

5

Fonctionnement d'Oracle 10g sous Windows

Dans ce chapitre :

- une présentation rapide du fonctionnement d'Oracle : les fichiers, les programmes, la mémoire ;
- la notion de base de données et d'instance ;
- l'architecture d'Oracle sous Windows ;
- l'utilisation de la base de registres ;
- démarrer et arrêter Oracle sous Windows ;
- l'utilitaire ORADIM ;
- l'assistant d'administration Oracle pour Windows.

Ce chapitre constitue une introduction indispensable pour comprendre les bases du fonctionnement d'Oracle sous Windows, en vous donnant une vision d'ensemble des interactions entre Oracle et Windows. Il précède d'autres chapitres décrivant plus précisément à la fois le fonctionnement d'Oracle et certains liens avec Windows, mais cette présentation générale demeure essentielle.

La première partie de ce chapitre présente les grands composants d'une base de données Oracle et introduit des concepts et les termes techniques utilisés dans ce livre.

La seconde partie traite des spécificités d'Oracle sous Windows. Nous abordons dans le détail les liens qui unissent le fonctionnement de Windows et Oracle. Des exemples pratiques et des cas concrets sont exposés.

L'utilisation d'ORADIM et de l'assistant d'administration Oracle pour Windows sont détaillées.

Oracle 10g : une seule version

Lorsqu'il fonctionne sous Windows, Oracle 10g possède des caractéristiques identiques à celles d'Oracle sous d'autres systèmes d'exploitation comme Unix ou Linux. Cependant, Oracle adapte son logiciel pour exploiter au mieux les possibilités offertes par chaque système d'exploitation.

L'interface entre Oracle et Windows a été modifiée pour s'adapter et tirer des avantages de l'architecture et des évolutions de Windows, y compris sur les environnements 64 bits. Entre les versions d'Oracle sous Windows et sous Unix, les différences d'architecture ne concernent essentiellement que des aspects très proches des « couches basses » des systèmes d'exploitation. Les ingénieurs d'Oracle ont adapté la base de données sous Windows pour permettre une intégration parfaite autorisant un fonctionnement performant et stable.

Des outils d'administration complémentaires propres à Windows sont venus enrichir la palette des outils Oracle que l'on retrouve sur tous les systèmes d'exploitation. Ainsi, un administrateur Windows ne sera pas dépaysé : il contrôlera tous les aspects liés à Windows en utilisant des outils familiers. De même, un administrateur Unix retrouvera sous Windows des outils connus.

Les composants d'Oracle 10g sous Windows

Avant de les détailler dans les chapitres suivants, nous vous présentons succinctement les différents composants d'une base Oracle 10g, à savoir :

- les composants « fichiers » ;
- les composants « programmes » ;
- les composants « mémoire ».

Les composants « fichiers »

Nous ne présentons que les fichiers qui composent votre ou vos bases de données. Les autres fichiers, mis en place lors de l'installation Oracle, sont décrits au chapitre 13, *L'installation d'Oracle 10g sous Windows*.

Toute base de données se réduit en définitive à des fichiers qui contiennent les données et à d'autres qui interviennent dans le fonctionnement. Chaque base de données possède des fichiers spécifiques, qu'il faut pouvoir identifier et dont il s'agit de connaître les objectifs. Les quatre types de fichiers indispensables au fonctionnement d'une base Oracle sont :

- les fichiers de données ;
- les fichiers redo-log ;
- les fichiers de contrôle ;
- les fichiers d'administration.

Les fichiers de données contiennent les tables, index, procédures, fonctions, etc., ainsi que le dictionnaire de données (conçu lors de la création d'une base de données). Le dictionnaire, géré à 100 % par Oracle, est le cœur de la gestion interne de la base. Les fichiers de données constituent plus de 90 % du volume global d'une base Oracle. Comme se sont les fichiers les plus volumineux, leur taille, leur emplacement et leur paramétrage sont de la responsabilité de l'administrateur de la base de données.

Les fichiers redo-log mémorisent l'historique de tous les ordres modifiant des données ou la structure de la base de données. Ils sont au minimum trois, chaînés circulairement ; ce qui signifie que le premier se remplit, puis le deuxième, le troisième, puis le premier, etc. Ils assurent une mission de sécurité. Ils garantissent que dès que vous validez la modification d'une donnée, elle sera réalisée quoi qu'il arrive (arrêt brutal, panne électrique, etc.). Leur taille est fixée lors de la création de la base de données (modifiable ultérieurement). Leur fonctionnement est géré à 100 % par Oracle.

Les fichiers de contrôle renseignent la base de données sur son état précédent lors du démarrage, de l'arrêt et lors de certaines phases de fonctionnement. Par exemple, lors de son démarrage, leur consultation permet de savoir quel type d'arrêt a eu lieu précédemment : normal, brutal, etc. Ils contiennent par ailleurs les chemins d'accès et les noms de tous les fichiers (base de données et redo-log) qui composent la base de données. Pour démarrer une base de données, il suffit que le logiciel Oracle sache où se trouvent les fichiers de contrôle. Il y puise un ensemble de renseignements qui lui serviront à réaliser l'opération. Leur taille, quelques centaines de kilo-octets maximum et leur fonctionnement sont gérés à 100 % par Oracle.

Les fichiers d'administration servent à l'administration courante d'Oracle. Ce sont par exemple les fichiers d'initialisation utilisés lors du démarrage de la base, les fichiers servant à paramétrer certains outils et enfin les différents fichiers de trace et d'alerte générés par Oracle.

Tous ces fichiers sont utilisés par les composants « programmes ».

Les composants « programmes »

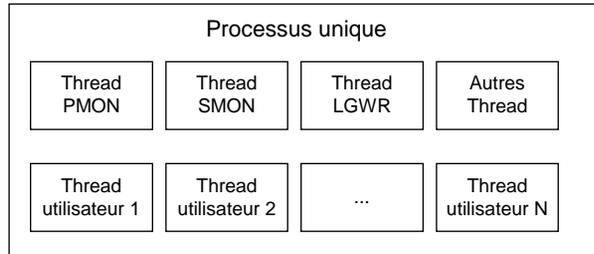
Ces composants « programmes » sont les exécutable qui assurent le fonctionnement de la base de données. Ils réalisent toutes les actions sur les fichiers (base, contrôle, redo-log). Sous Unix, on parlera des « processus » ; sous Windows, d'« exécutable », de « processus » et de « threads ».

L'architecture du système d'exploitation Windows permet à l'exécutable d'Oracle d'être multitâche et multithread. Ce sont ces différents threads qui interagissent avec les composants mémoires, les fichiers et les programmes utilisateurs.

Figure 5-1

Architecture d'Oracle
sous Windows

Architecture d'Oracle sous Windows :
des Threads communicant au sein d'un Processus



Les composants « mémoires »

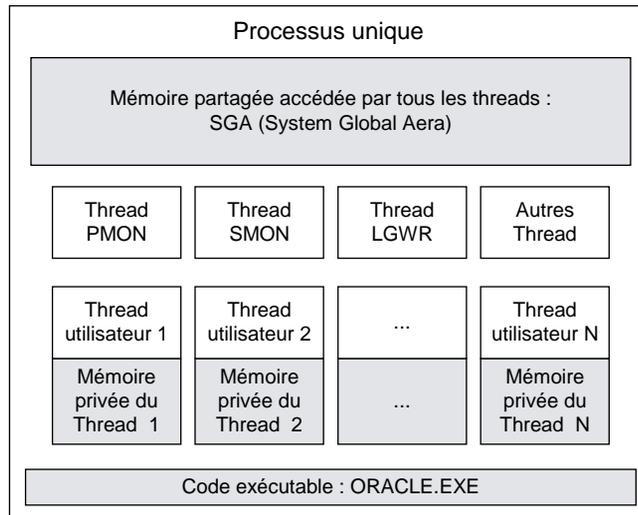
Pour assurer son fonctionnement et ses performances, Oracle a besoin de mémoire pour plusieurs usages :

- une zone mémoire partagée par tous les utilisateurs Oracle ;
- une zone mémoire privée pour chaque utilisateur connecté ;
- une zone mémoire pour exécuter les programmes.

Une zone mémoire partagée (paramétrable suivant votre configuration) est primordiale pour les performances de la base de données : c'est la System Global Area (SGA). Elle est divisée en plusieurs parties qui assurent : le cache des données manipulées, des zones de tris, l'espace pour analyser et réutiliser les ordres SQL déjà exécutés, l'exécution des programmes PL/SQL et Java au sein de la base, etc.

Figure 5-2

Mémoire utilisée
par une instance Oracle



Les autres zones mémoires sont automatiquement allouées pour le fonctionnement de la base de données (la mémoire prise par les exécutables) et par les connexions utilisateur (chaque connexion nécessite un minimum de mémoire).

Base de données et instance

Une base de données Oracle se compose donc de :

- fichiers : base de données, redo-log, contrôle, administration ;
- exécutables : pour faire fonctionner la base ;
- mémoire.

On parlera de *base de données* pour qualifier l'ensemble des fichiers qui la composent.

