

Institut Supérieur des Etudes Technologiques en Communication de Tunis

Projet de fin d'études

**configuration de l'agent SNMP d'Oracle et interfaçage avec
un gestionnaire de réseau.**

Réalisé par
Garouachi Sami
Labiadh Boulbaba
TS-Télécommunication

Encadré par
Aidoudi Imed

Remerciement

C'est avec un grand plaisir que nous réservons cette page en signe de gratitude et de profonde reconnaissance à tous ceux qui nous ont aidé de près ou de loin à la réalisation de ce travail.

Nous remercions surtout Mr ‘Aidoudi Imed ‘ pour sa disponibilité, son aide et sa précieuse directive, qui nous a permis de mener à terme ce projet de fin d’études.

sommaire

I.	Cadre de projet.....	2
II.	Présentation de projet.....	3
	II.1. Problématique	3
	II.2. Travail demandé	3
III.	Société d'accueil	3
	III.1. Présentation générale	3
	III.2. Fiche d'identité	4
	III.3. Domaines d'activité	4
	III.4. Domaines Activités scientifiques.....	4
IV.	Outils de Travail	5
 CHAPITRE 2 : ADMINISTRATIONS DES RESEAUX.....		6
I.	Introduction.....	7
II.	Fonctions d'administration.....	7
	II.1. La gestion des erreurs	7
	II.2. La gestion de configuration	8
	II.3. La gestion de sécurité	9
	II.4. La gestion des performances	10
	II.5. La gestion de coût.....	10
III.	Le responsable réseau et ses besoins.....	11
	III.1. Qui administre ?.....	11
	III.2. Administrer quoi ?.....	11
IV.	Conclusion.....	12
 CHAPITRE 3 : LE PROTOCOLE SNMP.....		13
I.	Introduction	14
II.	Architecture de SNMP	14
	II.1. Principe de base.....	14
	II.2. Les acteurs de l'administration.....	15
	II.2.1 La station d'administration.....	15
	II.2.2 Les entités managées.....	16
	II.2.3 Le protocole de gestion.....	16
	II.2.4 Les informations administratives.....	18
III.	Management information base (MIB)	18
IV.	Echange de données	20
	IV.1. Les commandes SNMP	20
	IV.2. Le format des messages SNMP.....	27

V. Conclusion.....	31
CHAPITRE 4 : ARCHITECTURE DE MIB ORACLE.....	32
I. Introduction	
II. L'outil d'administration d'Oracle	
II.1 Présentation	
II.2 Les Composantes d'Entreprise Manager d'Oracle	
II.2.1 Console	
II.2.2 Agents intelligents	
II.3 Lancement d'Oracle Enterprise Manager	
III Les composants SNMP d'Oracle au niveau de l'agent	
III.1. Agent de maître	
III.2. Sous-agent	
IV Architecture de MIB SNMP d'Oracle	
IV.1. Interprétation de l'identificateur d'Oracle OID	
IV.2. Format d'une variable MIB	
V Interprétations des tables de variables MIB Oracle	
V.1 Les tables des variables MIB de base de données Oracle	
V.2 Les tables des variables MIB Oracle Listener	
V.3 Les tables des variables MIB Oracle Name	
V.4 les tables des variables de MIB RDBMS public	
V.5 Les variables MIB de mise en œuvre de service réseau	
V.6 Les variables MIB de Manager d'Entreprise	
VI Conclusion	

CHAPITRE 5 : CONFIGURATION DE L'AGENT SNMP D'ORACLE ET INTERFACAGE AVEC UN GESTIONNAIRE DE RESEAU

I. Introduction	
II.. Le Listener Oracle	
II.1. Configuration des alias	
II.2 Etude du fichier Tnsnames.ora	
II.2.1 Configuration pour un alias TCP/IP	
II.2.2 Configuration pour un alias NetBios ou SPX/ IPX	
II.3 Le fichier Sqtnet.ora	
II.4 le fichier listener.ora	
II.5 Où placer les fichiers Tnsnames, Sqtnet, Listener ?	

III Oracle SNMP Agent pour Services d'Oracle

III.1 Configuration d'Oracle SNMP Agent	
III.2 Direction de l'Agent de Maître et l'Encapsulator	
III.2.1 Départ de l'Agent de Maître	
III.2.2 Départ de l'Encapsulator	
III.3 Compréhension du Fichier MASTER.CFG	
III.4 Compréhension du Fichier ENCAPS.CFG	

IV Exemple de gestionnaire de réseau

IV.1 HP Open View

IV.2 Gestionnaire De Réseau SNMPC 5.0

IV.2.1 Console à distance.

IV.2.2 Affichages En temps réel De MIB

V. Conclusion

Conclusion générale

Bibliographie

ANNEXES

A. L'environnement réseau d'Oracle.

A.1 Qu'est-ce que le middleware Net8 ?

A .1.1 Comment dialoguer entre plusieurs serveur Oracle

A .1.2 Oracle Net8

B MIB RMON

D .ODBC, Oracle Objets pour OLE, JDBC ET SQLJ

D.1. Configuration d'un pilote ODBC

D.2. Lien ODBC entre Excel et Oracle8

D.3. Lien ODBC entre Word et Oracle8

E. Interprétation des variables MIB d'Oracle

E.1 Interprétation des Variables de la Base de Données d'Oracle MIB

E.2 Interprétation des Variables MIB de Listener d'Oracle

E.3 Interprétation des Variables MIB d'Oracle names

E.4 Interprétation des Variables MIB du RDBMS Public

E.5 Interprétation des Variables MIB de Network Services

E.6 Interprétation des Variables MIB d'Entreprise Manager

Glossaire

RFCs

CHAPITRE 1

INTRODUCTION GENERALE

Aujourd'hui, les réseaux sont de plus en plus des éléments clés de transfert d'information d'un endroit vers un autre. Aussi est-il nécessaire de constater qu'on sait attaché à fournir des solutions aux problèmes de communication plutôt que problèmes de gestion des réseaux ?

La conséquence de cette démarche est qu'actuellement on est capable de construire des infrastructures de réseaux très complexes. Paradoxalement, on s'aperçoit alors que l'on possède que peu d'information décrivant le fonctionnement de tel environnement. La mise en place d'un système d'administration paraît donc l'élément clé pour assurer une utilisation optimale de réseau.

L'activité d'administration implique la réalisation d'un ensemble de fonctions permettant une action efficace dans le temps sur le court terme (exploitation et contrôle des activités des réseaux), sur le moyen et long terme (planification et développement), cela suppose l'utilisation d'un ensemble de moyen aussi bien logiciel que matériel de transmission, de surveillance, de traitement et d'archivage.

Ce pendant la multiplicité des réseaux et l'hétérogénéité des outils d'administration ne facilitent pas la tâche d'administrateur. Il est donc nécessaire de mettre en oeuvre une structure d'administration globale qui prend en compte la disparité des systèmes à administrer. Tout cela n'est pas facile et de traduit plus souvent par des difficultés et des lourdeurs supplémentaires pour la gestion des réseaux.

I. Cadre de projet :

Le présent projet : Configuration de l'agent SNMP d'Oracle et interfaçage avec un gestionnaire de réseau, est réalisé dans le cadre de la présentation d'un mémoire de fin d'études en vue de l'obtention du diplôme de technicien supérieur en télécommunication à l'Institut Supérieur des Etudes Technologiques en Communication de Tunis l'ISET'COM pour l'année universitaire 2000/2001.

❖ Cadre de projet à l'irsit.

le projet s'intitule « Gestion automatique des messages aéronautiques ». L'objectif du système est d'assurer une banque de données des messages aéronautique. Cette base de données doit contenir des informations liées à la sécurité des vols des différents compagnies aériennes. C'est pour cette raison qu'elle doit être distribuée sur une architecture réseau et disponible continuellement, pour satisfaire les requêtes des utilisateurs qui sont les pilotes. Afin, d'assurer une

haute disponibilité de l'information, l'équipe IRSIT recourt aux outils d'administration d'oracle et aux services offertes par la base MIB SNMP d'oracle.

II. Présentation de projet :

II.2. Problématique :

Dans l'environnement quotidien des utilisateurs, des ressources matériels que logiciels sont distribués entre plusieurs réseaux et sont disponible sur des systèmes très variés. L'administrateur doit gérer son réseau pour obtenir des informations en temps réel sur le trafic ou les caractéristiques des performances.

La mission de l'administration des réseaux est d'autant plus difficile lorsque la taille du réseau est plus important et que l'environnement est hétérogène.

II.3. Travail demandé :

Dans ce cadre notre projet se situe, dont nous sommes demandés d'analyser les messages trap générer par des instances Oracle sur un réseau, en interfacant un gestionnaire de réseau. Pour cela on doit accomplir le travail suivant :

- Mise en place d'une configuration réseau de plusieurs instances Oracles.
- Configuration de l'outil d'administration d'Oracle (ORACLE ENTREPRISE MANAGER).
- Analyse et description de la base MIB d'Oracle.
- Configuration de l'agent SNMP d'Oracle.
- Interfaçage avec un gestionnaire de réseau.
- Interprétation des résultats.

III. Société d'accueil :

III.1. Présentation générale :

Ce projet est réalisé au sein de l'Institution de Recherche en Sciences Informatiques et Télécommunication IRSIT. L'IRSIT est une société de R&D en technologies de l'information. Plus précisément, l'IRSIT est spécialisée en informatique, réseaux, télécommunications, téléservices, systèmes d'information, système d'aide à la décision et technologies de la langue. Le rôle proéminent de l'IRSIT est de développer de nouvelles applications de technologies de

l'information, promouvoir les technologies d'information à l'échelle national et régionale et les adapter aux marchés des pays en voie de développement.

III.2. Fiche d'identité :

Raison Sociale :

Forme Juridique :

Création : IRSIT créée en 1986

Siège Sociale : Parc technologique des communications, route de Raoued

Km 3.5 – 2083 Ariana Tunisie

Téléphone / Fax : (216 1) 856.722 / (216 1) 857 747

Email : ext.rel@irsit.rnrt.tn

Web : <http://www.irsit.rnrt.tn>

III.3. Domaines d'activité :

La mission principale du centre de recherche était de promouvoir et de développer les potentialités et les compétences dans le domaine de l'informatique et des télécommunications. Dans le cadre de cette mission, l'institut était chargé de :

- d'œuvrer pour le développement de la recherche fondamentale et appliquée dans le domaine de l'informatique et des technologies associées,
- d'assurer une formation post universitaire, le recyclage aux techniques avancées et l'initiation à la recherche dans le domaine de l'informatique et des technologies associées,
- de contribuer à la sauvegarde et à la mise en valeur du patrimoine culturel et civilisatrice face à la transformation des structures économiques, scientifiques et culturelles dues aux mutations engendrées par l'informatique,
- de participer au développement de l'industrie informatique et des technologies associées par le transfert des innovations technologiques aux industriels.

III.4. Domaines Activités scientifiques :

Les activités de recherche-développement et de maîtrise et transfert de technologies de l'Institut s'opèrent autour des thématiques suivantes :

- Télécommunications et Réseaux :

- Interconnexion des Réseaux,
- Télématique,
- Multimédia,
- Technologies Spatiales.
- Systèmes d'Aide à la Décision :
 - Aide à la décision en gestion des ressources et système d'informations géographiques,
 - Aide à la décision en industrie.
- Arabisation et Communication Homme-Machine :
 - Traitement Automatique des Documents,
 - Traitement du langage naturel,
 - Communication parolière arabe.

CHAPITRE 2

ADMINISTRATIONS DES RESEAUX

I. Introduction :

Administrer, c'est savoir gérer, nous l'avons constaté, cela devient indispensable pour les systèmes informatiques d'aujourd'hui. La synergie qu'est une entreprise passe inévitablement par un vecteur d'information optimum ; or de nos jours, le médium de ce vecteur d'information n'est autre que l'informatique. La complexité de ce système n'a fait que croître continuellement tant par son étendu que par sa diversité. L'administration de ce système global est stratégique pour un bon fonctionnement global de l'entreprise, et cela reste ardu car il se matérialise par un ensemble d'éléments humains et matériels.

Il faut savoir faire évoluer le système d'information de l'entreprise afin d'éviter son obsolescence tout en gérant au mieux les coûts qu'induit cette mutation. Une solution d'administration a pour objectif d'assurer la disponibilité du réseau et d'augmenter le rendement global du système d'information de l'entreprise.

II. Fonctions d'administration :

L'I.S.O. décrit dans un document intitulé « Specific Management Information Services Specifications » les fonctions de bases pour l'administration d'un réseau. Il est à remarquer que ces spécifications sont valables à la fois pour les LAN's et pour les WAN's. On distingue principalement cinq fonctions de gestion.

II.1. La gestion des erreurs :

Il s'agit de fournir des fonctions permettant de détecter puis d'isoler et enfin de résoudre des erreurs qui surviennent sur le réseau. Les anomalies empêchent le fonctionnement normal du système où elles se produisent de façon temporaire mais aussi de façon persistante.

Le diagnostic du réseau passe par une identification des composants en panne ou mal configurés, suivi des indicateurs stratégiques pour prévenir les pannes.

Elles sont détectées par un mécanisme de détection d'erreurs utilisant souvent les recherches dichotomiques ou des outils de tests. L'administrateur réseau est prévenu très rapidement par des alarmes visuelles et sonores, voir même par l'émission de messages sur radio-messagerie.

Donc trois grandes caractéristiques à cet aspect gestion des erreurs. Tout d'abord une fonction de surveillance qui consiste à « montrer » les pannes, c'est à dire prendre en compte les événements inhabituels, à créer et à examiner des journaux pour valider et confirmer les fautes.

Puis une fonction de localisation des pannes va déterminer le ou les éléments causant ces troubles, ceci en tenant compte des effets de bords pouvant survenir. Enfin, une fonction de détermination des pannes qui s'effectue par l'analyse des journaux et par un choix d'actions curatives.

