the OCI documentation provided by Oracle.

OCINumberFromText

OCINumberFromText (err ; str ; fmt ; nls_params ; number) -> Function result

Parameter	Type		Description
err	Longint	⇒ ⇐	Error handle
str	String	⇒	Text to convert to number
fmt	String	⇒	Conversion format
nls_params	String	⇒	Global Support format specification
number	Real	⇐	Result of text converted to number
Function result	Longint	⇐	Status

Description

Please refer to the **OCINumberFromText** command in the OCI documentation provided by Oracle.

OCINumberHypCos

OCINumberHypCos (err ; number ; result) -> Function result

Parameter	Type		Description
err	Longint	⇒	Error handle
		⇐	
number	Real	⇒	Argument of cosine hyperbolic
result	Real	⇐	Result of cosine hyperbolic
Function result	Longint	↩	Status

Description

Please refer to the **OCINumberHypCos** command in the OCI documentation provided by Oracle.

OCINumberHypSin

OCINumberHypSin (err ; number ; result) -> Function result

Parameter	Type		Description
err	Longint	⇨ ⇦	Error handle
number	Real	⇨	Argument of sine hyperbolic
result	Real	⇦	Result of sine hyperbolic
Function result	Longint	⇩	Status

Description

Please refer to the **OCINumberHypSin** command in the OCI documentation provided by Oracle.

OCINumberHypTan

OCINumberHypTan (err ; number ; result) -> Function result

Parameter	Type		Description
err	Longint	⇒ ⇐	Error handle
number	Real	⇒	Argument of tangent hyperbolic
result	Real	⇐	Result of tangent hyperbolic
Function result	Longint	↩	Status

Description

Please refer to the **OCINumberHypTan** command in the OCI documentation provided by Oracle.

OCINumberIntPower

OCINumberIntPower (err ; base ; exp ; result) -> Function result

Parameter	Type		Description
err	Longint	⇒	Error handle
		⇐	
base	Real	⇒	Base of exponentiation
exp	Real	⇒	Exponent to which base is raised
result	Real	⇐	Result of exponentiation
Function result	Longint	⇐	Status

Description

Please refer to the **OCINumberIntPower** command in the OCI documentation provided by Oracle.

OCINumberLn

OCINumberLn (err ; number ; result) -> Function result

Parameter	Type		Description
err	Longint	⇒ ⇐	Error handle
number	Real	⇒	Number for which logarithm is computed
result	Real	⇐	Logarithm result
Function result	Longint	↩	Status

Description

Please refer to the **OCINumberLn** command in the OCI documentation provided by Oracle.

OCINumberLog

OCINumberLog (err ; base ; number ; result) -> Function result

Parameter	Type		Description
err	Longint	⇒ ⇐	Error handle
base	Real	⇒	Base of logarithm
number	Real	⇒	Operand
result	Real	⇐	Logarithm result
Function result	Longint	↩	Status

Description

Please refer to the **OCINumberLog** command in the OCI documentation provided by Oracle.

OCINumberMul

OCINumberMul (err ; number1 ; number2 ; result) -> Function result

Parameter	Type		Description
err	Longint	⇒	Error handle
		⇐	
number1	Real	⇒	Number to be multiplied by number2
number2	Real	⇒	Number by which number1 is multiplied
result	Real	⇐	Result of multiplication
Function result	Longint	↩	Status

Description

Please refer to the **OCINumberMul** command in the OCI documentation provided by Oracle.

OCINumberPower

OCINumberPower (err ; base ; exp ; result) -> Function result

Parameter	Type		Description
err	Longint	⇒	Error handle
		⇐	
base	Real	⇒	Base of exponentiation
exp	Real	⇒	Exponent to which base is to be raised
result	Real	⇐	Result of exponentiation
Function result	Longint	↩	Status

Description

Please refer to the **OCINumberPower** command in the OCI documentation provided by Oracle.

OCINumberRound

OCINumberRound (err ; number ; decplace ; result) -> Function result

Parameter	Type		Description
err	Longint	⇒	Error handle
		⇐	
number	Real	⇒	Number to round
decplace	Longint	⇒	Number of digits to right of decimal point to round to.
result	Real	⇐	Result of rounding
Function result	Longint	↩	Status

Description

Please refer to the **OCINumberRound** command in the OCI documentation provided by Oracle.

OCINumberSin

OCINumberSin (err ; number ; result) -> Function result

Parameter	Type		Description
err	Longint	⇒ ⇐	Error handle
number	Real	⇒	Argument of sine in radians
result	Real	⇐	Result of sine
Function result	Longint	⇐	Status

Description

Please refer to the **OCINumberSin** command in the OCI documentation provided by Oracle.