Une *instance* regroupe la base de données (les fichiers), la zone mémoire allouée (SGA) et les exécutables assurant le fonctionnement de la base. On peut simplifier en disant qu'une instance est une base de données « en action ».

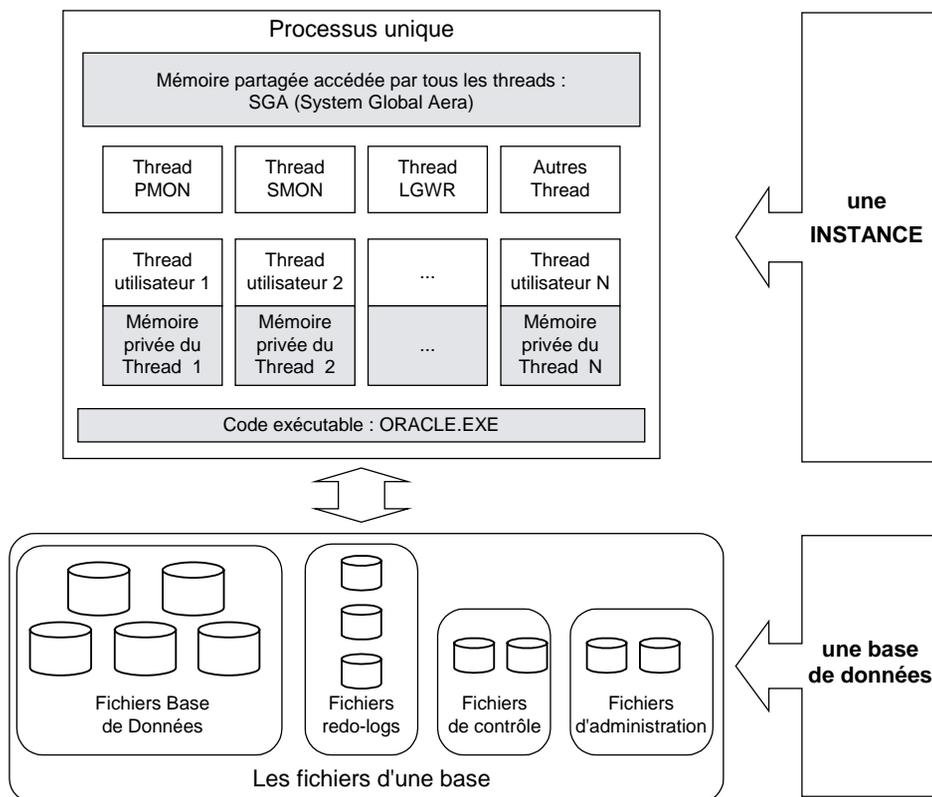


Figure 5-3

Base de données et instance

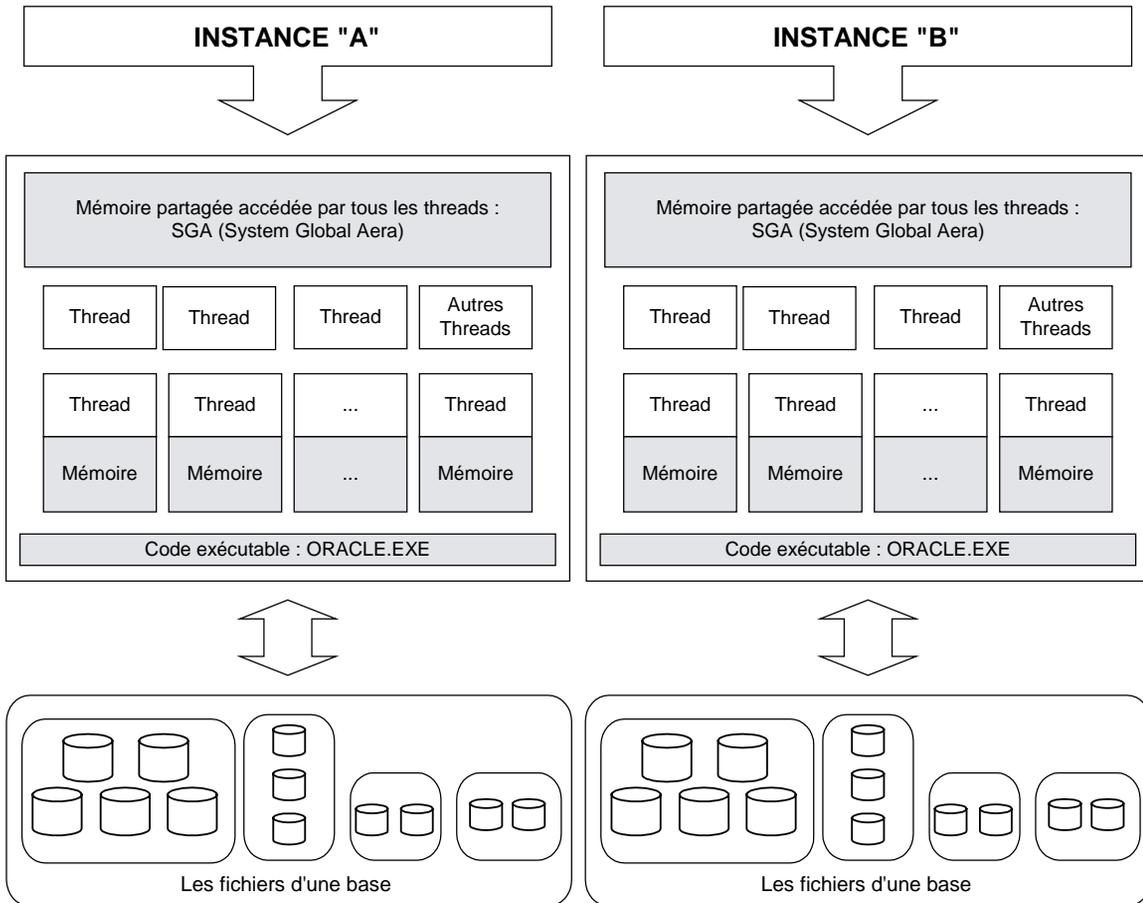


Figure 5-4

Fonctionnement simultané de deux instances sous Windows

Nous aborderons les avantages et inconvénients d'avoir une ou plusieurs bases de données sur un seul serveur Windows au chapitre 14, *Création d'une base Oracle 10g*.

Architecture interne d'Oracle 10g sous Windows

Le serveur Oracle 10g fonctionne sous Windows comme un exécutable unique ayant de multiples threads. Chaque thread a ses propres tâches à réaliser. Certains threads sont dus aux connexions des utilisateurs, d'autres réalisent le fonctionnement de la base de données en tâche de fond.

Les threads internes au fonctionnement d'Oracle 10g sont les équivalents des processus sous Unix. On retrouve des threads équivalents à PMON, DBWR, ARCH, etc., c'est-à-dire

à toutes les grandes fonctions internes du moteur Oracle 10g. Le nombre de threads lancés dépend à la fois des fonctions internes du moteur Oracle 10g (certaines, comme la réplication, sont optionnelles) et du nombre d'utilisateurs connectés.

Cet exécutable Oracle 10g fonctionne sous la forme d'un *service*, car c'est le mode de fonctionnement sur les systèmes d'exploitation Microsoft. Les services associés à une base de données sont abordés ultérieurement dans ce chapitre.

Réservée exclusivement aux environnements Microsoft, l'utilisation des services est l'une des caractéristiques principales d'Oracle sous Windows. La notion de service n'existe pas sous les systèmes Unix et Linux.

L'exécutable oracle.exe

Chaque instance d'Oracle 10g consiste donc en un programme exécutable unique ayant de multiples threads. Pour que chaque nouvelle instance soit identifiée par Windows, son nom figure dans la base de registre sous la forme d'un *service*. Le nom du service sera toujours OracleServiceSID, où SID représente l'identifiant de l'instance Oracle. Ce service est associé à un exécutable oracle.exe.

Le service OracleServiceSID se compose initialement de quelques threads, qui ont pour tâche de gérer l'ensemble des threads ultérieurs. C'est sous le contrôle de ce service initial que s'effectuent des opérations telles que le démarrage ou l'arrêt de la base de données.

S'il y a plusieurs instances Oracle sur un serveur, il y aura autant de services OracleServiceSID et d'exécutables oracle.exe lancés que d'instances.

Fonctionnement de l'exécutable oracle.exe sous Windows

Le démarrage d'oracle.exe sous la forme d'un service permet d'être indépendant d'une session utilisateur Windows. Le service et son instance associée peuvent être lancés automatiquement lors du démarrage de Windows.

Il est indispensable que le service soit lancé **avant** de pouvoir démarrer l'instance Oracle. Ces deux opérations peuvent être synchronisées.

Par défaut, les services sont lancés sous le compte Windows Système. Système n'est pas un utilisateur capable de créer une session Windows, il sert à assurer la gestion des services.

La création d'un service associé à une instance est assurée par l'utilitaire ORADIM, présenté ultérieurement dans ce chapitre.