La gestion des anomalies représente une part importante de la gestion OSI, car le but de la gestion des réseaux est d'optimiser son utilisation or pour l'optimiser, il s'agit d'abord qu'il fonctionne « normalement ».

II.2. La gestion de configuration :

L'administration de réseau concerne également la gestion des configurations. Cette dernière manipule des données qui vont permettre de préparer l'initialisation et le lancement du fonctionnement des services d'interconnexion. Elle assure aussi la continuité de la première phase, par exemple, elle établit les paramètres qui contrôlent le comportement normal d'un système ouvert, elle diffuse les informations concernant l'état des systèmes, avertit d'éventuel changement d'état, etc.

Grâce à cette gestion des configurations, une topologie du réseau peut être dressée à tout moment en établissant un inventaire des différents éléments qui le constitue. Elle permet de contrôler la configuration sur le plan fonctionnel et opérationnel.

Il est primordial que le responsable du fonctionnement du réseau puisse à tout moment visualiser le plan du réseau et de maîtriser les informations relatives aux équipements connectés. Ainsi, comme pour la détection d'erreurs, des alarmes peuvent être mises en place lorsqu'une machine inconnue est indûment reliée au réseau.

Dans ce domaine d'administration, on constate trois facettes. La fonction d'installation qui consiste à paramétrier un élément, à l'initialiser et à le mettre en service, ou à le supprimer. Cette fonction permet le contrôle et la mise à jour des différents paramètres, et tient à jour les différentes versions implémentées. On peut aisément changer le protocole d'un routeur à distance, tout en gardant une trace de la version précédente.

Une fonction de contrôle et de surveillance consiste à rester à l'affût de tout changement d'état. Enfin une fonction de gestion des noms est nécessaire pour permettre une localisation sans ambiguïté de tout élément du réseau.

La gestion des configurations tient dans cette description OSI une place prépondérante car non seulement elle assure des fonctions autonomes propres à son rôle, mais elle propose aussi aux autres domaines fonctionnels d'administration, comme celui du traitement de la sécurité ou des erreurs par exemple, des fonctions de bases.

II.3. La gestion de sécurité :

L'administration réseau consiste à assurer la confidentialité de l'accès aux données critiques de l'entreprise qui peut traiter des informations hautement sensibles comme des données bancaires ou militaires par exemple. Certains réseaux, de part leur diversité et de part l'étendue de la population qu'ils touchent, nécessitent une mise en place de systèmes de sécurité spécifiques. L'administrateur doit donc être en mesure, grâce à des moyens efficaces, de contrôler ou d'interdire à certaines personnes l'accès à des données critiques. Des procédures d'enregistrement et de modification des priviléges des utilisateurs doivent être disponibles, tout comme un suivi des tentatives de violations d'accès.

Un encryptage des données est parfois nécessaire pour les véhiculer de façon plus sécurisée sur le réseau, l'administrateur devra contrôler ces codages.

Donc de façon générale, trois grandes fonctionnalités sont énoncées par l'OSI. D'abord une gestion de la confidentialité, par l'utilisation de clé d'encryptage et de mécanismes d'authentification. L'audit prend une part importante dans cette gestion pour surveiller les tentatives de connexion intempestives, ceci par la mise en place d'examens réguliers de journaux.

Enfin concernant les utilisateurs, il faut enregistrer et gérer aisément les droits de connexion de chacun.

La gestion de la sécurité d'un point de vue OSI couvre aussi bien les aspects relatifs au bon fonctionnement du réseau, que les aspects protection des éléments qui le constitue. Bien sûr les équipements des réseaux actuels ne sont pas tous en mesure d'offrir la totalité des fonctions citées ci-dessus. En fait le but de ces spécifications est que le responsable du réseau soit averti dès qu'un problème d'ordre sécuritaire intervient, c'est en ce sens que les fabricants s'investissent aujourd'hui.

II.4. La gestion des performances :

Il est vital pour un responsable de réseau de connaître l'état de fonctionnement de celui-ci, c'est à dire de pouvoir, à tout moment, analyser et identifier les éléments causant une dégradation des performances. Car il faut pouvoir maintenir des temps de réponse sur le réseau à un niveau constant et acceptable pour l'utilisateur. Un segment de câble surchargé, un routeur mal configuré, un serveur mal démarrer sont des problèmes qu'un administrateur doit détecter avant l'apparition d'anomalies lors de l'utilisation du réseau.

Il faut pouvoir suivre la charge du réseau et prévoir sa capacité à absorber des montées en charge. La gestion des performances permet d'évaluer le comportement des ressources OSI des systèmes ouverts et l'efficacité des communications des données sur le réseau. Ainsi des informations statistiques, des journaux de divers états sont régulièrement mis à jour pour informer l'administrateur qui pourra agir en modifiant les modes de fonctionnement de certains éléments pour accroître, optimiser les performances générales.

Donc dans ce pôle d'activité, il faut des fonctions de surveillance pour collecter des informations diverses sur les fautes et pour élaborer des statistiques sur le trafic. La gestion du trafic permet de réguler les données transitant sur le réseau de façon uniforme et optimisée. Une fonction d'observation de la qualité de service évaluera les paramètres de configuration, et leur incidence sur le fonctionnement du système. On pourra constater de cette manière si le temps de réponse est satisfaisant, ou la qualité d'une connexion à un serveur etc.

La gestion des performances d'un réseau permet donc de prévoir l'évolution du réseau et par conséquent de planifier les investissements nécessaires.

II.5. La gestion de coût :

La gestion de la comptabilité passe par quatre fonctions qui sont :

- La charge des ressources a fin de fixer et surveiller les ressources de réseau.
- Le coût de ressource : pour fixer le prix d'initialisation de ressource.
- La facturation pour récupérer l'information sur le coût à un utilisateur donné.
- La gestion des limites des utilisateurs dont le rôle de déterminer et de surveiller le quota d'utilisation des ressources.

III. Le responsable réseau et ses besoins :

III.1. Qui administre ?

L'administrateur est quelqu'un qui doit faire travailler de façon transparente, des machines variées, des systèmes hétérogènes et des utilisateurs qui ont tous des besoins divers. Une des qualités d'un responsable réseau est donc le sens des responsabilités, car désormais la plupart des informations vitales pour l'entreprise sont numérisées et sont véhiculées sur le réseau.

La disponibilité est aussi importante car certaines actions peuvent perturber le bon fonctionnement du réseau ce qui est préjudiciable donc parfois le responsable doit effectuer ses manipulations, aux heures creuses, souvent le soir très tard. Un bon administrateur réseau fait preuve d'une rapidité d'intervention pour limiter les perturbations dans le temps, 90% du travail est consacré à la résolution de problèmes déjà rencontrés, ce qui prouve que l'expérience et une confrontation à un nombre important de cas fait la qualité de l'homme.

Des qualités relationnelles sont indispensables pour comprendre les attentes des utilisateurs, mais aussi parce que l'administrateur est à la fois policier et pompier.

L'administration réseau nécessite une compétence humaine, tant sur le plan opérationnel pour la gestion que sur le plan fonctionnel pour la planification. On peut dégager trois objectifs principaux pour une équipe d'administration dans leur tâche. D'abord, à court terme, il est nécessaire de pouvoir exécuter des actions d'exploitation du réseau au quotidien: prise en compte et résolution de panne, contrôle des performances, gestion de la configuration, de la sécurité et tous les événements pouvant survenir sur un réseau. A moyen terme, il faut pouvoir coordonner, déployer et mettre à niveau le réseau. Enfin à plus longue échéance, une planification de l'évolution cohérente du réseau est nécessaire. Ceci passe par une politique globale de sécurité, de configuration, de performance et bien sur de coût.

III.2. Administrer quoi ?

Le système d'administration doit permettre au responsable d'agir à distance, sur tous les systèmes du réseau. En fait on peut dire que l'outil d'administration idéal serait celui qui permet à partir d'une seule station, de visualiser et de contrôler tout ce qui se passe sur le réseau. Mais un tel outil n'existe pas vraiment, notamment à cause de la disparité et de l'hétérogénéité des matériels présents dans l'architecture. Les informations parvenues d'une carte réseau et d'une application de base de donnée relationnelle ne sont pas comparables.

D'une façon générale on peut découper les éléments d'un réseau avec deux optiques. Une première en terme de catégorie, c'est à dire si l'élément est du matériel, du « Hard » ou si c'est du logiciel, du « soft ». Une autre façon de classer les éléments à gérer serait de tenir compte de leur localisation. A un niveau local on trouverait par exemple les cartes réseau, les serveurs, les NOS, les postes de travail, à un niveau départemental on aurait les ponts, les routeurs et autres commutateurs, et enfin à un niveau nettement plus « distant » on considère les opérateurs, les lignes...

IV. Conclusion :

Le système d'administration d'un réseau doit permettre à son responsable d'intervenir à partir d'une station sur tout son parc indépendamment des différents systèmes ou matériels. Nous l'avons vu, une solution d'administration doit répondre à certains critères, à certaines attentes de la part de l'administrateur, et ceci de façon stable et durable. Il ne s'agit en aucun cas d'être handicapé dans sa gestion par un nouveau matériel ou une nouvelle application, c'est pourquoi afin de garantir une pérennité dans l'administration de tous les éléments d'un réseau, il est préférable d'adopter une plate-forme d'administration basée sur les standards aux normes internationales.

CHAPITRE 3

LE PROTOCOLE SNMP

I. Introduction :

Vers la fin des années 70, les petits réseaux isolés évoluèrent vers des grands réseaux interconnectés. Ils deviennent de plus en plus difficiles à gérer.

En 1988, l'IAB approuva le développement de SNMP. A ce moment là, SNMP devait être une solution à court terme. Ensuite 1989 SNMP a été adopté par de nombreux constructeurs et est devenu à ce jour un standard très répandu de gestion réseau.

II. Architecture de SNMP :

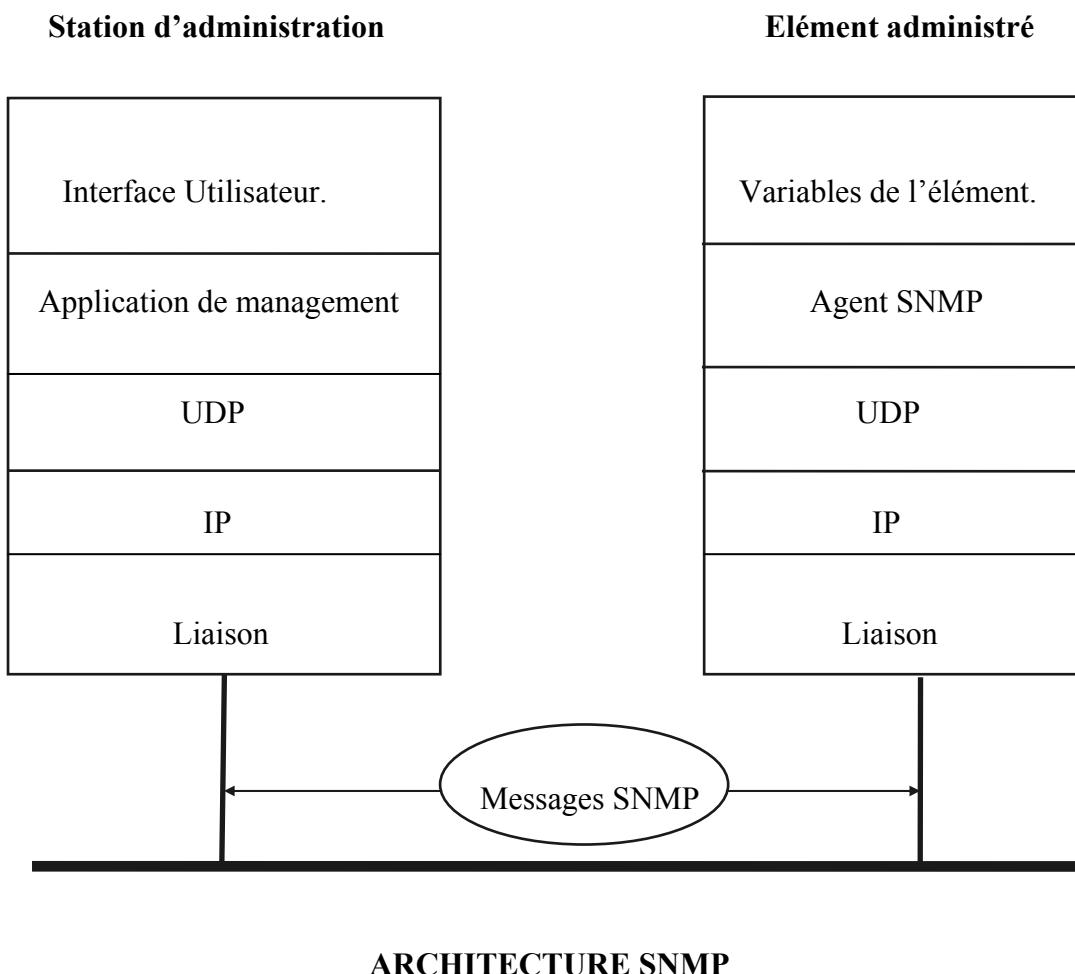
II.1. Principe de base :

L'architecture SNMP a été initialement développée pour être supportée par le système d'exploitation UNIX sous protocole TCP/IP et fonctionner en mode UDP (User Datagram Protocol). C'est un point à remarquer, le protocole SNMP va utiliser une méthode de transport commune à l'échange d'information et non pas particulière à l'administration. L'architecture SNMP est composée par un ensemble de nœuds gérés tels les routeurs, les serveurs, les ponts etc. Des agents qui ont pour tâche de faire exécuter les fonctions de gestion ordonnées par la ou les stations d'administration. Le protocole SNMP est donc chargé de véhiculer les informations de gestion entre les stations d'administration et les agents présents sur différents éléments du réseau. Pour revenir à UNIX, les agents sont sous forme de processus (daemons) lancés au démarrage de la machine hôte.