OCINumberSqrt

OCINumberSqrt (err ; number ; result) -> Function result

Parameter	Type		Description
err	Longint	⇒ ⇐	Error handle
number	Real	⇒	Input number
result	Real	⇐	Square root of number
Function result	Longint	↩	Status

Description

Please refer to the **OCINumberSqrt** command in the OCI documentation provided by Oracle.

OCINumberSub

OCINumberSub (err ; number1 ; number2) -> Function result

Parameter	Type		Description
err	Longint	⇒ ⇐	Error handle
number1	Real	⇒	Number from which number2 is subtracted
number2	Real	⇒	Number which is subtracted from number1
Function result	Longint	⇐	Status

Description

Please refer to the **OCINumberSub** command in the OCI documentation provided by Oracle.

OCINumberTan

OCINumberTan (err ; number ; result) -> Function result

Parameter	Type		Description
err	Longint	⇨ ⇦	Error handle
number	Real	⇨	Argument of tangent in radians
result	Longint	⇦	Result of tangent
Function result	Longint	⇨	Status

Description

Please refer to the **OCINumberTan** command in the OCI documentation provided by Oracle.

OCINumberToText

OCINumberToText (err ; number ; fmt ; nls_params ; buf) -> Function result

Parameter	Type		Description
err	Longint	⇒	Error handle
		⇐	
number	Real	⇒	Number to convert
fmt	String	⇒	Conversion format
nls_params	String	⇒	Global Support format specification
buf	String	⇐	Buffer where converted string is placed
Function result	Longint	⇐	Status

Description

Please refer to the **OCINumberToText** command in the OCI documentation provided by Oracle.

OCINumberTrunc

OCINumberTrunc (err ; number ; decplace ; result) -> Function result

Parameter	Type		Description
err	Longint	⇒	Error handle
		⇐	
number	Real	⇒	Input number
decplace	Longint	⇒	Number of digits to right of decimal point at which to truncate
result	Real	⇐	Result of truncation
Function result	Longint	⇐	Status

Description

Please refer to the **OCINumberTrunc** command in the OCI documentation provided by Oracle.

Relational Commands

- OCIBindDateByName
- OCIBindDateByPos
- OCICollAppend
- OCICollAssign
- OCICollAssignElem
- OCICollGetElem
- OCICollMax
- OCICollSize
- OCICollTrim
- OCIDateAddDays
- OCIDateAddMonths
- OCIDateFromText
- OCIDateLastDay
- OCIDateNextDay
- OCIDateSysDate
- OCIDateToText
- OCIDateZoneToZone
- OCIDefineDateByPos
- OCIIterCreate
- OCIIterDelete
- OCIIterGetCurrent
- OCIIterInit
- OCIIterNext
- OCIIterPrev
- OCIRawAllocSize
- OCIRawAssignBytes
- OCIRawAssignRaw
- OCIRawPtr
- OCIRawResize
- OCIRawSize
- OCIRefAssign
- OCIRefClear
- OCIRefFromHex
- OCIRefHexSize
- OCIRefIsEqual
- OCIRefIsNull
- OCIRefToHex
- OCITableDelete
- OCITableExists
- OCITableFirst
- OCITableLast
- OCITableNext
- OCITablePrev
- OCITableSize

OCIBindDateByName

OCIBindDateByName (stmtp ; bindpp ; errhp ; placeholder ; valuep ; valuep2 ; dty ; indp ; alenp ; rcodep ; maxarr_len ; mode) -> Function result

Parameter	Type		Description
stmtp	Longint	⇒	Statement handle
		⇐	
bindpp	Longint	⇒	Address of bind handle
		⇐	
errhp	Longint	⇒	Error handle
		⇐	
placeholder	String	⇒	Name of placeholder
valuep	Pointer	⇒	Address of data value(s) of type specified in dty parameter
		⇐	
valuep2	Pointer	⇒	Address of data value(s) of type specified in dty parameter
		⇐	
dty	Longint	⇒	Datatype of value(s) being bound
indp	Pointer	⇒	Indicator variable or array
		⇐	
alenp	Pointer	⇒	Array of actual lengths of array elements
		⇐	
rcodep	Pointer	⇐	Array of column-level return codes
maxarr_len	Longint	⇒	Maximum array length parameter
mode	Longint	⇒	Specifies mode of operation
Function result	Longint	⇒	Status

Description

Please refer to the **OCIBindDateByName** command in the OCI documentation provided by Oracle.

Note: The **OCIBindDateByName** command is similar to the **OCIBindByName** command except that there is an additional *valuep2* parameter to adapt it for working with Oracle columns of the TIMESTAMP type. For this command, the *valuep* parameter is a pointer to a date (C_DATE) and *valuep2* is a pointer to a time (C_TIME). When both parameters are used in a bind, the command takes the date (day/month/year) from the *valuep* parameter and the time (hour:minute:second) from the *valuep2* parameter.