Les différents threads d'oracle.exe

Les administrateurs d'Oracle sur d'autres systèmes d'exploitation connaissent bien les différents processus d'Oracle (PMON, SMON, etc.). Sous Windows, ce sont des threads du processus oracle.exe. Les premiers threads sont utilisés pour la gestion des threads suivants et ne sont pas documentés par Oracle. Les autres threads sont PMON, DBWR,

LGWR, SMON, RECO. Après ceux-ci, il en existe d'autres qui sont optionnels (ARCH, CKPT, etc.). Suivent ensuite les threads créés par les connexions des utilisateurs. Nous commentons plus largement les différents threads au chapitre 7, *Le processus Oracle 10g*.

Thread	Abréviation	Description
Visibles base fermée		Threads gestionnaires des autres threads d'Oracle 10g
Process Monitor	PMON	Il administre les différents threads et processus d'Oracle 10g. Il est chargé de la libération des ressources occupées si une session est anormalement tuée (exemple : un client distant éteint son PC sans se déconnecter d'Oracle).
Database Writer	DBWR	Il est chargé de l'écriture dans la base de données. Il écrit les blocs de données modifiés en SGA (mémoire centrale) dans les fichiers base.
Log Writer	LGWR	Il écrit séquentiellement depuis la SGA dans le fichier redo-log courant. LGWR écrit un enregistrement de petite taille dans le fichier redo-log courant. Cela garantit la possibilité de restaurer la base de données suite à un incident (exemple : coupure électrique).
System Monitor	SMON	C'est le grand « nettoyeur » : il est responsable de la validation d'une base lors de son démarrage. Il vérifie si la base a été arrêtée correctement. Si ce n'est pas le cas, il récupère dans les fichiers redo-log les enregistrements validés (par un COMMIT ou un ROLLBACK) qui n'ont pas encore été écrits en base par Oracle afin de les y écrire. Les autres enregistrements (non validés) sont annulés.
Recover	RECO	Résout les pannes lorsque l'option distribuée est utilisée.
Archivage	ARCH	C'est un thread optionnel. Par défaut, il n'est pas actif. Si le mode ARCHIVELOG est activé, sa mission consiste à archiver (copier) chaque redo-log plein dans un endroit particulier (répertoire ou bande). Les fichiers log peuvent être utilisés pour reconstruire une base de données lorsqu'elle a été perdue.
Checkpoint Dédié	CKPT	Prévient le DBWR de réaliser une opération de <i>checkpoint</i> (littéralement : point de contrôle) sur tous les composants de la base de données. S'il n'est pas présent, ce rôle est tenu par le LGWR.
Autres threads		Se sont les threads dédiés aux utilisateurs connectés à la base de données ou à certaines options d'Oracle.

La liste complète des autres threads est présentée dans ce chapitre. Elle est accessible par l'ordre SQL :

```
select name, description from v$bgprocess.
```

Les « sessions utilisateurs » d'Oracle

Chaque connexion à la base Oracle 10g crée un thread dédié dans l'exécutable oracle.exe qui gère votre instance. Toutes les actions de cet utilisateur connecté seront gérées par ce thread. C'est lui qui dialoguera avec la SGA (pour exécuter les ordres SQL et accéder aux données), et il s'allouera automatiquement de la mémoire (par exemple pour assurer le rôle de « buffer » et vous retourner vos données à l'issue d'un ordre SQL).

Comment identifier facilement la session d'un utilisateur ?

Avec le Process Viewer (présenté au chapitre 4, *Environnement Windows : l'indispensable*), il n'est pas possible d'identifier précisément quel est l'utilisateur Oracle connecté à un thread particulier. C'est grâce aux outils fournis par Oracle que vous pouvez identifier quel thread est associé à une connexion à la base Oracle 10g. Il existe plusieurs outils qui utilisent tout le SQL pour interroger le dictionnaire de données Oracle 10g. Le résultat peut être retourné sous forme textuelle ou graphique, suivant l'outil utilisé :

- ordre SQL lancé « manuellement » depuis SQL*Plus ;
- liste des processus graphiques depuis l'*Oracle Administration Assistant for Windows* ;
- liste des processus depuis *Oracle Enterprise Manager*.

Un exemple d'ordre SQL particulièrement complet est donné plus loin dans ce chapitre.

Oracle et la base de registre

Pour conserver toutes les informations concernant l'installation, le paramétrage d'Oracle et d'outils sur Windows, Oracle les conserve dans la base de registres (ou *registry*) de Windows. Vous pouvez consulter son contenu avec l'utilitaire Windows Regedit ou plus simplement avec l'*Oracle Administration Assistant for Windows* fourni par Oracle. Ce dernier outil n'offre l'accès qu'à un nombre limité de clés de registres utilisées par Oracle.

La façon dont Oracle structure ses clés de registre est détaillée au chapitre 9, *Les fichiers d'une base Oracle 10g*.

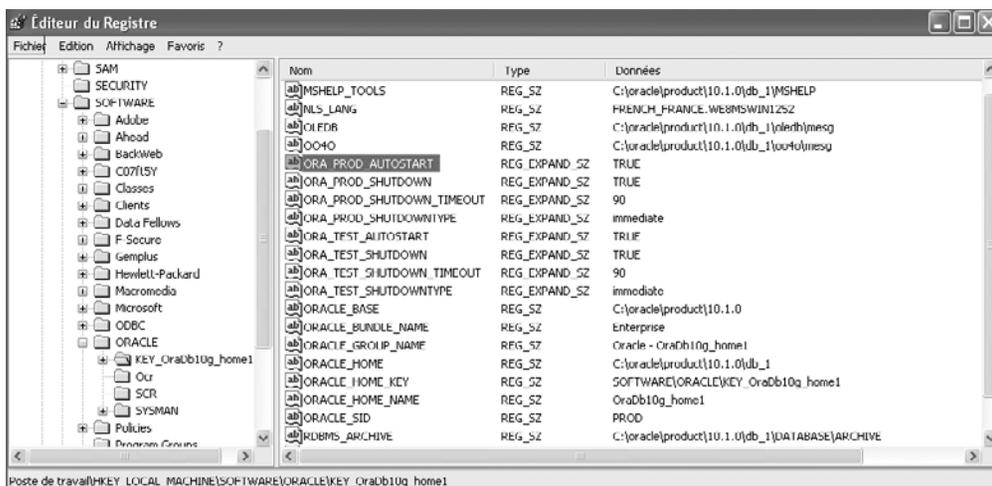


Figure 5-5

La base de registre

L'utilisation du registre par Oracle

Oracle utilise pleinement la base de registre et de nombreux paramètres y figurent. Il est important de comprendre comment Oracle a structuré ses clés de registre. Hormis les services, la majorité des clés de registre utilisées par Oracle sont regroupées sous HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE. Le paramétrage des services est conservé en HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet001\Services.

Ces points sont détaillés au chapitre 9, *Les fichiers d'une base Oracle 10g*.

Plusieurs versions d'Oracle peuvent être installées simultanément sur un serveur. Chacune d'elle sera identifiée par un nom d'Oracle Home différent. On parlera alors du HOMEID associé à l'Oracle Home. Cette sous-clé du registre Oracle contiendra tous les paramètres utilisés par cette version d'Oracle.

Pour chaque version Oracle installée, le numéro ID sera incrémenté en partant de zéro. Par exemple, si les versions Oracle 10.1 et 10.2 ont été installées, leurs paramètres respectifs seront placés sous les clés :

```
HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\KEY_OraDb10g_home1  
HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\KEY_OraDb10g_home2
```

Les différentes versions d'Oracle peuvent fonctionner simultanément. Certains programmes ont un nom identique quelle que soit la version, par exemple SQLPLUS, EXP ou IMP. Pour travailler avec l'une ou l'autre de ces versions, Oracle propose un outil graphique, l'*Oracle Administration Assistant for Windows*, qui positionne toutes les variables nécessaires pour aiguiller vers la version à utiliser.

Plusieurs bases de données peuvent fonctionner simultanément. Chacune d'elles possède des caractéristiques qui peuvent être différentes de celles d'une autre base, comme la langue utilisée ou les propriétés de démarrage et d'arrêt. Là aussi, des paramètres propres à chaque base de données sont enregistrés dans le registre.

Pour l'instant, les valeurs essentielles de la base de registre que nous allons utiliser sont :

ORACLE_HOME

Cette variable placée en HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\KEY_OraDb10g_home1 indique le répertoire racine où sont installés les logiciels Oracle. Ce point est abordé au chapitre 9, *Les fichiers d'une base Oracle 10g*.

ORACLE_SID

ORACLE_SID permet de définir une instance par défaut. C'est à elle que vous vous connecterez par défaut, par exemple avec SQL*Plus. Il est possible de changer cette valeur ou de se connecter à une autre instance :

- en modifiant ORACLE_SID dans le registre avec regedit ;
- depuis l'*Oracle Administration Assistant for Windows* ;

- en utilisant un « alias » Oracle Net pour se connecter en indiquant précisément la base cible (voir le chapitre 15, *Oracle .NET, le middleware Oracle*).