Donc le modèle SNMP est composé :

- Des nœuds managés, entités managées qui contiennent des agents.
- Une station d'administration contenant une application de management.
- Du protocole SNMP qui permet le dialogue entre les agents et la station d'administration.
- D'informations administratives.

Le schéma ci-dessous permet de fixer ces premières notions:



II.2. Les acteurs de l'administration :

II.2.1. La station d'administration

Cette station, appelée Network Management Station (NMS), possède une, voir plusieurs applications d'administration de réseau et un protocole dédié à cette tâche. La gestion des éléments du réseau doit les perturber au minimum, donc la station d'administration devra être puissante pour gérer les problèmes elle-même. La politique retenue pour SNMP est la minimisation du nombre et de la complexité des fonctions de gestion donc de la charge de travail des machines gérées. Néanmoins de nos jours, de nombreux produits dit intelligents possèdent des fonctions d'administration.

La station d'administration surveille l'état du réseau en centralisant les informations venant des éléments administrés. Pour cela, elle leur demande périodiquement des informations pour connaître leur niveau de fonctionnement, on parle alors de polling. Cette station supporte à la fois le protocole de gestion et les opérations de gestion inhérentes à une bonne administration, c'est à dire qu'elle sait interpréter les informations administratives véhiculées par le protocole et agir ou permettre d'agir en conséquence.

Elle doit être capable de conserver un historique des états précédents, pour faire des statistiques par exemple, afin d'établir une politique à long terme.

Une station d'administration peut gérer plusieurs nœuds et plusieurs stations de gestion peuvent gérer des nœuds identiques.

II.2.2. Les entités managées :

Les entités managées, aussi appelées nœuds managés (Managed Nœuds), représentent différents matériels que l'on peut regrouper en trois catégories. Une première composée des serveurs, terminaux, stations de travail, des mainframes, des imprimantes; une autre regroupant les moyens de transmission tels les ponts, les répéteurs, les hubs; et enfin une dernière catégorie pour le système de routage.

Chaque nœud managé présente deux points communs marquant leur appartenance à l'architecture SNMP. La première de ces caractéristiques est la MIB (Management Information Base) qui regroupe tous les paramètres associés à l'entité, ceux-ci sont sans cesse remis à jour pour refléter l'état du nœud. Ces différentes bases de données sont gérées par des agents, deuxième élément SNMP, qui permettront de mettre à jour, de faire des requêtes sur la MIB.

II.2.3. Le protocole de gestion :

C'est le moyen de transport des informations, il spécifie la nature des communications entre le programme client, situé sur la station de gestion et le programme présent sur le nœud, l'agent. C'est le langage entre les éléments, il ne définit pas d'objet, il permet seulement le dialogue.

Le protocole SNMP est asynchrone, de type requête/réponse donc non connecté, il utilise le protocole de transport UDP pour véhiculer ses informations. Ce dernier fait a une incidence sur la sécurité et la fiabilité car on ne sait par exemple s'il y a perte de paquet. Les applications

s'envoient les messages directement, il n'est pas utile d'ouvrir, de maintenir, et de fermer des connexions de niveau application pour effectuer des échanges d'informations.

Un nœud est vu, identifié dans son état par différentes variables . Il faut pouvoir lire et modifier ces variables pour contrôler le nœud; donc le protocole permet de lire, d'écrire de façon pertinente, une autre opération nécessaire pour l'entité managée, une possibilité de signaler à la station de management, un événement extraordinaire. Précisons ces deux concepts que sont la lecture/écriture et l'émission d'événement.

Les matériels sont différents donc ils ont des variables plus ou moins spécifiques. Il faut savoir que toutes ces variables sont regroupées pour former un groupe particulier. Ainsi lors de la spécification de SNMP on a voulu rester ouvert aux variétés techniques de chaque constructeur et aux améliorations techniques futures. Le problème résidait dans l'accès et la manipulation de cette liste de différentes variables spécifiques car il ne s'agit pas toujours de simple valeur scalaire. Il y a par exemple les tables de routage IP qui comportent un nombre de colonne variable. Donc le protocole devra pouvoir trouver une valeur précise dans une liste, mais aussi savoir faire un parcours efficace de toutes les valeurs du tableau.

Des événements extraordinaires peuvent se produire sur un nœud, il faut alors prévenir la station de management appropriée. Cette notification immédiate porte le nom de « Trap ». Il subsiste un inconvénient à ce type d'alerte car si le message comporte trop d'information, l'agent passe beaucoup de temps à la valider (bonne réception, bon destinataire, réception de Ack ok, ...) ce qui peut augmenter les troubles ! On peut même dire que cela peut provoquer d'autres troubles, donc même générer d'autres Traps et créer une congestion du bâton et paralyser le trafic.

On restreint la possibilité de détection d'erreurs et par conséquent d'émission de trap; la politique est de faire alors du Polling, c'est à dire d'interroger périodiquement les nœuds managés afin de connaître leurs états. Cette méthode a l'avantage de donner une vue globale du réseau mais l'inconvénient d'augmenter le trafic de part le nombre de requêtes sans cesse émises et un temps de détection en cas de problème assez aléatoire. Ces deux inconvénients prennent de plus en plus d'importance avec le nombre d'entités managées.

Dans le cas de SNMP, quand il y a des événements extraordinaires, le nœud émet une alerte unique et la station de management prend elle-même en charge les actions à réaliser, les dialogues

pour résoudre le problème. Reste un détail gênant, SNMP utilise le protocole de transport UDP donc il n'y a aucune garantie que le Trap ait bien été reçu par la station de management. A ce sujet,

le choix d'un protocole de transport est fondamental car il a des impacts directs sur la façon de manager, de concevoir le protocole d'administration. Si on choisit un mode connecté, il y a la « difficulté » de connexion, de maintien du dialogue etc. ce qui augmente les délais de réactions et fait chuter l'efficacité du management.

II.2.4. Les informations administratives :

Il s'agit de définir la syntaxe des informations administratives échangées entre un nœud et la station d'administration. Une unité d'information sera nommée « managed object » alors qu'une collection d'objet sera organisée sous forme de « MIB » (Management Information Base)..

Le plus important pour la définition de la méthode de description de l'information, est de ne pas être restrictive, c'est à dire de rester ouverte aux évolutions techniques et à l'hétérogénéité du marché. Un concepteur de routeur par exemple pourra implémenter une unité d'information particulière à son produit tout en restant compatible avec le protocole d'administration.

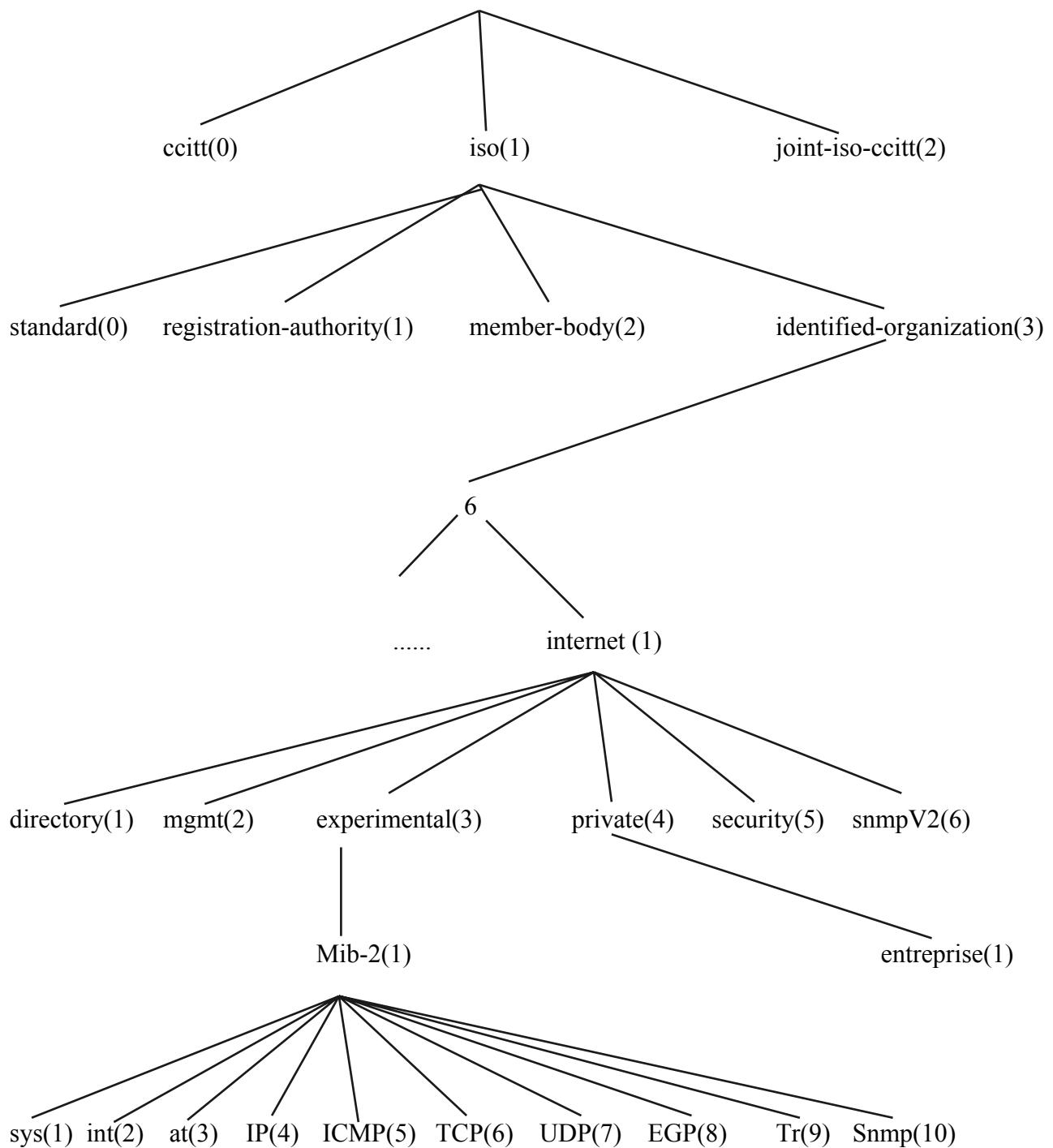
Pour formaliser les informations, il a été défini une syntaxe abstraite ne tenant pas compte des contraintes machine. Ce langage proposé par l'OSI s'appelle Abstract Syntax Notation One (A.S.N.1). Mise en application pour les applications d'administration, ce langage permet de définir les objets managés et le format des informations échangées par le protocole de management. Bien sur ASN1 est un langage formel, avec sa grammaire mais grâce aux B.E.R. (Basic Encoding Rules), on peut le coder avec n'importe quel langage machine, comme le C ou le Pascal. En fait, ASN1 a été utilisé dans l'optique de la transition de SNMP vers le modèle de protocole d'administration d'OSI.

III. Management information base (MIB) :

Chaque entité gérée possède un certain nombre de variables qualifiant son état. Ces données sont représentées par des objets qui sont organisés de façon logique sous forme de MIB. Les MIBs sont des sortes de base de données décrites en ASN.1. Elles sont présentes dans une zone virtuelle de stockage et les valeurs des objets qu'elles décrivent sont accessibles par un processus (deamon) appelé agent qui réagit aux sollicitations, aux requêtes SNMP émises par l'application de gestion.

Donc chaque agent décrit son environnement administrable propre dans une MIB (structurée suivant le standard SMI définissant un système hiérarchique d'arbres utilisant le langage ASN-1).

Une partie de la représentation de l'arbre officiel est la suivante:



Il existe une arborescence organisant les différents objets, ainsi il y a trois sortes de MIB:

- Les standards qui sont développées par les groupes de travail de l'IETF puis officialisées par l'IESG. Ces MIBs sont localisées dans le sous-arbre mgmt.
- Les expérimentales comme leur nom l'indique ont été développées, mais pas standardisées, donc elles se trouvent dans la branche experimental
- Les spécifiques aux entreprises sont nécessaires pour y placer toutes les particularités d'un produits qui ne rentre pas dans les standards. Les fabricants utilisent bien sur beaucoup ce type de Mib se trouvant dans la branche entreprises.

La base de donnée que représente la MIB possède des objets qui sont nommés de façon unique et triés par ordre lexicographique (le groupe système, Interfaces ...). Ce dernier fait favorise les accès séquentiels des variables de la base par des opérations de lectures successives, action GetNext avec SNMP, chaque objet identifié dans la MIB a un mode d'accès (ro : read only, rw : read write ou na : no access).

La base peut être enrichie par de nouveaux objets où groupes d'objets appartenant à l'une des trois catégories citées ci-dessus (standard, entreprise, experimental), il s'agit d'ajouter un nœud à l'arborescence. Un premier standard de MIB , la MIB1, fut créé avec un certain nombre d'objets répertoriés; puis dans un second temps, vers fin 1990 une nouvelle version, la MIB-II fut standardisée intégrant RMON, et on peut dire qu'une MIB-III va sûrement bientôt voir le jour . Le sous-arbre de private permet à quiconque de façonne ses propres MIB's; il est à noter qu'il existe des compilateurs de Mib pour que lorsque l'on installe une MIB propriétaire, elle soit interprétable par le standard . Il y a deux compilateurs de MIB disponibles librement, « mosy » présent sur le système BSD et « smic » qui permet de faire des MIB's sur système DOS et UNIX.