For more information about handling dates, see **Working with Oracle Date types**.

OCIBindDateByPos

OCIBindDateByPos (stmtp ; bindpp ; errhp ; position ; valuep ; valuep2 ; dty ; indp ; alenp ; rcodep ; maxarr_len ; mode) -> Function result

Parameter	Type		Description
stmtp	Longint	⇒	Statement handle
		⇐	
bindpp	Longint	⇒	Address of bind handle
		⇐	
errhp	Longint	⇒	Error handle
		⇐	
position	Longint	⇒	Specifies placeholder attributes
valuep	Pointer	⇒	Address of data value(s) of type specified in dty parameter
		⇐	
valuep2	Pointer	⇒	Address of data value(s) of type specified in dty parameter
		⇐	
dty	Longint	⇒	Datatype of value(s) being bound
indp	Pointer	⇒	Indicator variable or array
		⇐	
alenp	Pointer	⇒	Array of actual lengths of array elements
		⇐	
rcodep	Pointer	⇐	Array of column-level return codes
maxarr_len	Longint	⇒	Maximum array length parameter
mode	Longint	⇒	Specifies mode of operation
Function result		⇒	Status

Description

Please refer to the **OCIBindDateByPos** command in the OCI documentation provided by Oracle.

Note: The **OCIBindDateByPos** command is similar to the **OCIBindByPos** command except that there is an additional *valuep2* parameter to adapt it for working with Oracle columns of the TIMESTAMP type. For this command, the *valuep* parameter is a pointer to a date (C_DATE) and *valuep2* is a pointer to a time (C_TIME). When both parameters are used in a bind, the command takes the date (day/month/year) from the *valuep* parameter and the time (hour:minute:second) from the *valuep2* parameter.

For more information about handling dates, see **Working with Oracle Date types**.

Example

See the example for **Executing an SQL INSERT request**.

OCICollAppend

OCICollAppend (env ; err ; elem ; elemind ; coll) -> Function result

Parameter	Type		Description
env	Longint	⇒ ⇐	OCI environment handle
err	Longint	⇒ ⇐	Error handle
elem	Longint	⇒	Element appended to end of collection
elemind	Longint	⇒	Pointer to element's NULL indicator
coll	Longint	⇒ ⇐	Updated collection
Function result	Longint	⇐	Status

Description

Please refer to the **OCICollAppend** command in the OCI documentation provided by Oracle.

OCICollAssign

OCICollAssign (env ; err ; rhs ; lhs) -> Function result

Parameter	Type		Description
env	Longint	⇒ ⇐	OCI environment handle
err	Longint	⇒ ⇐	Error handle
rhs	Longint	⇒	Right-hand side (source) collection to be assigned from
lhs	Longint	⇐	Left-hand side (target) collection to be assigned to
Function result	Longint	↩	Status

Description

Please refer to the **OCICollAssign** command in the OCI documentation provided by Oracle.

OCICollAssignElem

OCICollAssignElem (env ; err ; index ; elem ; elemind ; coll) -> Function result

Parameter	Type		Description
env	Longint	⇒	OCI environment handle
		⇐	
err	Longint	⇒	Error handle
		⇐	
index	Longint	⇒	Index of element being assigned
elem	Longint	⇒	Source element
elemind	Longint	⇒	Pointer to element's NULL indicator
coll	Longint	⇒	Collection to be updated
		⇐	
Function result	Longint	↩	Status

Description

Please refer to the **OCICollAssignElem** command in the OCI documentation provided by Oracle.

OCICollGetElem

OCICollGetElem (env ; err ; coll ; index ; exists ; elem ; elemind) -> Function result

Parameter	Type		Description
env	Longint	⇒	OCI environment handle
		⇐	
err	Longint	⇒	Error handle
		⇐	
coll	Longint	⇒	Collection whose element you want to get
index	Longint	⇒	Index of element whose pointer is returned
exists	Longint	⇐	False if element does not exist; otherwise, True
elem	Longint	⇐	Address of element
elemind	Longint	⇐	Address of NULL indicator
Function result	Longint	⇐	Status

Description

Please refer to the **OCICollGetElem** command in the OCI documentation provided by Oracle.

OCICollMax

OCICollMax (env ; coll) -> Function result

Parameter	Type		Description
env	Longint	⇒	OCI environment handle
		⇐	
coll	Longint	⇒	Collection whose number of elements is returned
Function result	Longint	⇐	Status

Description

Please refer to the **OCICollMax** command in the OCI documentation provided by Oracle.

OCICollSize

OCICollSize (env ; err ; coll ; size) -> Function result

Parameter	Type		Description
env	Longint	⇒ ⇐	OCI environment handle
err	Longint	⇒ ⇐	Error handle
coll	Longint	⇒	Collection whose size is returned
size	Longint	⇐	Current number of elements in collection
Function result	Longint	⇐	Status

Description

Please refer to the **OCICollSize** command in the OCI documentation provided by Oracle.

