

# Oracle Transparent Gateways

Using Transparent Gateways with Oracle9i Application Server

Release 1.0.2.1

February 2001

Part No. A88729-01

Oracle offers two solutions for integrating data from non-Oracle systems with data stored in an Oracle database. They are Generic Connectivity and Oracle Transparent Gateways. These solutions provide the ability to transparently access data stored in non-Oracle systems creating the appearance that all of the data resides within a local Oracle database server. Generic Connectivity uses the ODBC or OLE DB standards for accessing the non-Oracle systems. Oracle Transparent Gateways are agents designed specifically for a particular non-Oracle system. Both solutions extend Oracle9i Application Server functionality to non-Oracle systems.

---

---

**Note:** The phrase *non-Oracle system* denotes both non-Oracle databases that are accessed using SQL, and systems that are accessed using procedures.

---

---

This document discusses the following topics:

- [Architecture in Oracle9i Application Server](#)
- [Installing Oracle Transparent Gateways](#)
- [Configuring Oracle9i](#)
- [Configuring Oracle Transparent Gateways](#)

**ORACLE**<sup>®</sup>

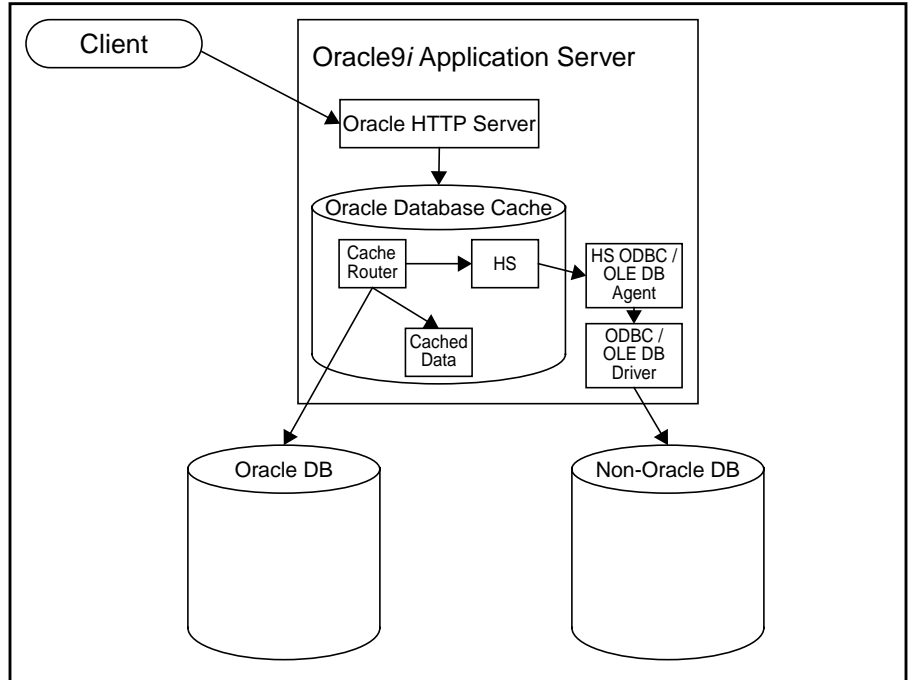
Copyright © 2001, Oracle Corporation.  
All Rights Reserved.

Oracle is a registered trademark, and Oracle Transparent Gateways, Oracle Heterogeneous Services, Oracle9i Application Server are trademarks or registered trademarks of Oracle Corporation. Other names may be trademarks of their respective owners.

## Architecture in Oracle9i Application Server

The following figures illustrate the path of a client request to either the Oracle or non-Oracle system. [Figure 1](#) shows the path using Generic Connectivity and [Figure 2](#) shows the path using Transparent Gateways.