IV. Echange de données :

IV.1. Les commandes SNMP :

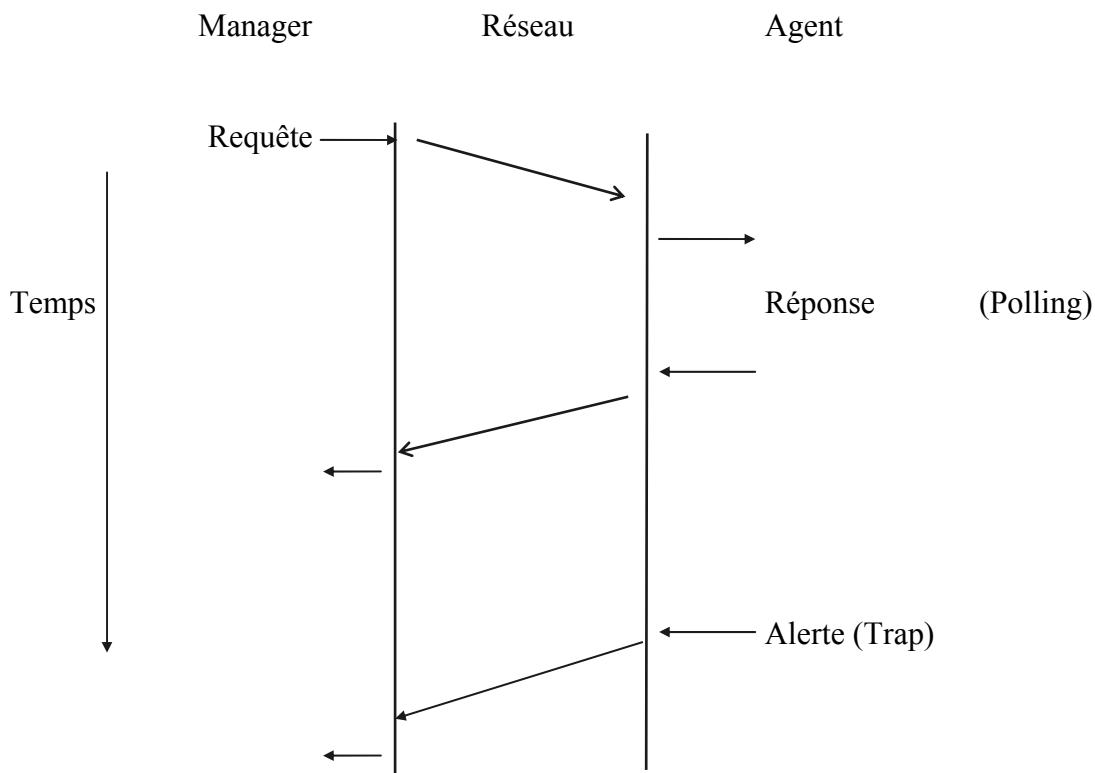
Les spécifications du protocole SNMP est présente dans la RFC 1157. Ce protocole a une approche particulière pour la gestion du réseau; il ne propose pas une kyrielle de fonctions spécifiques, mais deux opérations de bases qui peuvent être qualifiées conceptuellement de

lecture/écriture. L'avantage de ces fondements est bien entendu la simplicité, la stabilité et la flexibilité. Simple car deux opérations basiques faciles à implémenter et à comprendre, on évite

tout un panel de cas particulier. Stable car la définition de nouveaux objets s'intègre parfaitement dans la base de données.

Les interactions entre la station de management et l'entité gérée, où plutôt son agent, se déroule la plupart du temps avec le même schéma: requête de la station, réponse de l'agent. L'initiative de ce dialogue est à la station de management qui périodiquement va interroger tous les agents sous sa responsabilité pour évaluer la situation du réseau, on parle alors de Polling. On sent immédiatement les limites de ce genre de politique, on n'a pas de gestion en temps réel, sans parler du risque de congestion du réseau qui paralyse les dialogues d'administration.

Il arrive parfois que des événements exceptionnels se produisent sur un nœud, qui peut informer la station d'administration par l'émission d'une alerte (Trap).



Ce schéma résume les opérations de SNMP, le sondage (polling) de la station d'administration qui se traduit par l'émission d'une requête, l'agent répond. Si un événement inopiné survient,

l'agent, s'il le considère suffisamment important, émet une alerte en direction de la station de management.

Il est à remarquer que suite à une requête, on peut distinguer quatre types de réponse qui sont :

- Une réponse sans erreurs et sans exceptions
- Une réponse avec erreur.
- Une réponse avec une ou des exceptions.
- Un TimeOut.

Du point de vue de l'administrateur, le protocole SNMP reste transparent, c'est l'interface de l'application de management qui traduit différentes fonctionnalités en requêtes qui se scindent en deux catégories, lecture(GET) / écriture (SET). Dans la première, on a trois types d'opérations qui vont être maintenant présentées.

L'opérateur « get-request » permet de lire le contenu d'une variable scalaire dans la base de gestion. On passe en argument le libellé de la variable à laquelle on veut accéder, la réponse obtenue est une valeur du type demandé. Si le libellé est invalide, la valeur noSuchInstance est placée dans la réponse. Si l'agent ne connaît pas cette variable où qu'il ne l'a pas référencée, il retourne noSuchObject comme réponse. Dans la version originale de SNMP, les valeurs exceptionnelles n'existaient pas, on retournait une valeur NULL et le champ ErrorStatus étaient positionné par exemple à NoSuchName si le type demandé n'était pas connu de l'agent

L'opérateur « get-next-request » est différent du précédent par le fait que l'on obtient la valeur suivante dans la base de gestion de celle passée en argument. La recherche se fait dans l'arbre par un parcours lexicographique des noms des objets. Deux sortes d'erreurs peuvent être retournées: tooBig ou genErr. L'exception endOfMibView est répondu si il n'y a plus d'autre instance suivant celle spécifiée.

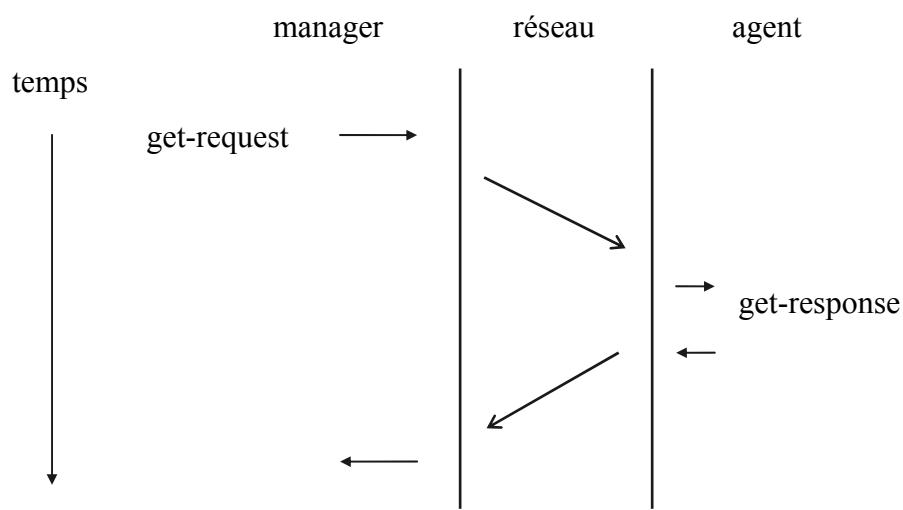
Pour conclure cette catégorie de fonction de lecture, il y a l'opérateur « get-bulk-request ». En fait dans la version originale de SNMP, l'opérateur « get-next-request » était souvent utilisé pour lire toute une série de valeurs contigües. Donc afin de minimiser le temps d'échanges de données, il a été implémenté pour SNMPv2 cet opérateur qui retourne toute une séquence de valeur sans y être invité pour chacune d'entre elles. Les arguments sont au nombre de trois et portent les noms de non-repeaters, max-repetitions et une liste d'objet. En fait, cette requête permet d'avoir les

successeurs directs comme avec get-next-request des non-repeaters premiers éléments de la liste. Et pour le reste des éléments de la liste, c'est à dire L-N si la liste a L éléments, la requête

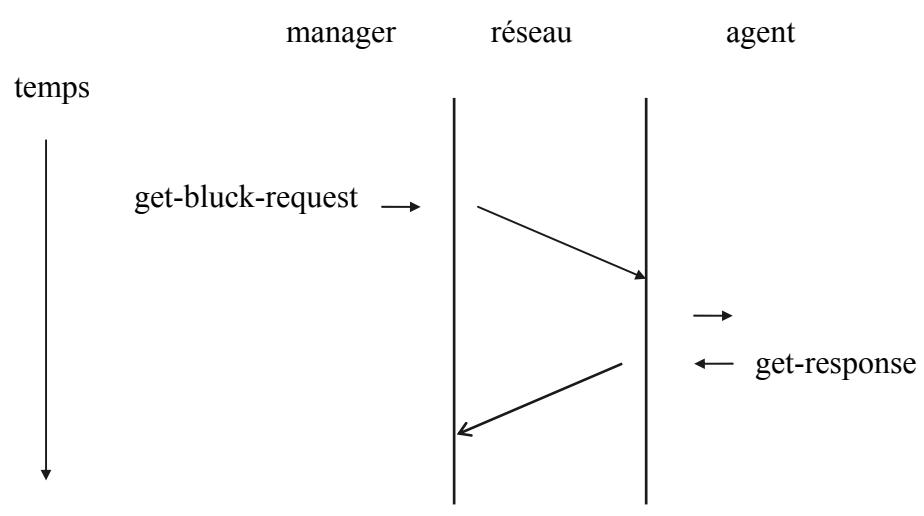
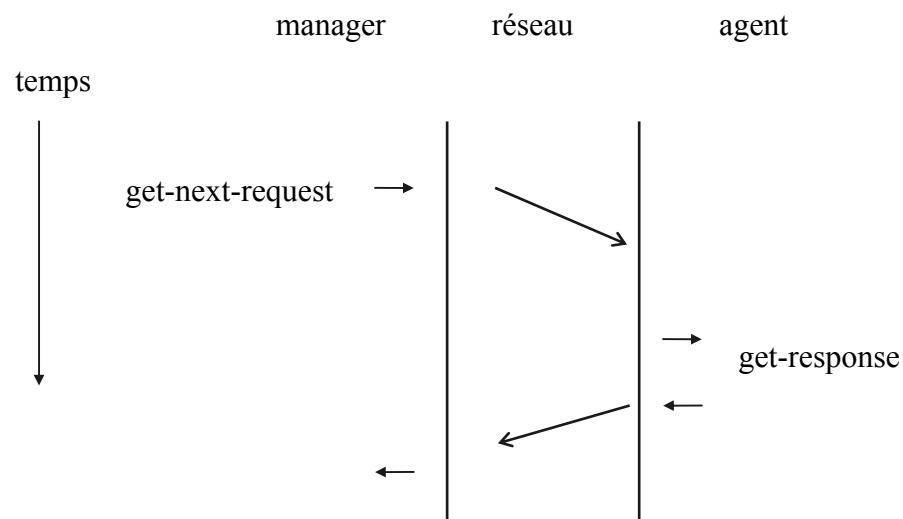
retournera respectivement leur max-repetitions successeurs. L'opérateur « get-bulk-request » génère le type d'erreur genErr et retourne la valeur endOfMibView si elle arrive en fin d'arbre.

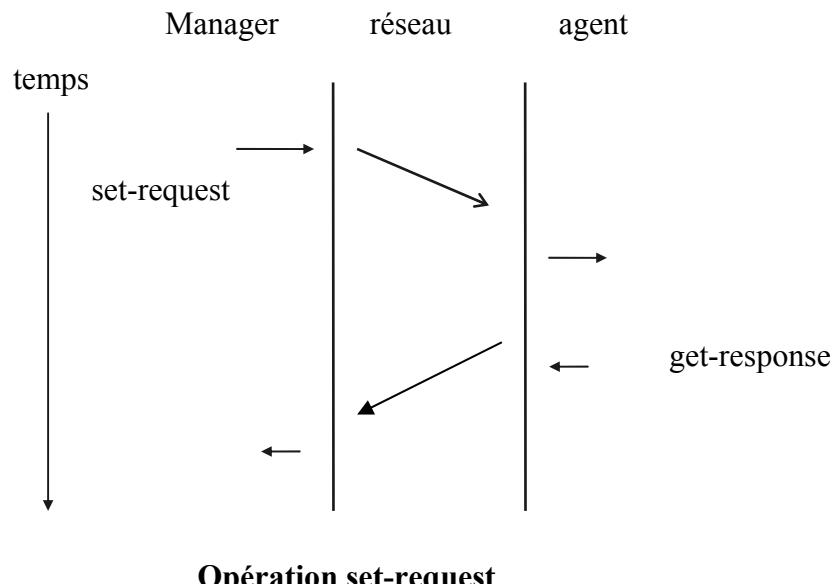
Pour écrire, on ne dispose que d'une seule fonction appelée « set-request ». Cette opération permet d'affecter une valeur à une variable en prenant comme argument, le nœud sur lequel doit être fait la modification, l'identificateur de la variable et bien sur la nouvelle valeur pour la dite variable. La valeur renvoyée est celle de la nouvelle affectation. Cette opération permet d'ajouter ou de supprimer, dans une table, des lignes. En fait cette opération présente plusieurs cas, la variable à affecter n'existe pas dans la Mib, la nouvelle valeur n'est pas correcte, la longueur n'est pas correcte etc. L'opération d'affectation se passe en deux phases réalisées par l'agent qui considère la requête puis qui éventuellement l'exécute. L'agent peut par exemple créer la variable si elle n'existe pas. Si l'opération d'écriture ne peut se faire, SNMPv2 par rapport à SNMPv1 permet une émission d'erreur spécifique, SNMPv1 dit « il y a une erreur », SNMPv2 précise laquelle !