OCICollTrim

OCICollTrim (env ; err ; trim_num ; coll) -> Function result

Parameter	Type		Description
env	Longint	⇒ ⇐	OCI environment handle
err	Longint	⇒ ⇐	Error handle
trim_num	Longint	⇒	Number of elements to trim
coll	Longint	⇒ ⇐	Collection from which elements are trimmed
Function result	Longint	↩	Status

Description

Please refer to the **OCICollTrim** command in the OCI documentation provided by Oracle.

OCIDateAddDays

OCIDateAddDays (err ; date ; num_days ; result) -> Function result

Parameter	Type		Description
err	Longint	⇒	Error handle
		⇐	
date	Date	⇒	Date from which to add or subtract
num_days	Longint	⇒	Number of days to be added or subtracted
result	Date	⇒	Result of adding/subtracting days to/from date
		⇐	
Function result	Longint	↩	Status

Description

Please refer to the **OCIDateAddDays** command in the OCI documentation provided by Oracle.

OCIDateAddMonths

OCIDateAddMonths (err ; date ; num_months ; result) -> Function result

Parameter	Type		Description
err	Longint	⇒	Error handle
		⇐	
date	Date	⇒	Date from which to add or subtract
num_months	Longint	⇒	Number of months to be added or subtracted
result	Date	⇒	Result of adding/subtracting days to/from date
		⇐	
Function result	Longint	↩	Status

Description

Please refer to the **OCIDateAddMonths** command in the OCI documentation provided by Oracle.

OCIDateFromText

OCIDateFromText (err ; date_str ; fmt ; lang_name ; date ; time) -> Function result

Parameter	Type		Description
err	Longint	⇒ ⇐	Error handle
date_str	String	⇒	String to be converted
fmt	String	⇒	Conversion format
lang_name	String	⇒	Language in which names of months and days are specified
date	Date	⇐	String converted to date
time	Time	⇐	String converted to time
Function result	Longint	↩	Status

Description

Please refer to the **OCIDateFromText** command in the OCI documentation provided by Oracle.

OCIDateLastDay

OCIDateLastDay (err ; date ; last_day) -> Function result

Parameter	Type		Description
err	Longint	⇒ ⇐	Error handle
date	Date	⇒	Input date
last_day	String	⇐	Last day of month in date
Function result	Longint	↩	Status

Description

Please refer to the **OCIDateLastDay** command in the OCI documentation provided by Oracle.

OCIDateNextDay

OCIDateNextDay (err ; date ; day ; next_day) -> Function result

Parameter	Type		Description
err	Longint	⇒	Error handle
		⇐	
date	Date	⇒	Returned date must be after this date
day	String	⇒	Specifies first day of week
next_day	Date	⇐	First day of week specified in day after specified date
Function result	Longint	⇐	Status

Description

Please refer to the **OCIDateNextDay** command in the OCI documentation provided by Oracle.

OCIDateSysDate

OCIDateSysDate (err ; sys_date ; sys_time) -> Function result

Parameter	Type		Description
err	Longint	⇒ ⇐	Error handle
sys_date	Date	⇐	Current system date of client
sys_time	Time	⇐	Current system time of client
Function result	Longint	⇒	Status

Description

Please refer to the **OCIDateSysDate** command in the OCI documentation provided by Oracle.

OCIDateToText

OCIDateToText (err ; date ; time ; fmt ; lang_name ; buf) -> Function result

Parameter	Type		Description
err	Longint	⇒	Error handle
		⇐	
date	Date	⇒	Date to be converted
time	Time	⇒	Time to be converted
fmt	String	⇒	Conversion format
lang_name	String	⇒	Language in which names of months and days are returned
buf	Date	⇐	Buffer where converted string is placed
Function result	Longint	⇐	Status

Description

Please refer to the **OCIDateToText** command in the OCI documentation provided by Oracle.

OCIDateZoneToZone

OCIDateZoneToZone (err ; date1 ; time1 ; zon1 ; zon2 ; date2 ; time2) -> Function result

Parameter	Type		Description
err	Longint	⇒ ⇐	Error handle
date1	Date	⇒	Date to convert
time1	Time	⇒	Time to convert
zon1	String	⇒	Zone of input date
zon2	String	⇒	Zone to be converted to
date2	Date	⇐	Converted date
time2	Time	⇐	Converted time
Function result	Longint	⇐	Status

Description

Please refer to the **OCIDateZoneToZone** command in the OCI documentation provided by Oracle.