Le service OracleServiceSID

Chaque instance Oracle (identifiée par un *SID*) nécessite la présence d'un service OracleServiceSID, situé en HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet001\Services. Il peut être paramétré (par ordre de préférence décroissant) :

- depuis l'*Oracle Administration Assistant for Windows* ;
- depuis le Gestionnaire de Services Windows ;
- directement, en agissant sur le registre avec Regedit.

Démarrer et arrêter Oracle sous Windows

Dans ce chapitre, nous abordons les *services* Windows, l'instance Oracle, la base de registre. Leur fonctionnement est très fortement imbriqué. Pour vous présenter de manière pratique ces éléments théoriques, nous allons utiliser plusieurs scénarios :

- le démarrage du service Windows ;
- le démarrage de l'instance Oracle ;
- l'arrêt de l'instance Oracle ;
- l'arrêt du service Windows.

Quels utilitaires pour quelle tâche ?

Les utilitaires sont nombreux et parfois l'utilisateur néophyte ne sait lequel choisir par rapport à un besoin bien précis.

Avant d'entrer dans le détail de chacun de ces utilitaires, voyons à quoi ils servent, quel est celui que nous vous recommandons en premier lieu, et ceux qui peuvent vous être utiles en second lieu.

Tâche à effectuer	Choix prioritaire	Choix complémentaire
Créer une base de données	Oracle Database Configuration Assistant (DBCA)	ORADIM et SQL*Plus
Démarrer une base	Gestionnaire de services de Windows	ORADIM SQL*Plus Oracle Enterprise Manager
Arrêter une base	Gestionnaire de services de Windows	ORADIM SQL*Plus Oracle Enterprise Manager
Paramétrer le démarrage et l'arrêt d'une base	sOracle Administration Assistant for Windows	Regedit

Tâche à effectuer	Choix prioritaire	Choix complémentaire
Changer le mot de passe d'une base	ORAPWD	ORADIM
Gérer les utilisateurs d'une base	Oracle Enterprise Manager Console	SQL*Plus Windows Oracle Administration Assistant for Windows
Exporter les données d'une base	Data Pump Export (EXPDP)	Export (EXP)
Importer les données d'une base	Data Pump Import (IMPDP)	Import (IMP)
Charger des données dans une base	Oracle Enterprise Manager Load Wizzard	SQL*Loader (SQLLDR)
Supprimer une base	Oracle Database Configuration Assistant (DBCA)	ORADIM et SQL*Plus
Supprimer le service d'une base	Oracle Database Configuration Assistant (DBCA)	ORADIM
Sauvegarder une base	Oracle Enterprise Manager Backup Wizzard	Recovery Manager (RMAN) OCOPY
Restaurer une base	Oracle Enterprise Manager Recovery Wizzard	Recovery Manager (RMAN) OCOPY

Consultez l'index pour identifier les chapitres où sont présentés chaque utilitaire.

Démarrer et arrêter Oracle sous Windows

Le fonctionnement d'Oracle sous Windows impose que le service Windows soit démarré avant de pouvoir démarrer/arrêter l'instance Oracle. L'ordre imposé est donc :

1. Démarrer le service OracleServiceSID.
2. Démarrer l'instance Oracle SID.
3. Arrêter l'instance Oracle SID.
4. Arrêter le service OracleServiceSID.

Certaines étapes peuvent s'effectuer simultanément. Par exemple, le démarrage du service OracleServiceSID peut entraîner celui de l'instance Oracle. De même, l'arrêt du service OracleServiceSID peut provoquer un arrêt « propre » ou « brutal » de l'instance Oracle. Étudions chacune de ces étapes, spécifiques à l'environnement Windows.

Pour cela, nous allons utiliser plusieurs outils présentés au chapitre 4, *Environnement Windows : l'indispensable*, ainsi que d'autres logiciels fournis par Oracle :

- le Gestionnaire de services ;
- le Gestionnaire de tâches ;

- le Process Viewer ;
- l'assistant *Oracle Administration Assistant for Windows* accessible depuis le menu Démarrer > Programmes > Oracle - OraDb10g_home1 > Configuration and Migration Tools.

Démarrer le service Windows OracleServiceSID

Le premier écran montre comment démarrer le service d'une base Oracle 10g à partir du Gestionnaire des services Windows. Nous utilisons pour cela de la base TEST, créée au chapitre 13, *Installation d'Oracle 10g sous Windows*. Vous noterez que ce service démarre automatiquement lorsque le système Windows est lancé.

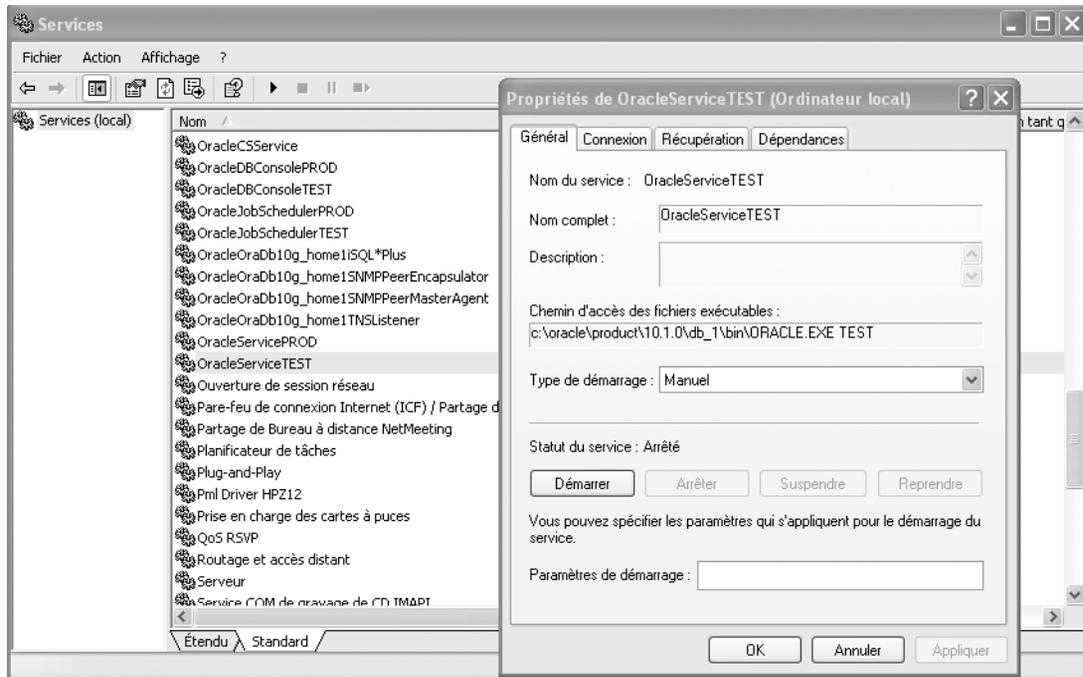


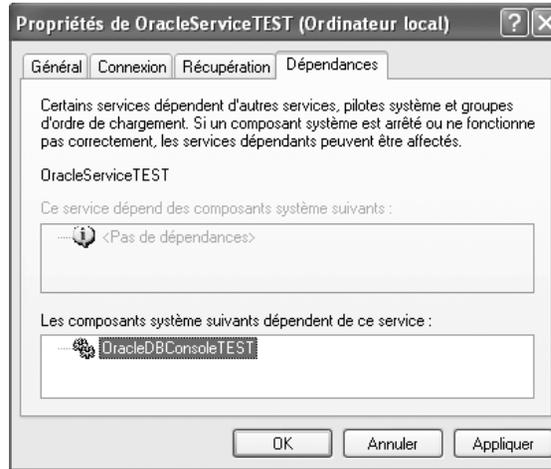
Figure 5-6

Le service OracleServiceTEST

Comme indiqué précédemment, le service lance l'exécutable oracle.exe en lui passant le nom de l'instance comme paramètre.

Par défaut, il s'exécute sous le compte local SYSTEM de Windows (à ne pas confondre avec le compte Oracle SYSTEM). Il est possible de le lancer sous un compte utilisateur particulier. Comme cela complique l'administration, nous vous conseillons de laisser l'option par défaut.

Figure 5-7
*Propriétés du service
 OracleServiceTEST*



Si vous avez installé Oracle Enterprise Manager Database Control, l'onglet *Dépendances* indique que le service OracleDBConsoleTEST démarre juste après le service OracleServiceTEST. Il permet l'accès à OEM.

Vérifions à l'aide du Gestionnaire de tâches de Windows que l'exécutable oracle.exe, dédié à la gestion de cette instance TEST, a été lancé. Il y aura autant de fois cet exécutable oracle.exe qu'il y aura d'instances Oracle lancées sur le serveur.