Les schémas ci dessous reporte les quatre types d'opération



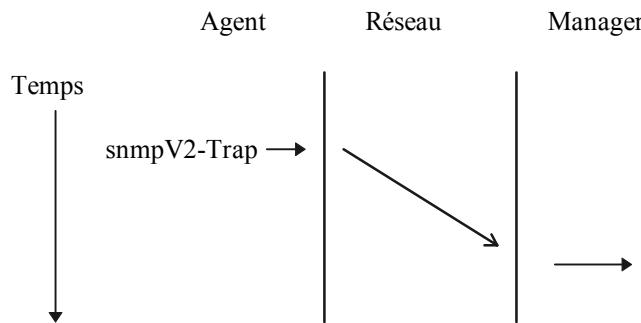
Opération get-request





Vous l'aurez compris l'opération get-reponse permet à un nœud géré de répondre à une requête provenant d'une station d'administration.

Les opérations évoquées jusqu'à là étaient toujours à l'initiative d la station de management, il existe une opération attribuée aux nœuds pour leurs permettre de signaler un événement non prévu, c'est le Trap. Dès que l'agent détecte un événement extraordinaire, il envoi une alerte (snmpV2-trap pour la version SNMPv2) qui peut être envoyée à une ou plusieurs stations de management. Contrairement aux Traps de la version originale de SNMP, le snmpV2-trap permet d'être envoyé dans des PDU's (Packet Data Unit) similaires aux autres requêtes dont nous avons parlées ci-dessus. Cette notion de PDU va être précisée dans le prochain paragraphe.



Ce trap se configure au niveau de l'agent et est envoyé sur le port de snmp. Les deux premières variables spécifiées dans le message d'alerte sont, un relevé de date de événement (sysUpTime.0) et l'identification de l'entité émettrice du Trap (snmpTrapOID.0).

La structure d'un trap est la suivante:

NOM	SIGNIFICATION
entreprise	valeur de l'objet sysObjectID de la MIB de l'agent (notation ASN.1)
agent-addr	valeur de l'objet Network Address de l'agent
Generic-trap	voir tableau des traps SNMP
specific-trap	identifie l'entrepriseSpecific trap (trap spécifique à l'agent donc non standardisée)
time-stamp	valeur de l'objet sysUptime de la MIB de l'agent lorsque l'événement s'est produit
variable-bindings	liste de variables contenant des informations sur le trap.

generic-trap définit un certain nombre de traps standardisés.

Un champ, dans le Trap précise la nature de l'alerte, sept cas possibles

Nom	Numéros SNMP	Signification
coldStart	0	l'agent se réinitialise et les objets peuvent changer (changement de configuration)
warmStart	1	l'agent se réinitialise mais les objets ne sont pas modifiés (pas de changement de configuration)
linkDown	2	une des interfaces de l'agent est tombée (la première variable dans la liste variable-bindings identifie l'interface)
LinkUp	3	une des interfaces de l'agent est à nouveau opérationnelle (la première variable dans la liste variable-bindings identifie l'interface)

AuthenticationFailure	4	un message SNMP a été reçu d'une entité SNMP et il y a eu un problème d'authentification en fonction du nom de communauté et des droits d'accès qui lui sont accordés.
EgpNeighborLoss	5	un EGP peer (EGP = Exterior Gateway Protocol) est tombé (la première variable dans la liste variable-bindings contient l'adresse IP).
EnterpriseSpecific	6	certaines événements dépendent de l'agent et ne sont donc pas standardisés, dans ce cas le numéro du trap est donné dans le champ specific-trap.

Il est à signaler que des Traps peuvent être envoyés d'une station de management vers une autre station de management ceci pour l'informer de fait. Le message est alors un « inform-request ».

Précisions que les fonctions sont acheminées par UDP par le port 161 pour les get/set et le port 162 pour les traps. Comme SNMP est un protocole asynchrone de requêtes/réponses, une entité n'a pas besoin d'attendre une réponse après avoir envoyé un message.

IV.2 Le format des messages SNMP :

Comme il a été dit en présentation, le protocole SNMP a ceci de particulier qu'il utilise un protocole à part entière pour le transport de ses requêtes qui n'est autre que UDP pour la version originale. En ce qui concerne la deuxième version de SNMP, d'autres protocoles de transport sont disponibles. Donc lorsque l'on envoie un message SNMP, il faut préciser sa version et c'est aussi utile dans la compatibilité entre les deux versions. Dans le message SNMP, il y a un champ appelé « community » que l'on précisera dans le prochain paragraphe et enfin l'unité d'information SNMP qui peut être une requête, une réponse, ou un trap.

Donc la structure générale des messages du protocole SNMP est :

Version	Community	SNMP PDU

La version SNMP, définie par le RFC 1157, est la version 1 par exemple :

ASN.1 et les BER's, serviront à définir les structures des PDU's (Protocol Data Units) échangés par le protocole de gestion. Les messages ont des structures différentes selon leur nature; voyons ce point plus en détail. Il est à noter que les requêtes Get, GetNext, Set et GetResponse ont le même format de PDU.

PDU type	request-id	error-status	error-index	variable-bindings
----------	------------	--------------	-------------	-------------------

Le PDU-type n'est pas décrit en ASN.1, donc les cinq valeurs qu'il peut prendre est définit implicitement.

Le champ request-id ce champ est un entier utilisé afin d'identifier la requête envoyée par la station d'administration à l'agent. Il est nécessaire afin d'identifier une réponse en provenance de l'agent et aide à la détection d'erreurs, dans le cas où deux réponses arrivaient à la NMS dans le même laps de temps avec même identifiant de requête.

En fait lorsque l'on a un PDU de type GetResponse, les champs error-status et error-index ont des valeurs, sinon dans les trois autres cas (Get, GetNext, Set) ils sont mis à zéro, on peut lire et écrire des PDU's SNMP avec seulement deux procédures.

Le champ error-status sert à indiquer le statut de la requête envoyée par la station d'administration Il est donc uniquement utilisé par l'agent dans l'envoi de la PDU GetResponse à la NMS. S'il est différent de 0, c'est que la requête précédente a généré une erreur (ci-dessous). Dans tous les autres PDU (Get, GetNext, Set), ce champ doit prendre la valeur 0.

Nom Erreur	Valeur Erreur
noError	0
tooBig	1
noSuchName	2
badValue	3
readOnly	4
genError	5

Le champ error-index est utilisé dans la PDU GetResponse en cas d'erreur, afin de fournir plus d'informations sur celle-ci. C'est un index qui pointe sur la première variable du champ

VarBindList ayant causée l'erreur, sa valeur étant la position de cette dite variable dans la liste (le décompte commençant à partir de 1). Les variables sont des instances d'objet.

Le PDU de Trap a un format particulier :

PDU	type					
	enterprise	agent-addr	generic-trap	specific-trap	time-stamp	variable-bindings

Le champ enterprise contient l'objet ayant généré l'alerte, c'est le sysObjetID de l'élément.

Le champ agent-addr contient l'adresse IP de l'objet ayant généré l'alerte.

Le champ generic-trap contient un entier de type generic-trap prédéfini en standard pour le protocole SNMP. Ces valeurs ont été vues dans le paragraphe précédent, en voici un résumé:

Valeur du type generic-trap	Libellé
0	coldStart
1	warmStart
2	linkDown
3	linkUp
4	authenticationFailure
5	egpNeighborLoss
6	enterpriseSpecific

Le champ specific-trap contient les valeurs de traps définies pour un constructeur ou une entreprise particulière. Il faut noter d'une part que ce champ est toujours présent, même si le trap est de type générique et, d'autre part, que la valeur qui y est contenue dépend de l'implémentation.

Le champ time-stamp contient le temps écoulé entre la dernière initialisation où réinitialisation de l'entité et la date où le trap a été générée. C'est la valeur de sysUpTime, elle est exprimée en centième de seconde.

Dans tous les cas, le champ variable-bindings possède successivement le nom de la variable puis sa valeur et ceci n fois le nombre de variable.

Les opérations de SNMP sont indépendantes de la méthode de transport qu'il utilise, il a seulement besoin d'un service de transport. La version original utilise le User Datagram Protocol (UDP) mais cela semble restrictif de part l'ouverture qu'offre SNMP car n'importe quel protocole de transport en mode connecté où non est tout à fait possible. Un protocole de transport doit

spécifier comment il organise les données en suite d'octet et comment il transmet ces paquets sur le réseau. Donc beaucoup de protocoles répondent à ces prérogatives, et par conséquent s'offrent à SNMP. Attention, il ne faut néanmoins pas oublier que les objets SNMP sont définis par ASN.1 et organisés en une structure par les BER's; donc lorsque l'on reconstitué le message à partir de la suite d'octet reçut, il faut retrouver la même sémantique, les bonnes informations ASN.1. Les BER's décrivent comment encoder les types ASN.1 qui sont d'ailleurs relativement simples, un entier, une suite de valeur ... Lors de l'encodage des types ASN.1 par les BER's, il y a ajout de deux champs; un tag pour indiquer la version de ASN.1 utilisée et un champ longueur pour donner la longueur de la valeur encodée comme ASN.1 permet la définition de structure complexe. Bien sur le champ de donnée ASN.1 suit ces deux champs.

Depuis, la deuxième version de SNMP, a introduit la possibilité d'implémenter celui-ci sur d'autre protocole de transport dont OSI en mode non connecté (ConnectionLess mode Network Service (CLNS)), Novell Internetwork Packet eXchange(IPX) et AppleTalk. SNMPv2 définit formellement l'implémentation du protocole sur ces couches de transports.

Toute une sous-arborescence, les snmpDomains, a été ajoutée à l'OID. Pour spécifier l'appartenance d'une entité à un type de protocole de transport, l'objet partyTDomain a les valeurs suivantes :

- **snmpUDPDomain** TCP/IP: le standard UDP.

C'est le plus courant, peut être à cause de l'histoire de SNMP. Il permet une certaine forme de compatibilité entre SNMPv1 et SNMPv2. Par convention, les agents SNMP écoutent sur le port UDP 161 (Get, GetNext, Set, Response) et notifie sur le port UDP 162, V2 continuera.

- **snmpCLNSDomain** OSI : ConnectionLess mode Network Service.

Les messages SNMPv2 utilisent un TSDU (Transport Service Data Unit) pour les ConnectionLess mode Transport Service (CLTS).

- **snmpCONSDomain** OSI: Connection-Oriented Network Service.

Les messages sont transportés en mode Connection-Oriented

- **snmpDDPDomain** AppleTalk: Datagram Delivery Protocol.

Les messages sont envoyés à travers un Datagram Delivery Protocol (DDP) datagram, qui est utilisé avec le type=8. Et l'entité SNMPv2 agit dans son rôle d'agent avec le numéro 8 de socket DDP, alors que c'est le numéro 9 de socket DDP utilisé pour exécuter les notifications (traps).

- **snmpIPXDomain** Novell: Internetwork Packet eXchange.

Le SNMPv2 est sérialisé dans des IPX datagrams, il utilise les paquets de type = 4. Son rôle d'agent est à l'écoute sur les sockets IPX de numéro 36879 (900FH) et envoie les notifications sur le numéro de socket IPX 36880 (9010H).

V. Conclusion :

SNMP n'est néanmoins pas la panacée, c'est un protocole gourmand en bande passante à cause de son principe de scrutation (polling), il est limité par son unique protocole de transport TCP/IP et surtout, il n'a aucune option de sécurité. En 1992, la version 2 de SNMP va combler ses lacunes originelles. D'abord pour réduire les échanges entre les entités, des nouvelles requêtes comme GetBulk vont réduire les charges de dialogue, tout comme l'amélioration de l'identification des

alertes (snmpv2Trap) où l'instauration de dialogue entre entités manageantes (InformRequest) qui fait gagner de la pertinence au processus de management. Une ouverture conséquente s'est opérée quand au mode de transport des PDU's de SNMP, ainsi des protocoles tels que IPX, DDP etc. sont à présent supportés. La liste est en voit de s'allonger considérablement.

La sécurité inexistante de la version 1 de SNMP, mis à part le principe de communauté (community name), a été entièrement repensée. Des procédés de cryptage et d'authentification ont été mis en place pour garantir un sécurité optimale.

Enfin, des principes de dialogue inter-gestionnaire et de gestion locale par des agents intelligents ont été définit pour permettre une administration plus délocalisée. Ce qui dans le même temps diminue les dialogues avec une NMS.

Donc SNMP est devenu un protocole d'administration à part entière, qui a sut s'assurer une pérennité durable de par sa simplicité, son efficacité et son ouverture vers les évolutions techniques et les contraintes économiques.

CHAPITRE 4

ARCHITECTURE DE MIB ORACLE

I. Introduction :

Dans ce chapitre nous présentons, l'outil d'administration d'oracle (Oracle Entreprise Manager OEM) ainsi que la particularité des composantes SNMP au niveau d'un agent ensuite nous détaillons l'architecture de la MIB privée d'Oracle et ces différentes tables qui contiennent ces variables.

II. l'outil d'administration d'Oracle :

II.1 Présentation

Oracle Entreprise Manager est un outil de gestion de système qui fournit une solution intégrée pour la gestion d'un environnement hétérogène. Il combine une console graphique, des agents des services communs et des outils pour fournir une plate-forme de gestion de systèmes intégrée, complète et compréhensive pour des produits d'Oracle dirigeants.

Oracle Entreprise Manager n'emploie pas SNMP directement. Au lieu de cela, il communique avec l'agent sur SQL*NET employant Transparent Network Substrate (TNS). L'agent écoute les requêtes SNMP ensuite il transmet ces requêtes vers le manager d'entreprise d'Oracle.