OCIDefineDateByPos

OCIDefineDateByPos (stmtp ; defnnpp ; errhp ; position ; valuep ; valuep2 ; dty ; indp ; rlenp ; rcodep ; mode) -> Function result

Parameter	Type		Description
stmtp	Longint	⇒	Handle to requested SQL query operation
		⇐	
defnnpp	Longint	⇒	Define handle
		⇐	
errhp	Longint	⇒	Error handle
		⇐	
position	Longint	⇒	Position of value in list
valuep	Pointer	⇒	Buffer(s) of type specified in dty parameter
		⇐	
valuep2	Pointer	⇒	Buffer(s) of type specified in dty parameter
		⇐	
dty	Longint	⇒	Datatype
indp	Pointer	⇒	Indicator variable or array
rlenp	Pointer	⇒	Array of length of data fetched
		⇐	
rcodep	Pointer	⇐	Array of column-level return codes
mode	Longint	⇒	Specifies mode of operation
Function result	Longint	⇐	Status

Description

Please refer to the **OCIDefineDateByPos** command in the OCI documentation provided by Oracle for more information.

Note: The **OCIDefineDateByPos** command is similar to the **OCIDefineByPos** command except that there is an additional *valuep2* parameter to adapt it for working with Oracle columns of the TIMESTAMP type. For this command, the *valuep* parameter is a pointer to a date (C_DATE) and *valuep2* is a pointer to a time (C_TIME). When both parameters are used, the command takes the date (day/month/year) from the *valuep* parameter and the time (hour:minute:second) from the *valuep2* parameter.

Example

See the example for **Working with Oracle Date types**.

OCIIterCreate

OCIIterCreate (env ; err ; coll ; itr) -> Function result

Parameter	Type		Description
env	Longint	⇒	OCI environment handle
		⇐	
err	Longint	⇒	Error handle
		⇐	
coll	Longint	⇒	Collection to be scanned
itr	Longint	⇐	Address of allocated collection iterator
Function result	Longint	↩	Status

Description

Please refer to the **OCIIterCreate** command in the OCI documentation provided by Oracle.

OCIIterDelete

OCIIterDelete (env ; err ; itr) -> Function result

Parameter	Type		Description
env	Longint	⇒ ⇐	OCI environment handle
err	Longint	⇒ ⇐	Error handle
itr	Longint	⇒ ⇐	Iterator to be deleted
Function result	Longint	↩	Status

Description

Please refer to the **OCIIterDelete** command in the OCI documentation provided by Oracle.

OCIIterGetCurrent

OCIIterGetCurrent (env ; err ; itr ; elem) -> Function result

Parameter	Type		Description
env	Longint	⇒ ⇐	OCI environment handle
err	Longint	⇒ ⇐	Error handle
itr	Longint	⇒	Iterator pointing to current element
elem	Longint	⇐	Address of element pointed by iterator
Function result	Longint	↩	Status

Description

Please refer to the **OCIIterGetCurrent** command in the OCI documentation provided by Oracle.

OCIIterInit

OCIIterInit (env ; err ; coll ; itr) -> Function result

Parameter	Type		Description
env	Longint	⇒	OCI environment handle
		⇐	
err	Longint	⇒	Error handle
		⇐	
coll	Longint	⇒	Collection to be scanned
itr	Longint	⇒	Pointer to allocated collection iterator
		⇐	
Function result	Longint	⇐	Status

Description

Please refer to the **OCIIterInit** command in the OCI documentation provided by Oracle.

OCIIterNext

OCIIterNext (env ; err ; itr ; elem ; elemind ; eoc) -> Function result

Parameter	Type		Description
env	Longint	⇒ ⇐	OCI environment handle
err	Longint	⇒ ⇐	Error handle
itr	Longint	⇒ ⇐	Iterator updated to point to next element
elem	Longint	⇐	Address of next element
elemind	Longint	⇐	Address of element's NULL indicator
eoc	Longint	⇐	True if next element does not exist; otherwise, False
Function result	Longint	⇐	Status

Description

Please refer to the **OCIIterNext** command in the OCI documentation provided by Oracle.

OCIIterPrev

OCIIterPrev (env ; err ; itr ; elem ; elemind ; boc) -> Function result

Parameter	Type		Description
env	Longint	⇒ ⇐	OCI environment handle
err	Longint	⇒ ⇐	Error handle
itr	Longint	⇒ ⇐	Iterator updated to point to previous element
elem	Longint	⇐	Address of previous element
elemind	Longint	⇐	Address of element's NULL indicator
boc	Longint	⇐	True if previous element does not exist; otherwise, False
Function result	Longint	⇐	Status

Description

Please refer to the **OCIIterPrev** command in the OCI documentation provided by Oracle.