Figure 5-8
*Compte d'exécution
 du service*

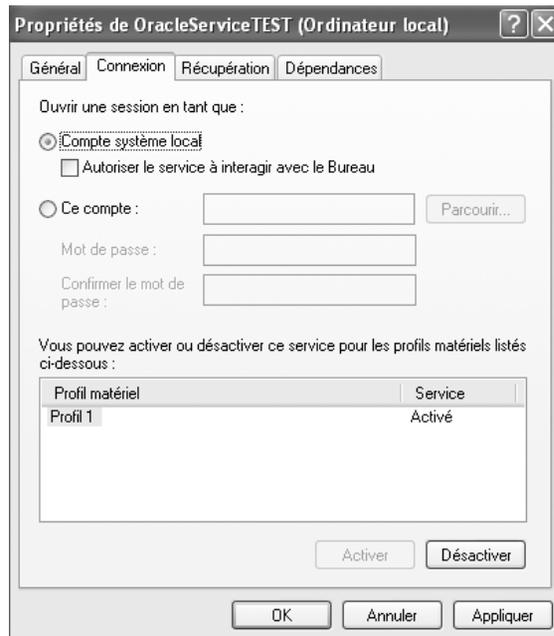
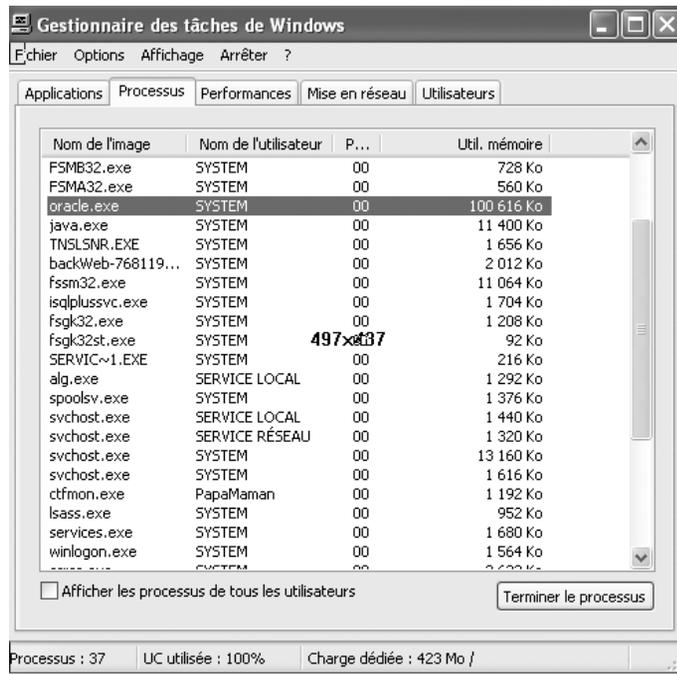
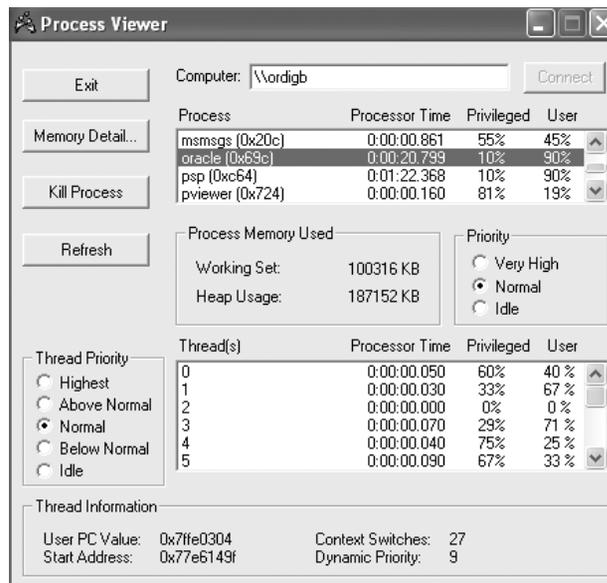


Figure 5-9
Le processus oracle.exe



Avec le Process Viewer (outil fourni avec le *Windows 2000 Support Tools*), nous pouvons entrer dans l'exécutable oracle.exe lancé et visualiser les threads actifs de cette instance.

Figure 5-10
Visualisation des threads
avec le Process Viewer



Dans cet exemple, de nombreux threads sont lancés : tout indique que l'instance a été lancée en même temps que le service. Vérifions cette hypothèse avec l'*Oracle Administration Assistant pour Windows* (faites un « clic droit » sur la base de données cible, TEST dans notre exemple).

Vous constaterez que seuls les threads liés à Oracle et ceux liés aux connexions utilisateurs sont affichés. Les autres threads, nécessaires au fonctionnement interne du processus et qui figurent dans le Process Viewer, ne sont pas montrés.

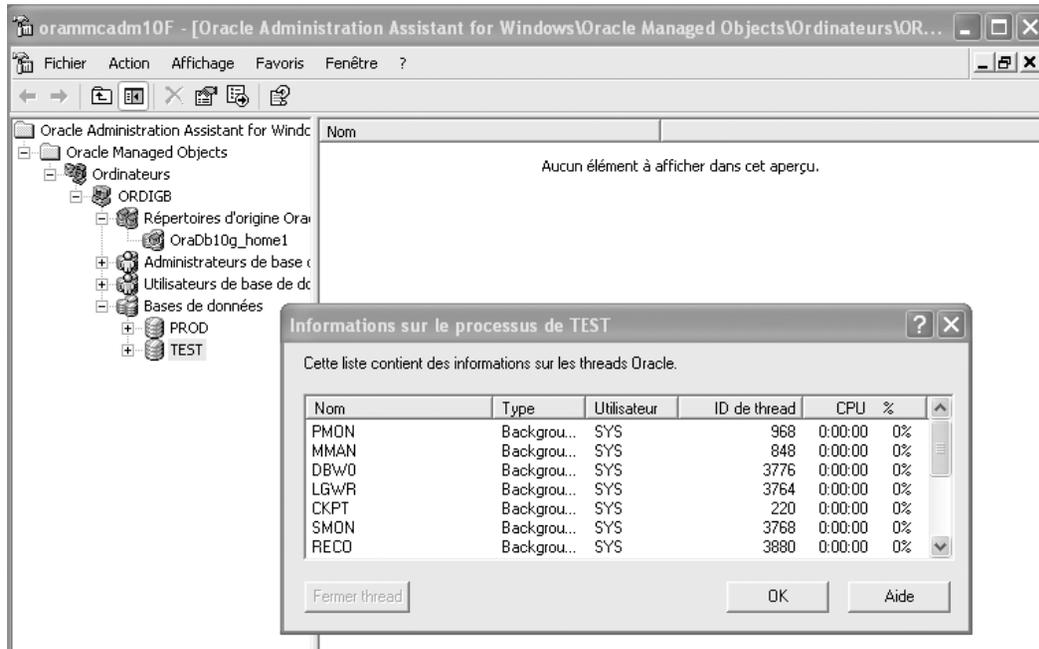


Figure 5-11

Visualisation des threads de l'instance

Nous sommes donc dans le cas idéal, où le service puis l'instance Oracle ont été démarrés. Nous verrons plus loin où se situe ce paramétrage.

Voyons quelles sont les possibilités offertes pour le démarrage du service OracleServiceSID :

Démarrer le service OracleServiceSID

On peut démarrer ce service depuis le Gestionnaire de services ou manuellement, à l'aide de la commande net start.

```
# Depuis une fenêtre Invite de Commandes
# lancement du service de l'instance TEST

c:\> net start OracleServiceTEST
```

Lors de cette étape, rien n'indique si le démarrage de l'instance est synchronisé avec celui sur service ou non.

Démarrer le service OracleServiceSID avec ORADIM

L'utilitaire Oracle ORADIM (présenté dans ce chapitre) gère la création et la suppression des services OracleServiceSID. Il permet aussi de démarrer ou d'arrêter le service et l'instance, en fournissant les paramètres requis.

```
# Depuis une fenêtre Invite de Commandes
# lancement de l'utilitaire ORADIM
# (démarrage du service uniquement)
```

```
c:\> ORADIM -startup -sid TEST -usrpwd oracle -starttype srvc -spfile
```

L'utilitaire Oracle ORADIM servant à la gestion des services Windows, il n'existe que sur les systèmes Microsoft.

Démarrer l'instance Oracle

Lorsque le service OracleServiceSID est lancé mais que l'instance associée n'est pas démarrée, la base de données est alors inactive. Dans ce cas, les threads de l'exécutable oracle.exe en attente du démarrage de l'instance sont visibles à l'aide du Process Viewer.

Figure 5-12

Threads d'oracle.exe avant le démarrage de l'instance

The screenshot shows the Windows Process Viewer window for the 'oracle' process. The process is running on computer 'Vordigb'. The process memory used is 32556 KB (Working Set) and 2868 KB (Heap Usage). The process priority is set to 'Normal'. The threads table shows five threads with the following details:

Thread(s)	Processor Time	Privileged	User
0	0:00:00.080	63%	37%
1	0:00:00.020	0%	100%
2	0:00:00.000	0%	0%
3	0:00:00.000	0%	0%
4	0:00:00.811	9%	91%

Thread Information:

User PC Value:	0x77fe0304	Context Switches:	181
Start Address:	0x77e6149f	Dynamic Priority:	9

Vous pouvez maintenant démarrer l'instance de plusieurs façons :

- en arrêtant puis relançant le service ;
- avec l'utilitaire ORADIM ;
- à partir de SQL*Plus ;
- depuis Oracle Enterprise Manager.