De la Console d'Entreprise Manager, vous pouvez :

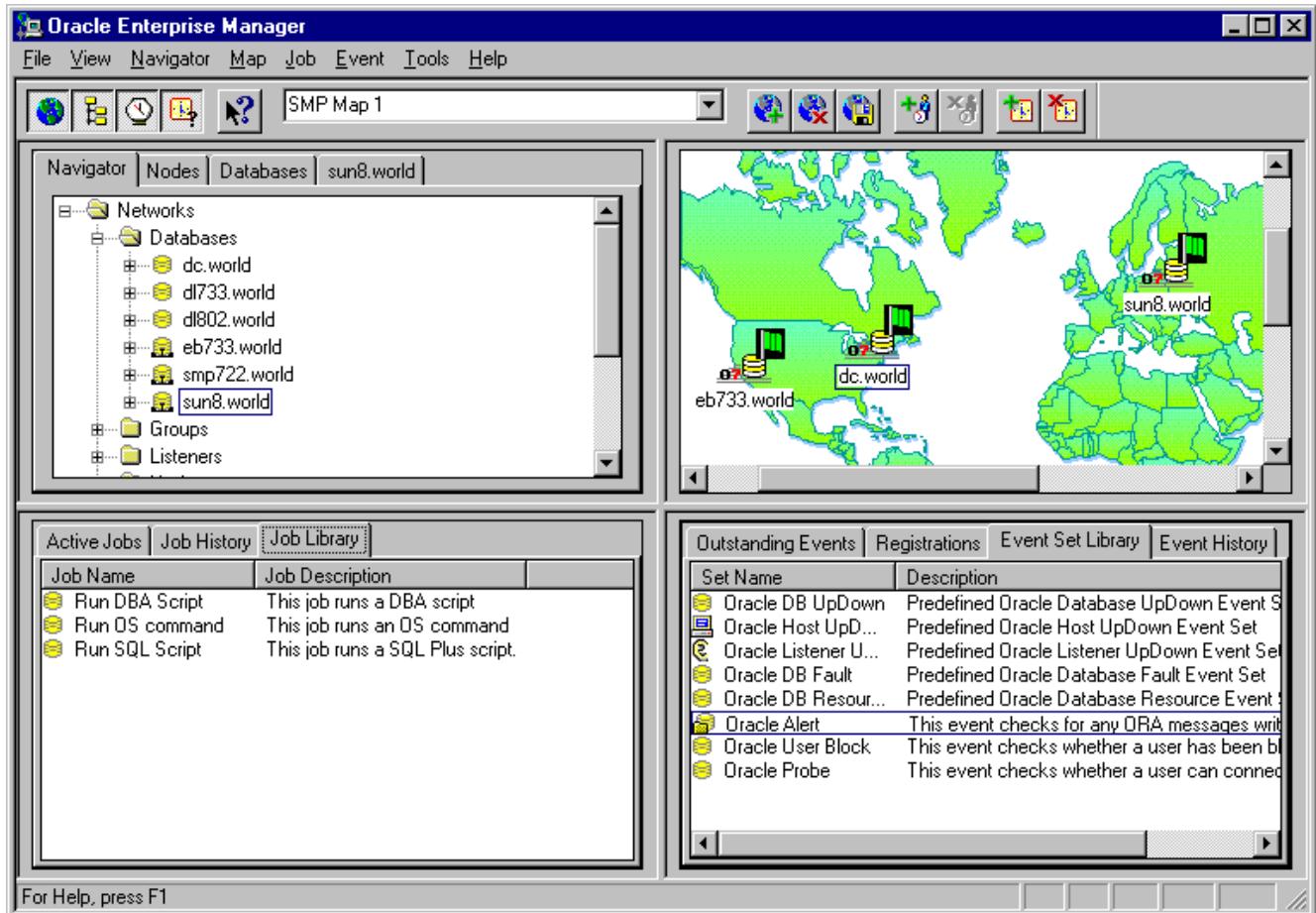
- Administrez, diagnostiquez et accordez des bases de données multiples.
- Distribuez le logiciel aux serveurs multiples et des clients.
- Emplois de programme sur noeuds multiples à intervalles temps variant.
- Le moniteur élève une objection et des événements partout dans le réseau.
- Personnalisez votre exposition employant des cartes multiples graphiques et les groupes d'objets de réseau, comme des noeuds et des bases de données.
- Administrez l'Oracle des Serveurs Parallèles. Pour l'information de l'administration de l'Oracle des Serveurs Parallèles, voir l'Oracle l'Appui de Serveur Parallèle pour le Manager d'Entreprise d'Oracle le Guide de Console.
- Intégrez l'Oracle participant et des outils de tiers.

II.2 Les composants d'Oracle Entreprise Manager :

L'Oracle Entreprise Manager consiste en composants multiples qui intègrent dans une interface d'utilisateur puissante et facile à utiliser graphiquement.

II.2.1 Console :

la Console d'Oracle Enterprise Manager est une interface d'utilisateur graphique qui fournit des menus, des barres d'outils, des palettes de lancement et la structure pour avoir accès aux outils d'Oracle et des utilités disponibles par d'autres vendeurs. Le format de la Console et les outils disponibles est déterminé par les produits achetés et des préférences d'utilisateur.



Les menus de console, des barres d'outils et des palettes d'outil fournissent l'accès aux composants de Console et l'administration de base de données (DBA) des demandes.

Les composants de la Console incluent :

- Navigateur
- Carte
- Planification de Travail
- Gestion d'Événement
- Dépôt

- Démon de Communication
- Cachette de Découverte

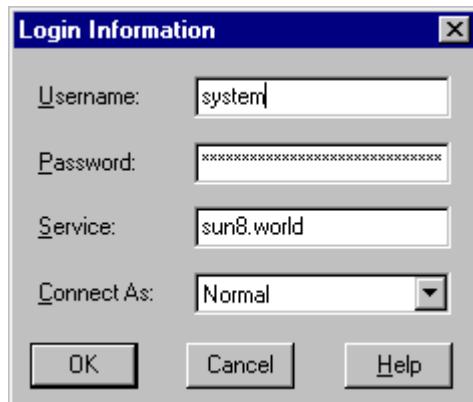
III.2.2 Agents Intelligents :

Les agents sont des processus intelligents courant sur des noeuds éloignés dans le réseau.

Oracle Entreprise Manager emploie des Agents Intelligents pour diriger des emplois et des événements de moniteur sur des sites éloignés. L'Agent Intelligent peut aussi être employé pour découvrir des services du noeud où il réside. Le Démon de Communication de Console communique avec les Agents Intelligents sur les noeuds éloignés dans le système.

III..3 lancement d'Oracle Entreprise Manager :

Lorsqu'on lance Oracle Entreprise Manager à partir de menu Démarrer>Programmes>Oracle Entreprise Manager>Enterprise Manager, il y l'apparition de la fenêtre de dialogue suivante ensuite vous pouvez se connecter à n'importe quel base de données d'Oracle.



. Les éléments de la boîte de dialogue sont décrits ci-dessous :

- Username : Votre Oracle username pour la base de données à lequel vous vous connectez.
- Password : Votre mot de passe d'Oracle pour la base de données à lequel vous voulez se connecter
- Service : Le nom de service de réseau.
- Connect As : NORMAL, pour SYSOPER et SYSDBA

III. Les composants SNMP d'Oracle au niveau de l'agent :

III.1 Agent de maître :

L'agent de maître est le processus sur un nœud géré qui accepte des requêtes envoyées par la station d'administration et communique avec les éléments à être gérés pour répondre à ces requêtes. Il peut aussi envoyer des traps SNMP indépendamment en réponse aux conditions spécifiques. Sur un nœud géré il existe un seul agent de maître, n'importe quel nœud qui ne présente pas d'agent de maître n'est pas capable de répondre aux demandes SNMP.

L'agent de maître peut être monolithique ou bien extensible. Si c'est monolithique, il communique directement avec les éléments à être géré. Bien qu'un tel agent puisse gérer des éléments multiples sur le même nœud, le jeu d'éléments qu'il peut se débrouiller est fixé quand l'agent est créé, parce que l'agent monolithique lui-même est responsable de l'interaffrontement aux éléments gérés.

Si, d'autre part l'agent de maître est extensible, il emploiera un sous-agent spécifique pour chaque élément qui doit gérer. Le sous-agent est responsable à l'interaffrontement à l'élément. Dans ce cas de nouveaux sous-agents peuvent se faire inscrire avec l'agent de maître à tout moment, donc de nouveaux éléments gérés peuvent être ajoutés dynamiquement.

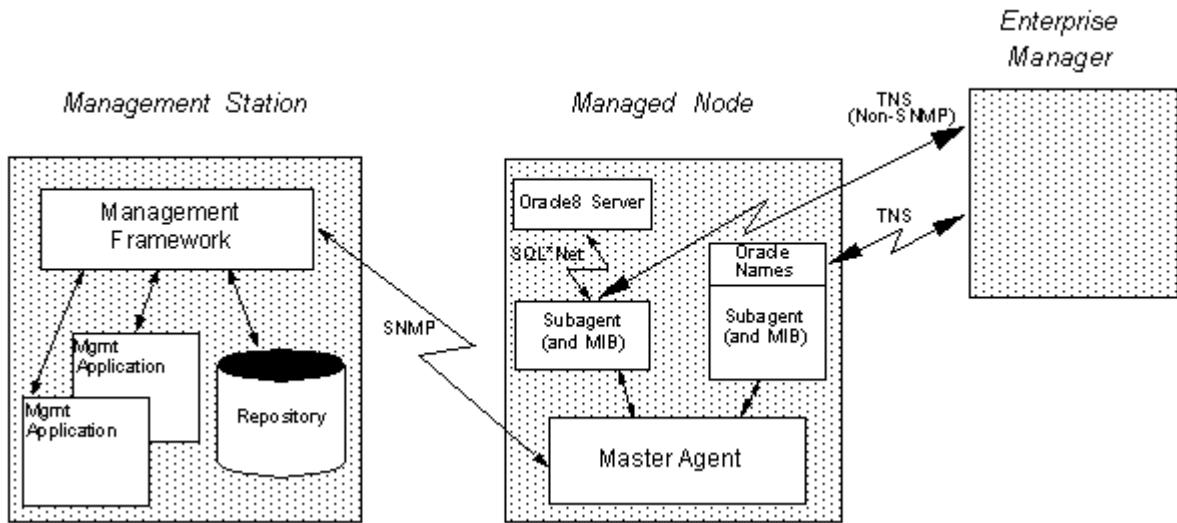
Quelques systèmes d'exploitation fournissent seulement des agents monolithiques, dans ce cas Oracle pourvoit un agent de maître qui peut se traiter à cet agent monolithique comme un sous-agent.

III.2 Sous-agent :

Le sous-agent est processus qui reçoit des questions pour un élément particulier géré de l'agent de maître et envoie en arrière les réponses appropriées à l'agent de maître. Un sous agent existe pour chaque élément géré résidant sur le nœud géré (à l'exception du fait qu'un sous-agent simple peut manipuler des cas de base de données d'Oracle multiples sur le même nœud). Dans la figure suivante, un sous-agent est consacré à l'Oracle7 ou la demande de serveur Oracle8 et un autre sous-agent est consacré à la demande de noms. Le sous-agent (s) et l'agent de maître communiquent en utilisant un protocole de multiplexage dicté par l'agent de maître. Il n'y a aucun protocole standard pour ce rapport et, tandis que quelques protocoles sont largement employés, aucun n'est une norme désignée.

Remarquez que le sous-agent pour l'Oracle7 et des serveurs Oracle8 est un processus séparé qui communique avec le serveur par SQL*NET (Net8 l'utilisation du protocole IPC). L'Oracle

nomme le sous-agent, d'autre part, est enfoncé dans le logiciel d'application lui-même. Toutes les deux de ces approches sont acceptables, comme le spécifique signifie que l'utilisation de sous-agent pour extraire des valeurs de SNMP est opaque à l'agent et à la structure.



IV Architecture de MIB SNMP d'Oracle :

IV.1 Interprétation de l'identificateur d'Oracle OID :

Un OID est un ordre des éléments qui indiquent une organisation hiérarchique d'identificateurs, ces éléments prennent la forme d'une série d'entiers pointillés.

On prend un exemple : 1.3.6.1.4.1.111.4.1.7.1.1

Dans cet OID :

- Le premier élément (1) se réfère à l'objet d'ISO.
- Le deuxième élément (3) se réfère à l'objet d'org.
- Le troisième élément (6) se réfère à l'objet dod.
- Le quatrième élément (1) se réfère à l'objet d'Internet.
- Le cinquième élément (4) se réfère à l'objet privé.
- Le sixième élément (1) se réfère à l'objet d'entreprise.
- Le septième élément (111) se réfère à l'objet d'Oracle.

Les variables dont leur OID commence par 1.3.6.1.4.1.111 se réfère à l'objet Oracle. L'objet Oracle a définit des espaces privées pour lui même ainsi :

- Le huitième élément (4) se réfère à l'identificateur de service de MIB.
 - (4) : Oracle base de données privée MIB.
 - (5) : Auditeur d'Oracle MIB.
 - (6) : Oracle nom MIB.
 - (7) : Echange de multiprotocole d'Oracle MIB.
- Le neuvième élément (1) est l'identificateur d'objet de MIB.
 - (1) : Il s'agit d'une variable SNMP.
 - (2) : Ils agit d'une trap SNMP.
- Le dixième élément (7) est l'identificateur de table MIB.
- L'onzième élément (1) est l'identificateur d'entrée de table (il est toujours à 1).
- Le douzième élément (1) est l'identificateur d'objet de feuille, il identifie une variable particulière dans la table.

III.2 Format d'une variable MIB :

- ❖ Nom de Variable :
 - **Syntaxe** : Il peut avoir l'une des états suivante { counter, integer, date/time, display string, ...}.
 - **Max-Access** : Il peut avoir l'une des états suivante {red-only, not accessible}.
 - **Status** : Il peut avoir comme état « mandatory » et ce champ n'est pas toujours présent.
 - **Explanation** : Ce champ contient l'explication de la variable, c'est-à-dire la fonction qu'elle assure.
 - **Typical range** : Décrit le typique, plutôt que théorique de la gamme de la variable MIB.
 - **Significance** : Il peut avoir l'une des états suivante {important, less important, very important not important }.
 - **Related variables** : Il contient les noms des variables qui peuvent être relier à cette variable.