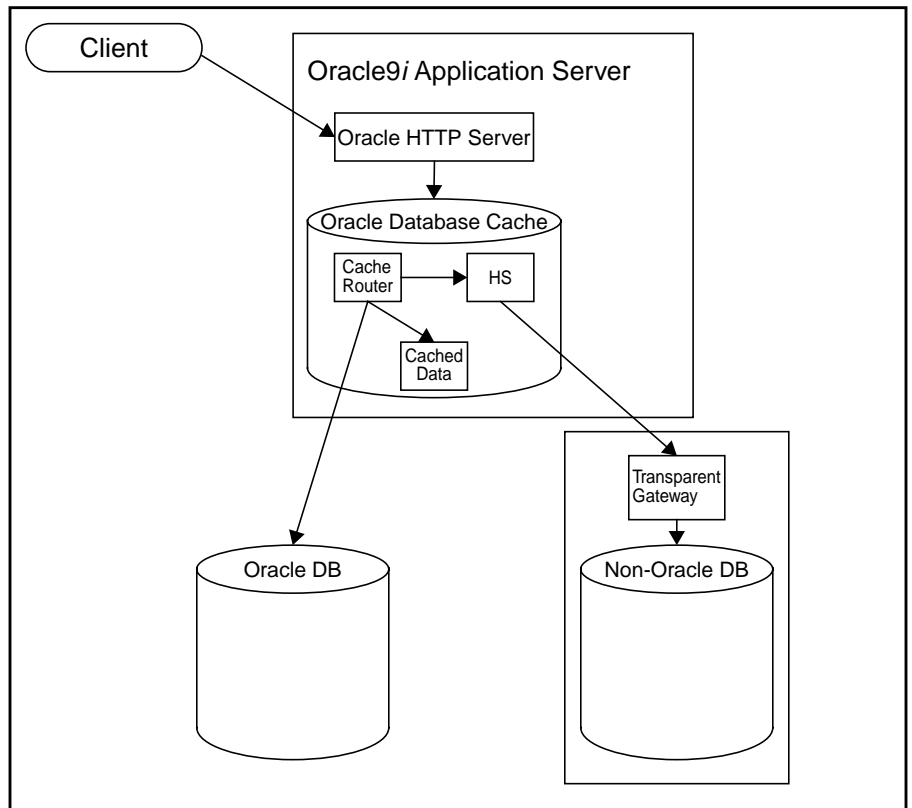
**Figure 1 Generic Connectivity Architecture**



This figure shows a client making a request for data to Oracle HTTP Server. The server forwards the data request to the Cache Router running in Oracle Database Cache. The cache router determines if the request is for data from an Oracle database or non-Oracle system. If the data is in an Oracle database, the router determines if the data is in the cache or in the back end database and forwards the request accordingly.

If the request is for data in a non-Oracle system, Cache Router forwards the request to Heterogeneous Services — the generic technology for accessing non-Oracle systems. In the Generic Connectivity case, Heterogeneous Services forwards the request to the HS ODBC or OLE DB agent which handles all communication with either the ODBC or OLE DB driver. The driver transforms the request into a form that the non-Oracle database can process.

**Figure 2 Transparent Gateway Architecture**



The Transparent Gateway scenario follows the same path as that for Generic Connectivity. The only difference is Heterogeneous Services forwards the request to a transparent gateway running on the non-Oracle system. The gateway routes the request to the non-Oracle database.

**See Also:**

- *Oracle Heterogeneous Services* in the Oracle9i Application Server Documentation Library
- Oracle Transparent Gateway for your non-Oracle database on Oracle9i Application Server Disc 1

## Installing Oracle Transparent Gateways

For installation instructions refer to the Oracle Transparent Gateway documentation for your non-Oracle database on Oracle9i Application Server Disc 1.

When installing a Transparent Gateway, you must:

- install each gateway its own Oracle home directory
- install the gateway on the same machine as the non-Oracle database

## Configuring Oracle9i

This section describes configuring a Oracle Server instance and Oracle Database Cache for use with Transparent Gateways and Generic Connectivity.

By creating the same database links and synonyms in both Oracle Server and Oracle Database Cache, you can deploy the same applications to either tier without any changes.

---

---

**Note:** You cannot cache data from a non-Oracle system in Oracle Database Cache.

---

---

1. On the back end Oracle database, create database links to the non-Oracle system.

Heterogeneous Services makes a non-Oracle system appear as a remote Oracle database server. To access or manipulate tables or to execute procedures in the non-Oracle system, create a database link that specifies the connect descriptor for the non-Oracle database. Use the following syntax to create a link to the non-Oracle system:

```
CREATE DATABASE LINK link_name
CONNECT TO user IDENTIFIED BY password
USING 'non_oracle_system';
```

**See Also:** *Oracle Heterogeneous Services* in the Oracle9i Application Server Documentation Library

2. On the back end Oracle database, create synonyms for schema objects.

Synonyms are useful because they hide the identity of the underlying object, including its location in a distributed environment. Use the following syntax to create a synonym for an object in the non-Oracle system:

```
CREATE [PUBLIC] synonym_name FOR [schema.] object_name@link_name;
```

**See Also:** *Oracle Distributed Database Systems* in the Oracle Database Documentation Library

3. For each Oracle9i Application Server instance, repeat steps 1 and 2 in the database cache. It is very important that the database cache and Oracle Server database are consistent with each other.

### **Configuring Oracle Transparent Gateways**

For configuration instructions refer to the Oracle Transparent Gateway documentation for your non-Oracle database on Oracle9i Application Server Disc 1.