Démarrer l'instance par le service OracleServiceSID

Comme précédemment, il est possible de démarrer le service depuis le Gestionnaire de services ou manuellement, à l'aide de la commande net start.

```
# Depuis une fenêtre Invite de Commandes
# lancement du service
c:\> net start OracleServiceTEST
```

Comment synchroniser le démarrage du service avec celui de l'instance ?

L'utilitaire *Oracle Administration Assistant for Windows* permet de paramétrer les options de démarrage et d'arrêt d'une instance Oracle par rapport au démarrage et à l'arrêt du service Windows. Sélectionnez la base cible (TEST dans cet exemple) et faites un « clic droit » pour accéder aux options de configuration du démarrage et de l'arrêt du service.

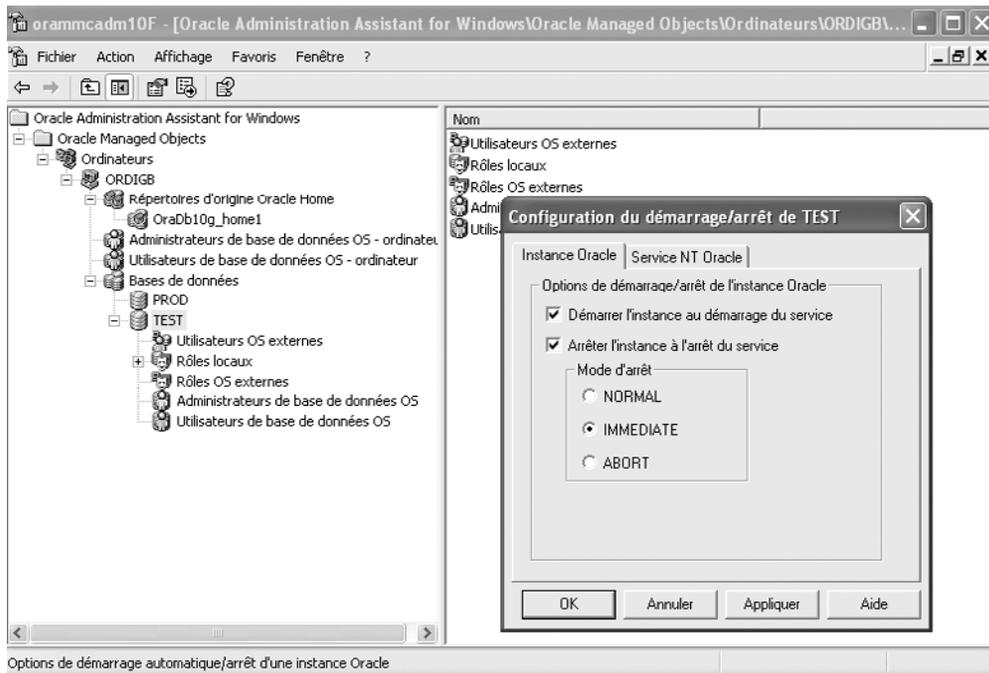


Figure 5-13

Paramétrage du service

L'onglet Instance Oracle synchronise le démarrage et l'arrêt de l'instance Oracle avec celui du service. L'onglet Service NT Oracle reprend les paramètres présents dans le Gestionnaire de services Windows.

Où sont conservés ces paramètres ?

Ces paramètres sont conservés dans la base de registre. Ils sont situés en HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\KEY_OraDb10g_home1.

Comme plusieurs instances peuvent fonctionner simultanément, le nom de l'instance figure dans chacune des clés.

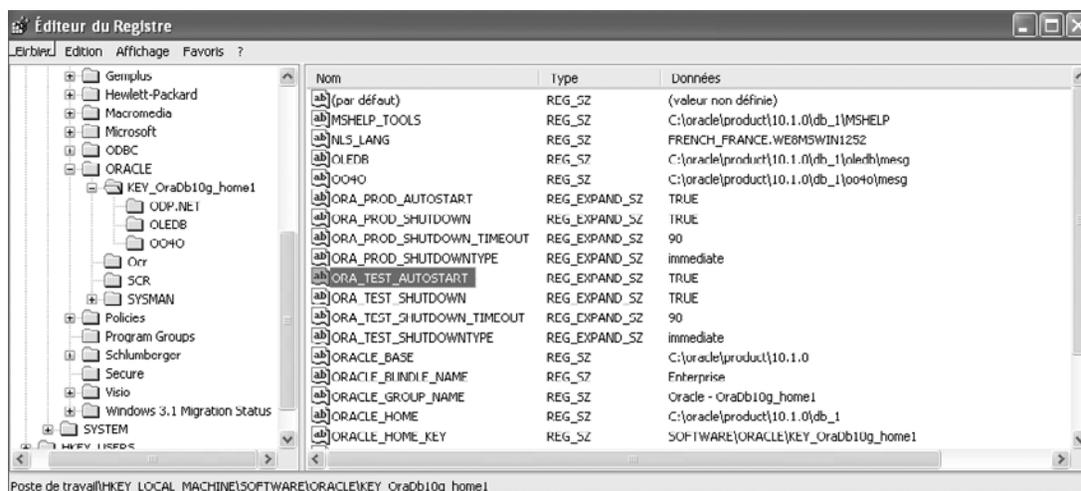


Figure 5-14

Les clés du registre concernées

Démarrer l'instance avec l'utilitaire ORADIM

L'utilitaire Oracle ORADIM (présenté dans ce chapitre) gère la création et la suppression des services OracleService.SID. Il permet également de démarrer ou d'arrêter l'instance, en fournissant tous les paramètres requis.

```
# Depuis une fenêtre Invite de Commandes
# lancement de l'utilitaire ORADIM (démarrage de l'instance)
```

```
c:\> ORADIM -startup -sid TEST -starttype inst -spfile
```

Démarrer l'instance avec SQL*Plus

Depuis plusieurs versions d'Oracle, l'ancien outil Server Manager a fusionné avec SQL*Plus. C'est maintenant SQL*Plus qu'il faut utiliser pour démarrer et arrêter les bases Oracle : c'est l'outil standard privilégié pour effectuer toutes les opérations sur une base Oracle.

Pour démarrer l'instance Oracle sous Windows, le service OracleServiceSID doit être préalablement lancé.

```
# Depuis une fenêtre Invite de Commandes
# Définissez le SID de l'instance
C:\> set ORACLE_SID=TEST

# lancez SQL*Plus
C:\> sqlplus /nolog
SQL> connect / as sysdba
Connecté.
SQL> startup
Instance ORACLE lancée.
Total System Global Area 135338868 bytes
Fixed Size 453492 bytes
Variable Size 109051904 bytes
Database Buffers 25165824 bytes
Redo Buffers 667648 bytes
Base de données montée.
Base de données ouverte.
SQL> exit
```

Démarrer l'instance avec Oracle Enterprise Manager (OEM)

Oracle Enterprise Manager permet de démarrer et d'arrêter les bases Oracle. Les bases peuvent être locales ou distantes et fonctionner sous Windows ou sous Unix et Linux. Le chapitre 25, *Oracle Enterprise Manager*, lui est consacré.

The screenshot shows the Oracle Enterprise Manager 10g Database Control interface. The main title is "ORACLE Enterprise Manager 10g Database Control". The current database is "Base de données : TEST.gilles.fr". The status is "Démarré" (Started) at "27 avr. 2005 22:11:18". The "Arrêter" (Stop) button is circled in red. The interface includes several sections:

- Général:** Displays instance details such as Nom de l'instance (test), Version (10.1.0.2.0), and Read Only (Non).
- UC de l'hôte:** A bar chart showing CPU usage for 'Autres', 'test', and 'test'.
- Sessions actives:** A pie chart showing 0.01 sessions active and 29.01 SQL response time.
- Haute disponibilité:** Shows recovery time (22 seconds) and archiving status (Désactivé).
- Utilisation de l'espace:** Shows database size (1 Go) and logical space issues (0).
- Récapitulatif des diagnostics:** Shows performance search results (0) and rule violations (10).

Figure 5-15

Arrêt de l'instance avec OEM

Arrêter l'instance Oracle

L'arrêt de l'instance s'effectue à l'aide des mêmes outils que leur démarrage :

- en arrêtant le service ;
- avec l'utilitaire ORADIM ;
- à partir de SQL*Plus ;
- depuis Oracle Enterprise Manager

Arrêt de l'instance par le service OracleServiceSID

Il est possible d'arrêter le service depuis le Gestionnaire de services ou manuellement, à l'aide de la commande net stop.

```
# Depuis une fenêtre Invite de Commandes
# arrêt du service de l'instance TEST

c:\> net stop OracleServiceTEST
```

Par défaut, l'arrêt du service impose l'arrêt « propre » de l'instance Oracle. Si ce choix n'a pas été fait, l'arrêt du service entraînera la fermeture brutale de l'instance Oracle. Les conséquences sont abordées au chapitre 10, *Démarrer et arrêter une base Oracle 10g*.