OCIRawAllocSize

OCIRawAllocSize (env ; err ; raw ; allocsize) -> Function result

Parameter	Type		Description
env	Longint	⇒ ⇐	OCI environment handle
err	Longint	⇒ ⇐	Error handle
raw	Longint	⇒	Raw data whose allocated size is returned (in bytes)
allocsize	Longint	⇐	Allocated size of raw memory (in bytes)
Function result	Longint	↩	Status

Description

Please refer to the **OCIRawAllocSize** command in the OCI documentation provided by Oracle.

OCIRawAssignBytes

OCIRawAssignBytes (env ; err ; rhs ; lhs) -> Function result

Parameter	Type		Description
env	Longint	⇒ ⇐	OCI environment handle
err	Longint	⇒ ⇐	Error handle
rhs	BLOB	⇒	Right-hand side (source)
lhs	Longint	⇒ ⇐	Left-hand side (target)
Function result	Longint	↩	Status

Description

Please refer to the **OCIRawAssignBytes** command in the OCI documentation provided by Oracle.

OCIRawAssignRaw

OCIRawAssignRaw (env ; err ; rhs ; lhs) -> Function result

Parameter	Type		Description
env	Longint	⇒ ⇐	OCI environment handle
err	Longint	⇒ ⇐	Error handle
rhs	Longint	⇒	Right-hand side (source)
lhs	Longint	⇒ ⇐	Left-hand side (target)
Function result	Longint	↩	Status

Description

Please refer to the **OCIRawAssignRaw** command in the OCI documentation provided by Oracle.

OCIRawPtr

OCIRawPtr (env ; raw) -> Function result

Parameter	Type		Description
env	Longint	⇒ ⇐	OCI environment handle
raw	Longint	⇒	Raw for which a pointer is returned
Function result	BLOB	⇐	Pointer

Description

Please refer to the **OCIRawPtr** command in the OCI documentation provided by Oracle.

OCIRawResize

OCIRawResize (env ; err ; new_size ; raw) -> Function result

Parameter	Type		Description
env	Longint	⇒ ⇐	OCI environment handle
err	Longint	⇒ ⇐	Error handle
new_size	Longint	⇒	New size of raw data (in bytes)
raw	Longint	⇒	Raw to be resized
Function result	Longint	↩	Status

Description

Please refer to the **OCIRawResize** command in the OCI documentation provided by Oracle.

OCIRawSize

OCIRawSize (env ; raw) -> Function result

Parameter	Type		Description
env	Longint	⇨	OCI environment handle
raw	Longint	⇨ ⇦	Raw whose size is returned
Function result	Longint	⇦	Status

Description

Please refer to the **OCIRawSize** command in the OCI documentation provided by Oracle.

OCIRefAssign

OCIRefAssign (env ; err ; source ; target) -> Function result

Parameter	Type		Description
env	Longint	⇒	OCI environment handle
		⇐	
err	Longint	⇒	Error handle
		⇐	
source	Longint	⇒	Ref from which to copy
target	Longint	⇒	Ref to copy to
		⇐	
Function result	Longint	↩	Status

Description

Please refer to the **OCIRefAssign** command in the OCI documentation provided by Oracle.

OCIRefClear

OCIRefClear (env ; ref) -> Function result

Parameter	Type		Description
env	Longint	⇒ ⇐	OCI environment handle
ref	Longint	⇒ ⇐	Ref to clear
Function result	Longint	↩	Status

Description

Please refer to the **OCIRefClear** command in the OCI documentation provided by Oracle.

OCIRefFromHex

OCIRefFromHex (env ; err ; svc ; hex ; ref) -> Function result

Parameter	Type		Description
env	Longint	⇒	OCI environment handle
		⇐	
err	Longint	⇒	Error handle
		⇐	
svc	Longint	⇒	Service context handle
hex	String	⇒	Hexadecimal string to convert
ref	Longint	⇒	Ref into which hexadecimal string is converted
		⇐	
Function result	Longint	↩	Status

Description

Please refer to the **OCIRefFromHex** command in the OCI documentation provided by Oracle.

OCIRefHexSize

OCIRefHexSize (env ; ref) -> Function result

Parameter	Type		Description
env	Longint	⇒ ⇐	OCI environment handle
ref	Longint	⇒	Ref whose size is returned
Function result	Longint	⇐	Status

Description

Please refer to the **OCIRefHexSize** command in the OCI documentation provided by Oracle.

OCIRefIsEqual

OCIRefIsEqual (env ; x ; y) -> Function result

Parameter	Type		Description
env	Longint	➡ ⬅	OCI environment handle
x	Longint	➡	Ref to compare
y	Longint	➡	Ref to compare
Function result	Longint	↩	Status

Description

Please refer to the **OCIRefIsEqual** command in the OCI documentation provided by Oracle.

OCIRefIsNull

OCIRefIsNull (env ; ref) -> Function result

Parameter	Type		Description
env	Longint	⇒ ⇐	OCI environment handle
ref	Longint	⇒	Ref to test
Function result	Longint	↩	Status

Description

Please refer to the **OCIRefIsNull** command in the OCI documentation provided by Oracle.