- **Suggested presentation :** Il peut avoir l'une des états suivante {counter, not applicable, simple string, clock, integer, alarm, date/time, ...}.

Remarque : certains champ comme Status ne sont pas indiqué dans tous les variables MIB.

III.3 Interprétation des Tables des variables MIB Oracle :

Les variables MIB Oracle sont classés suivant 6 catégories :

- Le variables MIB de base de données d'Oracle.
- Le variables MIB d'Oracle auditeur « listner ».
- Le variables MIB de nom d'Oracle « names ».
- Le variables MIB de SGBDR public « RDBMS public ».
- Le variables MIBde mise en œuvre de service réseau.
- Le variables MIB d'entreprise manager.

III.3.1 les variables MIB de base de données Oracle :

Cette classe contient 11 tables de variables chaque table correspond à une signification et des propres variables.

- ❖ **OraDbstable** : c'est une table de 25 variables, elle a l'habitude de soutenir les activités de base de données de sessions actuelles sur le nœud géré. Cette information a été trouvée en particulier pour être utile pour le contrôle de l'exécution de cas de base de données globales.
- ❖ **OraDbTablespaceTable** : c'est une table de 6 variables, elle contient l'information sur Tablespace dans un cas de base de données d'Oracle. un Tablespace est une partie logique d'une base de données d'Oracle qui a l'habitude d'aller de l'espace mémoire pour le stockage des données d'index de la Table. Dans un environnement de production, les tables peuvent se remplir comme des supports d'activité de traduction. Le contrôle de Tablespace est important pour éviter les épuiser et cause des espaces.
- ❖ **OraDbDatafileTable** : c'est une table de 9 variables, elle contient l'information sur les fichiers de données pour la base de données actuel sur le nœud géré. Un fichier de données dénote un secteur de disque allouée pour des données de base. Le contrôle de fichier de données est important pour deux raisons, d'abord pour déterminer si l'espace dans les fichiers est épuisé et la seconde pour déterminer le temps de réponse dans l'accès aux données sur le disque.
- ❖ **OraDbLibraryCacheTable** : c'est une table de 8 variables, elle contient l'activité de cachette de bibliothèque de mesure de variables depuis le démarrage de cas de base de données le plus

récent. Une cachette de bibliothèque est une structure de données contenant le SQL partagé et les secteurs PL/SQL. Le contrôle de la cachette de bibliothèque est important de déterminer s'il est nécessaire de redimensionner l'association partagée de l'Oracle.

- ❖ **OraDbLibraryCacheSumTable** : c'est une table de 6 variables, elle contient les traces de TableLibraryCacheSum pour toutes les cachettes de bibliothèque dans un cas de base de données dans l'ensemble de même information que les traces de TableLibraryCache pour les cachette individuellement. c'est principalement une convenance.
- ❖ **OraDbSGATable** : c'est une table de 4 variables, elle contient l'information d'assignation de stockage sur le système actuel (le secteur global SGA). Le SGA est un secteur de stockage de base de données commun pour l'information qui soutient des sessions actuelles. Comme les valeurs de ses attributs sont configurés dans le fichier Init.ora, ils ne changent pas tandis que la base se données est déjà démarrée, ainsi, le SGA à ce niveau global n'est pas aussi important de contrôler que d'autre structure de base de données .
- ❖ **OraDbConfigTable** : c'est une table de 31 variables, elle contient des variables dont les valeurs sont recouvrés directement de la V\$ PARAMETER. Ces paramètres contrôlent des ressources systèmes et peut l'impact les valeurs de variables dans le reste de MIB. Cette variable inclut ces paramètres d'initialisation qui touchent près à la tache de contrôle l'exécution de cas de base de données (performance de la base).
- ❖ **OraRepTable** : c'est une table de 4 variables, elle contient l'information générale des schémas reproduits dans la base de données actuelle. En plus d'un compte du numéro de schémas étant reproduits aux sites éloignés, il annonce des problème avec le nœud actuel. Ces problèmes incluent l'incapacité de propager ou des données reproduites.
- ❖ **OraRepSchemaTable** :c'est une table de 5 variables, elle contient l'information des schémas de base de données reproduites au site actuels. Pour chaque tel schéma reproduit, existe là un cas de cette table. Les valeurs de ces variables sont recouvrées directement du catalogue de reproduction, le l'environnement de reproduction actuel et la configuration sont utile pour l'exposition d'une représentation visuelle de l'environnement de reproduction.
- ❖ **OraRepMasterSchemaTable** : c'est une table de 3 variables, elle contient l'information spécifique pour des schémas reproduits au maître actuel ou le site de définition de maître. Les valeurs de cette table variable sont utiles pour l'exposition d'une représentation visuelle de configuration de maître multi. Elles fournissent aussi l'information pour la détection d'erreurs de demande d'administrateur de reproduction et un compte relatif du numéro de conflits rencontrés au site actuel.

- ❖ **OraRepLinkTable** : c'est une table de 5 variables, elle contient l'information pour chaque liaison de base de données employées. Les demandes d'administrateur sont seulement faites des site de définitions de maître aux sites de maître éloigné.

III.3.2. Les variables MIB d'Oracle Listener :

- ❖ **OraListenerTable** : c'est une table de 13 variables, elle contient l'information des auditeurs de réseau génériques installés sur un nœud géré. Un auditeur de réseau est un éditeur de serveur qui écoute pour les demandes de connexions d'un ou de plusieurs protocoles de réseau. Le contrôle de l'auditeur est important pour l'accès à la base de données.
- ❖ **OraSIDTable** : c'est une table de 4 variables, elle contient ou fournit l'information de Status des connexions à un cas de base de données, cette information est importante à contrôler pour s'assurer que les clients qui cherchent l'accès à une base de données ont cette occasion.
- ❖ **OraDedicatedSrvTable** : c'est une table de 3 variables, elle contient l'information des serveurs consacrés pourvus par l'auditeur géré. Un dispatcher est un processus de fonctionnement facultatif se présente seulement quand une configuration de serveur multi chaînée est employée. Le contrôle de dispatcher est obligatoire pour le même raison que précédemment.
- ❖ **OraPrespaunedSrvTable** : c'est une table de 8 variables, elle contient l'information sur les serveurs prespawned pourvus par l'auditeur géré associé au cas de base de données. Un serveur prespawned est un processus créé quand l'auditeur a commencé et ensuite disponible pour la fabrication des connexions à la base de données. Perspawned sont des serveurs qui réduisent le temps de connexions en éliminant le besoin de frayer une ombre pour chaque nouvelle connexion demandée.
- ❖ **OraListnerAdrssTable** : c'est une table de 2 variables, elle contient les adresses sur lesquelles l'auditeur écoute. Ces adresses sont importante à contrôler quand un auditeur ne peut pas aborder pendant la démarrage, ces adresses peuvent alors être examinées en format approprié.

III.3.3 Les tables des variables MIB Oracle Name :

- ❖ **OraNamesTNSTable** : c'est une table de 7 variables, elle contient les variables qui pourraient être communes à toutes les demandes courant de SQL*Net. SQL*Net est le logiciel d'accès de

données éloigné de l'oracle qui permet des communications client/serveur, serveur/serveur à tarer quel réseau.

- ❖ **OraConfigTable** : c'est une table de 29 variables, elle contient l'information sur cet OracleNomme, les paramètres de configuration de serveur qui ne sont pas inclus dans l'OraNamesTNSTable. le contrôle de cet opération est important pour caractériser l'opération de l'OracleNomme pour le serveur.
- ❖ **OraNamesServerTable** : c'est une table de 30 variables, elle contient l'information sur les variables qui décrivent le statut actuel opérationnel de l'Oracle Nomme de serveur. Le contrôle de ces mesures est important pour s'assurer que le serveur est entièrement fonctionnel.

III.3.4 les tables des variables de MIB RDBMS public :

- ❖ **RdbmsDBTable** : c'est une table de 5 variables, elle contient l'information générale sur chaque base de données installée sur le nœud géré. N'importe quel base de données installée qui a été configurée l'appui de SNMP est activement ouverte.
- ❖ **RdbmsDBinfoTable** : c'est une table de 5 variables, elle contient l'information complémentaire générale de cas de base de données activement ouverts sur le nœud géré. Si une base de données a été configurée pour SNMP soutiennent mais n'est pas activement ouverte, donc des tentatives d'avoir accès aux cas correspondants dans cette table peuvent aboutir ou bien à 'no Such Name' pour SNMPv1 ou bien 'no Such instance' pour SNMPv2.
- ❖ **RdbmsSrvTable** : c'est une table de 4 variables, elle contient l'information générale sur chaque serveur de base de données installé sur le nœud géré. Dans la configuration d'Oracle Standard, un serveur soutient une base de données. Ainsi, le contenu de RdbmsSrvTable est presque le même que le contenu de RdbmsDbTable, oracle à voulu mettre en œuvre toutes les deux tables pour adhérer à un des préceptes centraux de publique RDBMS MIB. Le précepte est une distinction entre la base de données et son serveur qui doit être faite pour représenter l'architecture d'autre vendeurs.
- ❖ **RdbmsSrvinfoTable** : c'est une table de 11 variables, elle contient l'information complémentaire de chaque cas de serveur de données courant activement sur le nœud géré. La visibilité d'un cas de serveur de base de données dans cette table est déterminer par la valeur de la transmission 'Appel Oper Status' trouver dans la table 'Appel Table'. Le contrôle de ces

variables est important pour le réglage de d'accord de fonctions de serveur comme l'entrée_sortie, l'accès et l'activité.

- ❖ **RdbmsParamTable** : c'est une table de 3 variables, elle contient l'information ou les paramètres de configuration pour un serveur de base de données installé sur ce nœud géré. Cette table correspond presque à la table 'OraDbconfigTable'
- ❖ **RdbmsRelTable** : c'est une table d'une seule variable, elle indique comment les bases de données et les serveurs sur ce nœud géré sont rapprochés.
- ❖ **RdbmsTraps** : c'est une table d'une seule variable, a la différence d'autre variables SNMP qu'une variable de trap n'est pas la réponse d'un agent à une station d'administration mais plutôt c'est un message qu'un agent peut envoyer non sollicité à une station de gestion quand cet agent remarque que quelque condition anormale est arrivée (il s'agit d'une alerte quand un seuil a été atteint).

III.3.5 Les variables MIB de mise en œuvre de service réseau :

C'est une table appeler 'Appel Table' de 11 variables, elle contient les informations sur les variables génériques conçus pour s'appliquer à tous les types des applications de service réseau. Certains variables de 'Appel Table' ont été mise en œuvre pour les trois services d'oracle :l'Oracle7, serveur de base de données Oracle8 et Noms d'oracle.

III.3.6 Les variables MIB de Manager d'Entreprise :

Il y a deux tables qui sont :

- ❖ **OraAgentEventTable**, qui est une table de 11 variables.
- ❖ **OraAgentTraps**, qui est une table d'une seule variable.
- ❖ **OraAgentTraps** : comme il est précédemment, une variable de trap n'est pas la réponse d'un agent à une station de gestion.

Tous les événement découvertes par le Manager d'Entreprise d'Oracle sont envoyés comme Traps SNMP à condition que l'agent sera configuré pour les envoyés.

Les traps envoyés par le Manager d'Entreprise d'oracle sont reçus par n'importe quel outil d'écoute sur le port 162 et capable d'exécuter le décodage d'ASN.1 exigé.

L'agent essaye de se connecter à l'agent de maître SNMP pendant le temps de démarrage, si ce la échoue ou bien parce que la plate-forme ne soutient pas SNMP ou bien l'agent de maître n'est pas commencé, donc aucun trap n'est envoyée.

La destination de tous les trap est déterminée par la question de configuration de l'agent de maître SNMP.

La table OraAgentEventTable contient 11 variables définie comme suit :

- **OraAgentEventIndex** : index de la table .
- **OraAgentEventName** : le nom d'événement qui est arriver.
- **OraAgentEventID** : l'identificateur ID de l'enregistrement qui a produit la présence de cette événement.
- **OraAgentEventService** : le nom de service (base de données, SQL*Net, ...) étant contrôlé par cet événement.
- **OraAgentEventTime** : le temps de détection de l'événement.
- **OraAgentEventSeverity** : une mesure relative à la sévérité de l'événement.
- **OraAgentEventUser** : le nom de l'utilisateur qui a enregistré cet événement.
- **OraAgentEventAppID** : l'identificateur de console 'Manager Entreprise' par laquelle cet événement à été enregistré.
- **OraAgentEventMessage** : message associé avec l'événement.
- **OraAgentEventArguments** :les arguments qui ont détecté l'événement.
- **OraAgentEventResult** :le résultat généré par un script lors de détection de l'événement.

V. Conclusion :

Dans ce chapitre nous avons présenté Oracle Enterprise Manager qui est un outil d'administration permettant grâce à son interface graphique une gestion des produits Oracles, en employant des cartes multiples, graphiques et les groupes d'objets de réseau, comme des noeuds et des bases de données.

Nous avons aussi interprété les variables MIB privé d'Oracle en indiquant par un exemple l'organisation des objets d'un OID et l'affectation de leurs numéros. Ces variables sont structurés dans des tables qui sont présenté suivant les six classes. Ces variables seront utiles pour un gestionnaire de réseau autre que l'outil d'administration d'Oracle qui ne permet ni l'extraction de ces variables MIB ni l'attrape des messages alertes (TRAP).

CHAPITRE 5

CONFIGURATION DE L'AGENT SNMP D'ORACLE ET INTERFACAGE AVEC UN GESTIONNAIRE DE RESEAU

I. Introduction :

Pour aboutir à l'interprétation des différentes variables MIB d'Oracle et les mettre en questions, ou bien pour recevoir les traps envoyées suite à une alerte, on doit assurer des précautions de communication entre le poste d'administration et l'agent SNMP d'Oracle. L'extraction d'une variable MIB ne sera effectuée qu'après la configuration de notre agent SNMP d'Oracle pour écouter le réseau et encapsule la réponse. La réception des traps ne sera détectée qu'après l'interfaçage d'un gestionnaire de réseau. Dans ce chapitre nous allons présentés les fichiers de configurations pour l'accès à la base de données et les fichiers qui assurent son administration, ensuite on va installer un gestionnaire de réseau sur un poste administrateur pour capter les traps émises. En fin on interprète les résultats obtenus.