Arrêter l'instance avec SQL*Plus

SQL*Plus permet d'arrêter l'instance Oracle sans arrêter le service OracleServiceSID.

```
# Depuis une fenêtre Invite de Commandes
# Définissez le SID de l'instance
C:\> set ORACLE_SID=TEST

# lancez SQL*Plus
C:\> sqlplus /nolog
SQL> connect / as sysdba
Connecté.
SQL> shutdown immediate
Base de données fermée.
Base de données démontée.
Instance ORACLE arrêtée.
SQL>
SQL> exit
```

Arrêter l'instance avec Oracle Enterprise Manager

La console d'administration d'Oracle Enterprise Manager permet de démarrer et d'arrêter toutes vos bases Oracle.

Avant l'arrêt de la base, comme elle fonctionne donc, le service OracleServiceSID est forcément actif à cet instant.

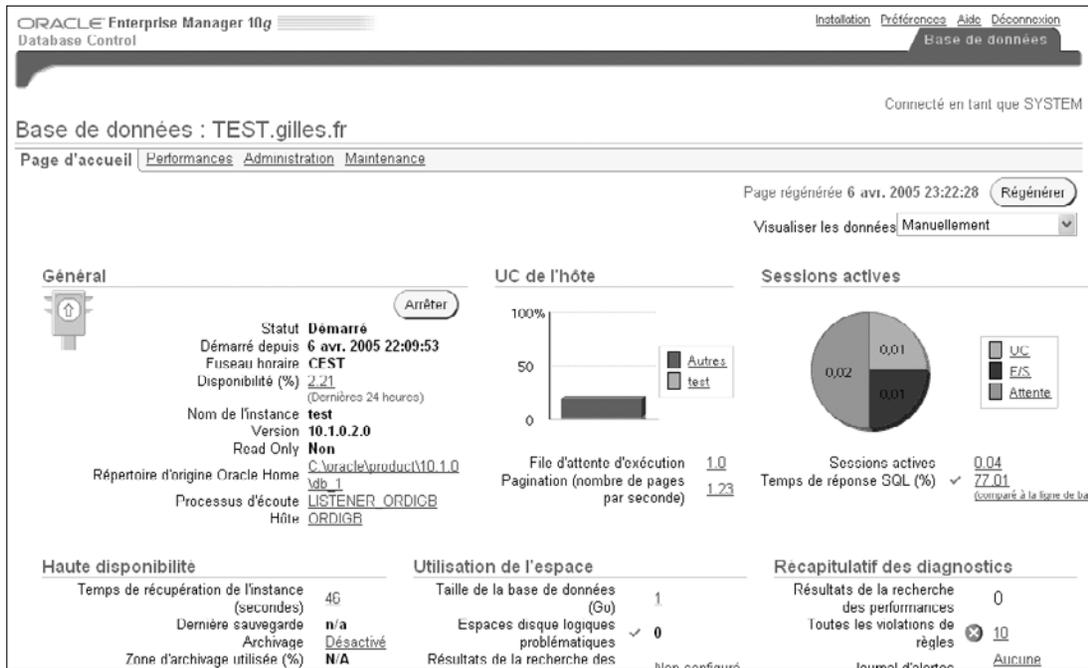


Figure 5-16

Arrêt de l'instance par OEM

Lorsque seul le service fonctionne, quels sont les threads restants ?

Après l'arrêt de l'instance Oracle, le service OracleServiceSID peut rester actif. Dans ce cas, seuls les threads nécessaires à son fonctionnement et à une demande de démarrage de l'instance restent actifs.

Arrêter le service Windows

De même que pour son démarrage, l'arrêt du service OracleServiceSID peut être effectué de plusieurs manières :

Arrêter le service OracleServiceSID par le Gestionnaire de services

On peut arrêter le service depuis le Gestionnaire de services ou manuellement, à l'aide de la commande net stop.

```
# Depuis une fenêtre Invite de Commandes
# arrêt du service de l'instance TEST
```

```
c:\> net stop OracleServiceTEST
```

Arrêter le service OracleServiceSID avec ORADIM

L'utilitaire Oracle ORADIM (présenté dans ce chapitre) gère la création et la suppression des services OracleServiceSID. Il permet également de démarrer ou d'arrêter le service et l'instance, en fournissant les paramètres requis.

```
# Depuis une fenêtre Invite de Commandes
# lancement de l'utilitaire ORADIM (arrêt du service uniquement)

c:\> ORADIM -shutdown -sid TEST -shuttype srvc
```

Que se passe-t-il si l'instance fonctionne lors de l'arrêt du service ?

Reportez-vous au paragraphe précédent pour paramétrer les options de démarrage et d'arrêt du service avec celui de l'instance Oracle.

Après l'arrêt du service, l'exécutable oracle.exe associé au service s'arrête.

Comment associer les threads aux process Oracle ?

En ce qui concerne le système d'exploitation, nous avons vu qu'il est simple de visualiser les threads du processus oracle.exe. De même, en ce qui concerne Oracle, on accède facilement à la liste des process Oracle. Mais comment relier un thread à un processus Oracle ? Comment connaître le nom de l'utilisateur connecté sous Windows ? Comment savoir quel est le programme qui a donné naissance au thread utilisateur ?

C'est ce que propose l'ordre SQL suivant, qui relie tous ces éléments. Nous vous le donnons comme exemple de la puissance des renseignements qu'il est possible de puiser dans le dictionnaire de données d'Oracle.

```
SQL> select p.spid "NoThread",
        b.name "Process",
        s.username "UserName",
        s.osuser "OS User",
        s.status "STATUS",
        s.sid "Session",
        s.serial# "Serial",
        s.program "OS Program"
        from v$process p, v$bgprocess b, v$session s
        where s.paddr = p.addr and b.paddr(+) = p.addr
```

No Thread	Proce ss	UserName	OS User	STATUS	Session	OS Program
532	PMON		SYSTEM	ACTIVE	170	ORACLE.EXE (PMON)
812	MMAN		SYSTEM	ACTIVE	169	ORACLE.EXE (MMAN)
1016	DBWO		SYSTEM	ACTIVE	168	ORACLE.EXE (DBWO)
1824	LGWR		SYSTEM	ACTIVE	167	ORACLE.EXE (LGWR)
1928	CKPT		SYSTEM	ACTIVE	166	ORACLE.EXE (CKPT)
2008	SMON		SYSTEM	ACTIVE	165	ORACLE.EXE (SMON)
248	RECO		SYSTEM	ACTIVE	164	ORACLE.EXE (RECO)

```

300   CJQO           SYSTEM    ACTIVE    163   ORACLE.EXE (CJQO)
2004  QMNC           SYSTEM    ACTIVE    159   ORACLE.EXE (QMNC)
152   MMON           SYSTEM    ACTIVE    158   ORACLE.EXE (MMON)
244   MMNL           SYSTEM    ACTIVE    156   ORACLE.EXE (MMNL)
2916          SYSTEM    ORDIGB\GB ACTIVE    147   sqlplusw.exe
1356          SCOTT      ORDIGB\GB ACTIVE    155   sqlplusw.exe
                                     Administra
                                     teur

```

Dans cet exemple, on visualise les processus Oracle et les numéros de threads Windows associés. On constate que les utilisateurs SCOTT et SYSTEM sont tous deux connectés depuis le compte administrateur du serveur ORASERV et qu'ils utilisent la version graphique de SQL*Plus pour accéder à la base. L'identification sans risque d'erreur du numéro de thread utilisateur permet ainsi de « tuer » une session utilisateur.

L'ordre SQL suivant permet de visualiser l'ensemble des threads qu'une base Oracle sous Windows peut créer. La liste est impressionnante ! Lors d'un fonctionnement « normal », seule une partie de ces threads sera démarrée.

```
SQL> select name, description from v$bgprocess
```

```

NAME  DESCRIPTION
-----
PMON  process cleanup
DIAG  diagnosability process
FMON  File Mapping Monitor Process
LMON  global enqueue service monitor
LMD0  global enqueue service daemon 0
LMS0  global cache service process 0
LMS1  global cache service process 1
LMS2  global cache service process 2
LMS3  global cache service process 3
LMS4  global cache service process 4
LMS5  global cache service process 5
LMS6  global cache service process 6
LMS7  global cache service process 7
LMS8  global cache service process 8
LMS9  global cache service process 9
LMSa  global cache service process 10
LMSb  global cache service process 11
LMSc  global cache service process 12
LMSd  global cache service process 13
LMSe  global cache service process 14
LMSf  global cache service process 15
LMSg  global cache service process 16
LMSh  global cache service process 17
LMSi  global cache service process 18
LMSj  global cache service process 19
MMAN  Memory Manager
DBW0  db writer process 0
DBW1  db writer process 1