OCIRefToHex

OCIRefToHex (env ; err ; ref ; hex) -> Function result

Parameter	Type		Description
env	Longint	⇒ ⇐	OCI environment handle
err	Longint	⇒ ⇐	Error handle
ref	Longint	⇒	Ref to be converted into a hexadecimal string
hex	String	⇐	Resulting hexadecimal string
Function result	Longint	⇐	Status

Description

Please refer to the **OCIRefToHex** command in the OCI documentation provided by Oracle.

OCITableDelete

OCITableDelete (env ; err ; index ; tbl) -> Function result

Parameter	Type		Description
env	Longint	⇒ ⇐	OCI environment handle
err	Longint	⇒ ⇐	Error handle
index	Longint	⇒	Index of element to be deleted
tbl	Longint	⇒	Table whose element is deleted
Function result	Longint	⇐	Status

Description

Please refer to the **OCITableDelete** command in the OCI documentation provided by Oracle.

OCITableExists

OCITableExists (env ; err ; tbl ; index ; exists) -> Function result

Parameter	Type		Description
env	Longint	⇨ ⇦	OCI environment handle
err	Longint	⇨ ⇦	Error handle
tbl	Longint	⇨	Table where index is checked
index	Longint	⇨	Index of element checked for existence
exists	Longint	⇦	True if element exists; otherwise, False
Function result	Longint	⇩	Status

Description

Please refer to the **OCITableExists** command in the OCI documentation provided by Oracle.

OCITableFirst

OCITableFirst (env ; err ; tbl ; index) -> Function result

Parameter	Type		Description
env	Longint		OCI environment handle
err	Longint		Error handle
tbl	Longint		Table to scan
index	Longint		Index of first existing element
Function result	Longint		Status

Description

Please refer to the **OCITableFirst** command in the OCI documentation provided by Oracle.

OCITableLast

OCITableLast (env ; err ; tbl ; index) -> Function result

Parameter	Type		Description
env	Longint	⇒ ⇐	OCI environment handle
err	Longint	⇒ ⇐	Error handle
tbl	Longint	⇒	Table to scan
index	Longint	⇐	Index of last existing element
Function result	Longint	↩	Status

Description

Please refer to the **OCITableLast** command in the OCI documentation provided by Oracle.

OCITableNext

OCITableNext (env ; err ; index ; tbl ; next_index ; exists) -> Function result

Parameter	Type		Description
env	Longint	⇒	OCI environment handle
		⇐	
err	Longint	⇒	Error handle
		⇐	
index	Longint	⇒	Index of start point for scan
tbl	Longint	⇒	Table to scan
next_index	Longint	⇐	Index of next existing element
exists	Longint	⇐	False if no next index available; otherwise, True
Function result	Longint	⇐	Status

Description

Please refer to the **OCITableNext** command in the OCI documentation provided by Oracle.

OCITablePrev

OCITablePrev (env ; err ; index ; tbl ; prev_index ; exists) -> Function result

Parameter	Type		Description
env	Longint	⇒	OCI environment handle
		⇐	
err	Longint	⇒	Error handle
		⇐	
index	Longint	⇒	Index of start point for scan
tbl	Longint	⇒	Table to scan
prev_index	Longint	⇐	Index of previous existing element
exists	Longint	⇐	False if no previous index available; otherwise, True
Function result	Longint	⇒	Status

Description

Please refer to the **OCITablePrev** command in the OCI documentation provided by Oracle.

OCITableSize

OCITableSize (env ; err ; tbl ; size) -> Function result

Parameter	Type		Description
env	Longint	⇒ ⇐	OCI environment handle
err	Longint	⇒ ⇐	Error handle
tbl	Longint	⇒	Table whose size is returned
size	Longint	⇐	Current number of elements in table
Function result	Longint	↩	Status

Description

Please refer to the **OCITableSize** command in the OCI documentation provided by Oracle.

Transactions

- OCITransCommit
- OCITransDetach
- OCITransForget
- OCITransPrepare
- OCITransRollback
- OCITransStart

OCITransCommit

OCITransCommit (svchp ; errhp ; flags) -> Function result

Parameter	Type		Description
svchp	Longint	⇒	Service context handle
errhp	Longint	⇒	Error handle
flags	Longint	⇒	Used for one-phase commit optimization in global transactions
Function result	Longint	⇐	Status

Description

Please refer to the **OCITransCommit** command in the OCI documentation provided by Oracle.

OCITransDetach

OCITransDetach (svchp ; errhp ; flags) -> Function result

Parameter	Type		Description
svchp	Longint	⇒	Service context handle
errhp	Longint	⇒	Error handle
flags	Longint	⇒	Pass OCI_DEFAULT
Function result	Longint	⇐	Status

Description

Please refer to the **OCITransDetach** command in the OCI documentation provided by Oracle.

OCITransForget

OCITransForget (svchp ; errhp ; flags) -> Function result

Parameter	Type		Description
svchp	Longint	⇨	Service context handle
errhp	Longint	⇨	Error handle
flags	Longint	⇨	Pass OCI_DEFAULT
Function result	Longint	⇦	Status

Description

Please refer to the **OCITransForget** command in the OCI documentation provided by Oracle.

OCITransPrepare

OCITransPrepare (svchp ; errhp ; flags) -> Function result

Parameter	Type		Description
svchp	Longint	⇨	Service context handle
errhp	Longint	⇨	Error handle
flags	Longint	⇨	Pass OCI_DEFAULT
Function result	Longint	⇦	Status

Description

Please refer to the **OCITransPrepare** command in the OCI documentation provided by Oracle.

OCITransRollback

OCITransRollback (svchp ; errhp ; flags) -> Function result

Parameter	Type		Description
svchp	Longint	⇒	Service context handle
errhp	Longint	⇒	Error handle
flags	Longint	⇒	Pass OCI_DEFAULT
Function result	Longint	⇐	Status

Description

Please refer to the **OCITransRollback** command in the OCI documentation provided by Oracle.

OCITransStart

OCITransStart (svchp ; errhp ; timeout ; flags) -> Function result

Parameter	Type		Description
svchp	Longint	⇒ ⇐	Service context handle
errhp	Longint	⇒ ⇐	Error handle
timeout	Longint	⇒	Time to wait for transaction to become available (seconds)
flags	Longint	⇒	Specifies info about transaction (new, read-only, serializable, etc.)
Function result	Longint	⇐	Status

Description

Please refer to the **OCITransStart** command in the OCI documentation provided by Oracle.

Types

- OCICacheFlush
- OCICacheFree
- OCICacheRefresh
- OCICacheUnmark
- OCICacheUnpin

OCICacheFlush

OCICacheFlush (env ; err ; svc ; ref) -> Function result

Parameter	Type		Description
env	Longint	⇒	OCI environment handle
		⇐	
err	Longint	⇒	Error handle
		⇐	
svc	Longint	⇒	Service context handle
ref	Longint	⇐	Points to object causing the error (if any)
Function result	Longint	⇐	Status

Description

Please refer to the **OCICacheFlush** command in the OCI documentation provided by Oracle.

OCICacheFree

OCICacheFree (env ; err ; svc) -> Function result

Parameter	Type		Description
env	Longint	⇒	OCI environment handle
		⇐	
err	Longint	⇒	Error handle
		⇐	
svc	Longint	⇒	Service context handle
Function result	Longint	⇐	Status

Description

Please refer to the **OCICacheFree** command in the OCI documentation provided by Oracle.

OCICacheRefresh

OCICacheRefresh (env ; err ; svc ; option) -> Function result

Parameter	Type		Description
env	Longint	⇒	OCI environment handle
		⇐	
err	Longint	⇒	Error handle
		⇐	
svc	Longint	⇒	Service context handle
option	Longint	⇒	If specified, all objects loaded within transaction are refreshed
Function result	Longint	⇐	Status

Description

Please refer to the **OCICacheRefresh** command in the OCI documentation provided by Oracle.

OCICacheUnmark

OCICacheUnmark (env ; err ; svc) -> Function result

Parameter	Type		Description
env	Longint	⇒ ⇐	OCI environment handle
err	Longint	⇒ ⇐	Error handle
svc	Longint	⇒	Service context handle
Function result	Longint	↩	Status

Description

Please refer to the **OCICacheUnmark** command in the OCI documentation provided by Oracle.

OCICacheUnpin

OCICacheUnpin (env ; err ; svc) -> Function result

Parameter	Type		Description
env	Longint	⇒	OCI environment handle
		⇐	
err	Longint	⇒	Error handle
		⇐	
svc	Longint	⇒	Service context handle
Function result	Longint	⇒	Status

Description

Please refer to the **OCICacheUnpin** command in the OCI documentation provided by Oracle.

Mapping 4D data types

The following table provides mapping between 4D data types and OCI constants in one place for easy reference:

4D Data Type	OCI Constant
C_REAL	SQLT_FLT
C_BOOLEAN	SQLT_INT
C_LONGINT	SQLT_INT, SQLT_ODT
C_TEXT	SQLT_STR, SQLT_LNG, SQLT_LBI
C_BLOB	SQLT_LNG, SQLT_LBI, SQLT_BLOB*
C_PICTURE	SQLT_LNG, SQLT_LBI, SQLT_BLOB*
C_TIME	SQLT_ODT
C_DATE	SQLT_ODT
	*When using an *OCILobLocator*