```

```
DBW2 db writer process 2
DBW3 db writer process 3
DBW4 db writer process 4
DBW5 db writer process 5
DBW6 db writer process 6
DBW7 db writer process 7
DBW8 db writer process 8
DBW9 db writer process 9
DBWa db writer process 10 (a)
DBWb db writer process 11 (b)
DBWc db writer process 12 (c)
DBWd db writer process 13 (d)
DBWe db writer process 14 (e)
DBWf db writer process 15 (f)
DBWg db writer process 16 (g)
DBWh db writer process 17 (h)
DBWi db writer process 18 (i)
DBWj db writer process 19 (j)
ARC0 Archival Process 0
ARC1 Archival Process 1
ARC2 Archival Process 2
ARC3 Archival Process 3
ARC4 Archival Process 4
ARC5 Archival Process 5
ARC6 Archival Process 6
ARC7 Archival Process 7
ARC8 Archival Process 8
ARC9 Archival Process 9
LNS0 Network Server 0
LNS1 Network Server 1
LNS2 Network Server 2
LNS3 Network Server 3
LNS4 Network Server 4
LNS5 Network Server 5
LNS6 Network Server 6
LNS7 Network Server 7
LNS8 Network Server 8
LNS9 Network Server 9
MRP0 Managed Standby Recovery
LGWR Redo etc.
LCK0 Lock Process 0
CKPT checkpoint
LSP0 Logical Standby
LSP1 Dictionary build process for Logical Standby
LSP2 Set Guard Standby Information for Logical Standby
CTWR Change Tracking Writer
RVWR Recovery Writer
SMON System Monitor Process
RECO distributed recovery
CJQ0 Job Queue Coordinator
EMNO Event Monitor Process 0
QMNC AQ Coordinator
```

```
DMON  DG Broker Monitor Process
RSM0  Data Guard Broker Resource Guard Process 0
RSM1  Data Guard Broker Resource Guard Process 1
NSV0  Data Guard Broker NetSlave Process 0
NSV1  Data Guard Broker NetSlave Process 1
NSV2  Data Guard Broker NetSlave Process 2
NSV3  Data Guard Broker NetSlave Process 3
NSV4  Data Guard Broker NetSlave Process 4
NSV5  Data Guard Broker NetSlave Process 5
NSV6  Data Guard Broker NetSlave Process 6
NSV7  Data Guard Broker NetSlave Process 7
NSV8  Data Guard Broker NetSlave Process 8
NSV9  Data Guard Broker NetSlave Process 9
INSV  Data Guard Broker INstance SlaVe Process
RBAL  ASM Rebalance master
ARB0  ASM Rebalance 0
ARB1  ASM Rebalance 1
ARB2  ASM Rebalance 2
ARB3  ASM Rebalance 3
ARB4  ASM Rebalance 4
ARB5  ASM Rebalance 5
ARB6  ASM Rebalance 6
ARB7  ASM Rebalance 7
ARB8  ASM Rebalance 8
ARB9  ASM Rebalance 9
ARBA  ASM Rebalance 10
ASMB  ASM Background
MMON  Manageability Monitor Process
MMNL  Manageability Monitor Process 2
```

Rassurez-vous, vous n'avez pas à connaître l'ensemble de ces threads. Seuls les principaux sont à identifier. Ils sont présentés dans les chapitres suivants.

L'utilitaire ORADIM

Le paragraphe précédent a montré le rôle important que joue l'utilitaire ORADIM. N'existant que dans l'environnement Windows, il permet :

- la création et la suppression des services OracleServiceSID dans la base de registre ;
- le démarrage et l'arrêt des services OracleServiceSID ;
- le démarrage et l'arrêt des instances Oracle SID.

L'utilitaire Oracle Database Assistant permet de piloter graphiquement toutes les options d'ORADIM. Cet utilitaire est présenté au chapitre 14, *Création d'une base Oracle 10g*.

L'aide en ligne d'ORADIM est accessible par la commande `ORADIM -help`. Dans la syntaxe qui suit, les paramètres entre crochets [] sont optionnels, les paramètres séparés par « | » indiquent une condition « ou ». Par exemple « A|B » signifie « A ou B ».

ORADIM permet aussi de créer le service Windows associé à une instance Automatic Storage Management (ASM). Les options nécessaires à ASM ne sont pas abordées dans ce chapitre.

Création d'une instance

ORADIM -NEW -SID *vosre_sid* | -SRVC *nom_service* [-SYSPWD *mot_de_passe_SYS*] [-STARTMODE AUTO | MANUAL] [-SRVCSTART system | demand] [-PFILE *fichier_initialisation* | -SPFILE] [-SHUTMODE normal | immediate | abort] [-TIMEOUT *secondes*]

- NEW indique que l'on va créer un nouveau service ;
- *vosre_sid* est l'identifiant de l'instance ou *SID* (utilisez quatre caractères majuscules) ;
- *nom_service* est le nom du service Windows à créer. Respectez une syntaxe du type OracleService*SID* (comme pour OracleServiceTEST) ;
- *mot_de_passe_SYS* est le mot de passe utilisé lorsqu'on se connecte sous le compte utilisateur possédant le privilège SYSDBA. Il est facultatif. Si vous ne l'utilisez pas, le groupe Windows ORA_DBA permet aux utilisateurs qui le possèdent de se connecter à Oracle sans fournir de mot de passe. Par défaut, seul les administrateurs Windows sont membres de ce groupe ;
- AUTO ou MANUAL indique le mode de démarrage du service lorsque Windows se lance ;
 - PFILE *fichier_initialisation* précise où se trouve le fichier d'initialisation INIT-SID.ORA (fournissez le nom du fichier et son chemin d'accès). Vous pouvez le remplacer par le SPFILE, qu'Oracle va chercher par défaut dans le répertoire ORACLE_HOME\database ;
 - SHUTMODE normal | immediate | abort indique le mode d'arrêt de l'instance lors de l'arrêt de Windows ;
 - TIMEOUT *secondes* détermine la durée d'attente d'un service avant son arrêt. L'arrêt de Windows est alors différé le temps que la base s'arrête « proprement ».

Cette commande crée uniquement les services, les variables dans la base de registre et le fichier mot de passe associé à la nouvelle base. La base de données n'est pas créée à cette étape.

Par exemple, pour créer le service associé à une base nommée TEST :

```
ORADIM -NEW -SID TEST -STARTMODE auto -SPFILE
```

Démarrage d'une instance

ORADIM -STARTUP -SID *vosre_sid* [-SYSPWD *mot_de_passeSYS*] [-STARTTYPE SRVC, INST] [-PFILE *fichier_initialisation* | -SPFILE]

- -STARTUP indique le démarrage ;

- *votre_sid* est l'identifiant de votre instance ou SID à démarrer ;
- *mot_de_passeSYS* est le mot de passe SYS s'il a été saisi lors de la création de l'instance ;
- SRVC démarre le service associé à l'instance (oracle.exe) sans démarrer l'instance ;
- INST démarre l'instance Oracle ;
- SRVC, INST : les deux paramètres peuvent être donnés ;
- -PFILE *fichier_initialisation* précise où se trouve le fichier d'initialisation INITSID.ORA (fournissez le nom du fichier et son chemin d'accès). Vous pouvez le remplacer par le SPFILE, qu'Oracle va chercher par défaut dans le répertoire ORACLE_HOME\database.

Les paramètres -STARTUP, -SID, -STARTTYPE sont obligatoires, les autres sont optionnels.

Par exemple, pour démarrer le service et la base associés à une base nommée TEST :

```
ORADIM -STARTUP -SID TEST -STARTTYPE SRV,INST auto -SPFILE
```

Arrêt d'une instance

```
ORADIM -SHUTDOWN -SID votre_sid ] [-SYSPWD mot_passeSYS ] [-SHUTTYPE  
SRVC | INST | SRVC,INST] -SHUTMODE [nomal, immediate, abort]
```

- -SHUTDOWN indique l'arrêt ;
- *votre_sid* est l'identifiant de votre instance ou SID ;
- *mot_passeSYS* est le mot de passe SYS s'il a été saisi lors de la création de l'instance ;
- SRVC arrête le service associé à l'instance (oracle.exe), INST arrête uniquement l'instance ;
- SRVC, INST : les deux paramètres peuvent être saisis ;
- SHUTMODE indique le type d'arrêt de l'instance : Abort ou « shutdown abort », le plus violent, Immediate (toutes les connexions sont fermées puis la base arrêtée), Normal (on attend que toutes les connexions soient fermées avant d'arrêter la base). Méfiez-vous de « N » si des process en tâche de fond sont connectés à la base (c'est le cas de l'agent intelligent Oracle et des serveurs Apache), ils empêchent alors l'arrêt de votre instance. Si vous omettez ce paramètre, l'option Normal est utilisée.

L'arrêt d'une base Oracle est traité au chapitre 10, *Démarrer et arrêter une base Oracle 10g*.

Les paramètres -SUTDOWN, -SID, -SHUTTYPE sont obligatoires, les autres sont optionnels.

Par exemple, pour arrêter l'instance et le service associés à une base nommée TEST :

```
ORADIM -SHUTDOWN -SID TEST -SHUTTYPE SRV,INST -SHUTMODE IMMEDIATE
```