II Le Listener Oracle :

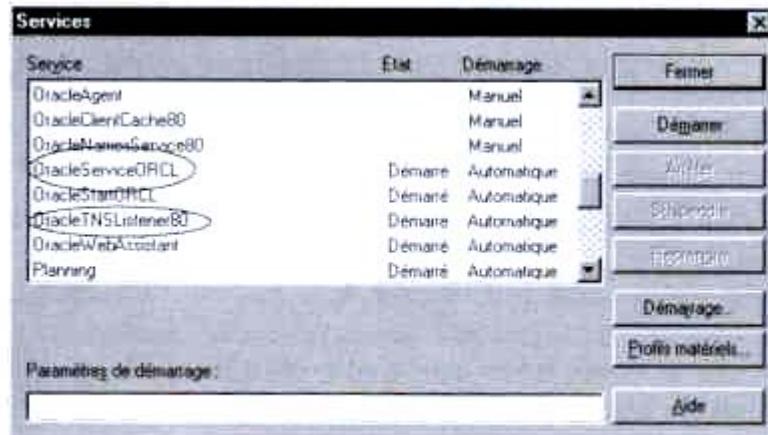
Les paramètres regroupés sous un alias significatif sont ceux qui sont envoyés à la machine cible. Celle-ci doit être à l'écoute des demandes de connexion. C'est le rôle de Listener Oracle. Ce Listener qui a la charge d'écouter doit être paramétré et lancé.

Il doit y avoir un Listener par serveur. Un Listener unique permet de se connecter à toutes les bases présentes sur le serveur. Il faudra modifier le fichier de configuration de Listener présent sur votre serveur et ajouter de nouveaux alias sur vos clients lorsque vous créez de nouvelles bases.

Le démarrage et l'arrêt du Listener sont gérés directement au niveau des services de Windows NT.

II.1. Configuration des alias :

Une nouvelle fois, nous allons nous appuyer sur l'instance ORCL, créée par défaut lors de l'installation d'Oracle8. Assurez-vous que la base ORCL et votre Listener sont lancés au niveau des services NT.

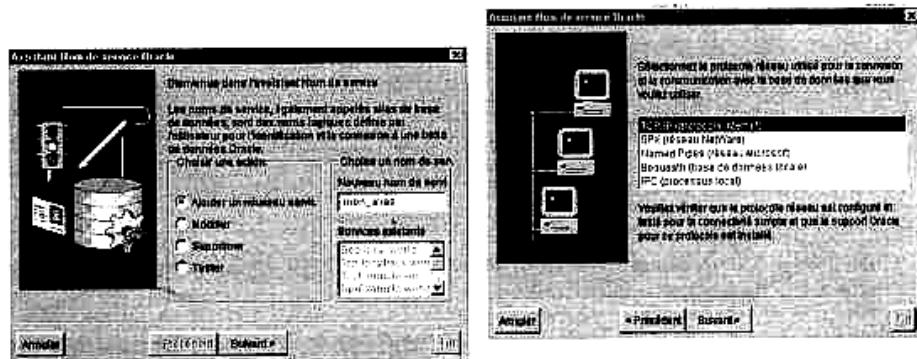


Note : Si vous avez des bases Oracle7 et Oracle8 sur le même serveur, seul le Listener Net8 doit être configuré et démarré pour gérer les deux versions de bases.

Pour paramétriser, créer et modifier les alias, nous avons le choix entre deux utilitaires accessibles à partir de : Démarrer>Programme>Oracle pour Windows NT :

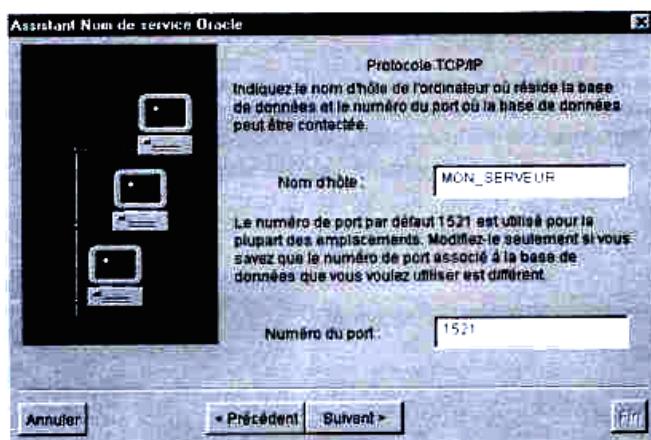
- Oracle Net8 Assistant (le plus complet),
- Oracle Net8 Easy Config.

Lancez ORACLE Net8 Easy Config. Pour ajouter un nouvel alias, donnez le nom que vous souhaitez à la place de « mon_alias », puis cliquez sur « suivant » et choisissez TCP/ IP.

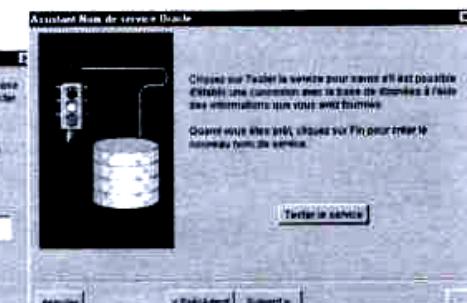
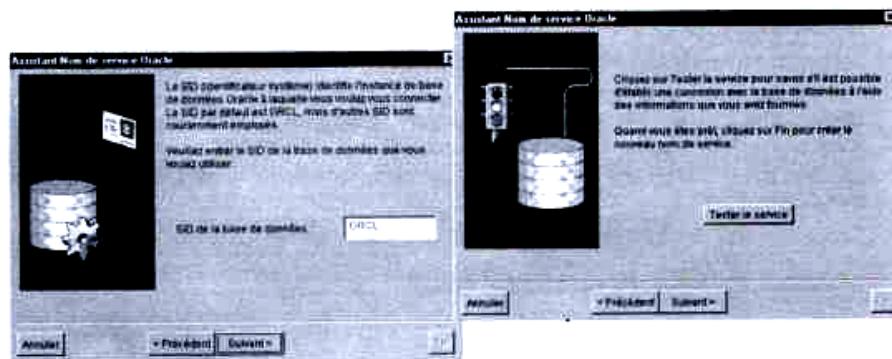


Donnez ensuite le nom de votre serveur (ce nom figure dans le paramétrage du réseau de votre serveur). Le numéro de port concerne un paramètre de TCP/ IP.

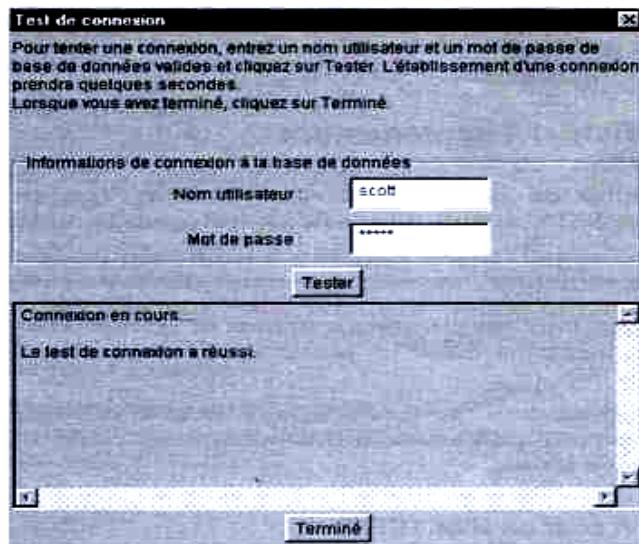
(Ce point est détaillé plus loin lors de l'étude des fichier de configuration). Spécifiez toujours 1526ou 1521 pour plus de simplicité.



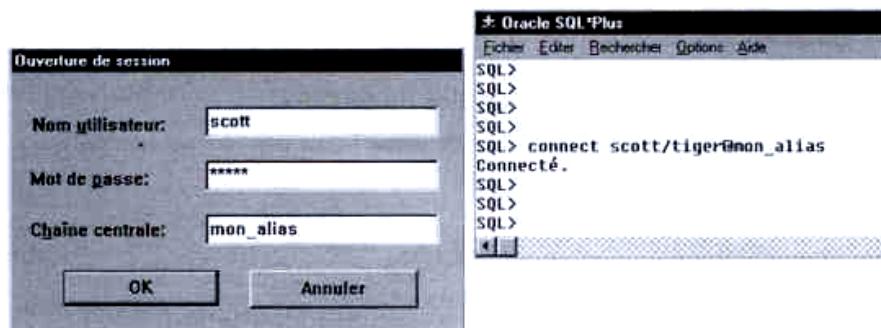
Précisez le SID de la base à laquelle vous souhaitez accéder. Une machine cible peut en effet accueillir plusieurs instances.



Il vous reste qu'à tester votre connexion avec l'utilisateur Scott (mot de passe Tiger) qui est créé par défaut dans l'instance de teste ORCL.



Maintenant, pour vous connecter, vous pouvez utiliser l'alias « mon_alias » créé précédemment. Par exemple, à partir de SQL*Plus. Nous vous précisons les deux syntaxes possibles.



L'assistant Oracle Net8 Easy Config agit sur le fichier Tnsnames.ora. C'est pourquoi nous allons en étudier la structure.

II.2 Etude du fichier Tnsnames.ora :

Le paramétrage de cet alias est stocké dans le fichier Tnsnames.ora situé en ORACLE_HOME\NET80\ADMIN, fichier texte que vous pouvez lire facilement.

Pour plus de clarté, nous vous présentons de l'exemple crée précédemment. Nous avons uniquement extrait les paramètres suivant. Les « # » indiquent des commentaires.

Pour qu'une configuration client-serveur fonctionne, il faut qu'un alias existe dans le fichier Tnsnames.ora, aussi que le serveur (le Listener Oracle) soit configuré pour écouter ce qui lui est envoyé par le client.

II.2.1 Configuration pour un alias TCP/IP :

```
mon_alias.world =                               # nom donné à mon alias
  (DESCRIPTION =
    (ADDRESS =
      (PROTOCOL = TCP)          # CTP/ IP
      (HOST = MON_SERVEUR)     # nom de serveur cible sous TCP/ IP
      (PORT = 1521)            # numéro de port IP
    )
    (CONNECT_DATA = (SID = ORCL) # SID de la base cible
    )
  )
}
```

l'extension « World » vient du fichier Sqlnet.ora que nous verrons par la suite.

II.2.2 Configuration pour un alias NetBios ou SPX/ IPX :

Voici les exemple standards provenant du même fichier pour des réseaux NetBios et SPX/ IPX :

```
NmpExample.world =                           # nom donné à mon alias
  (DESCRIPTION =
    (ADDRESS =
      (PROTOCOL = NMP)          # Name Pipes
      (Server = FinanceServer1) # nom NetBios du serveur cible
      (Pipe = ORAPIPE)         # canal écouté par le Listener
    )
    (CONNECT_DATA = (SID = ORCL) # SID de la base cible
    )
  )
)
SpxExample.world =                           # nom donné à mon alias
  (DESCRIPTION =
    (ADDRESS =
```

```

        (PROTOCOL = SPX          # SPX
         (Service = Server_lsnr) # nom de Listener (unique sous IPX)
         )
        (CONNECT_DATA =(SID = ORCL) # SID
         )
        )
    
```

SPS/ IPX a la particularité de proposer un nom unique pour tous les objets sur son réseau. Il n'est donc pas nécessaire de faire figurer le nom de la machine cible depuis le « noyau » Oracle, qui est retrouvée dès que l'on donne le nom de service.

II.3 Le fichier Sqtnet.ora :

Ce fichier est également placé en ORACLE_HOME\NET80\ADMIN. Il ne contient que des paramètres. Il est normalement inutile de le modifier.

```

TRACE_LEVEL_CLIENT = OFF
sqlnet.autentification_services = (NTS)
names_directory_path = (TNSNAMES , HOSTNAME)
names_default_domain = world
name_default_zone = off
automatic_ipc =off
    
```

vous constatez que nom de zone est « world », par défaut. Si vous le modifiez ici, il faudra répercuter la modification dans les fichiers Tnsnames.ora et Listener.ora.

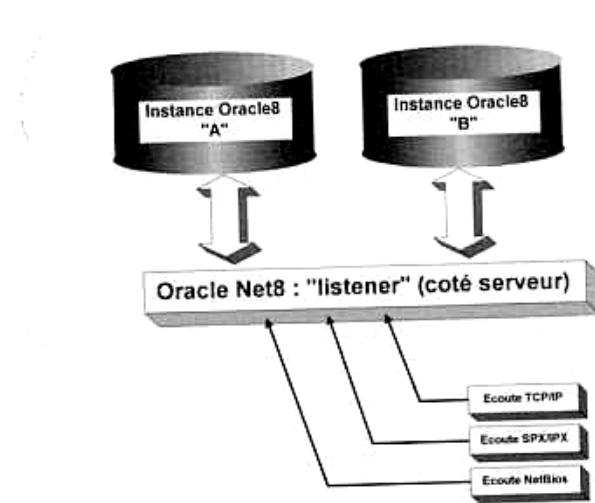
II.4 le fichier listener.ora :

Il permet de paramétriser le Listener. Oracle ne propose pas d'outils simples pour effectuer cette configuration.

Nous ne verrons que les paramètres à modifier dans le Listener.ora, lors de l'ajout d'une base sur un serveur.

Le schéma suivant en résume les deux grandes fonctions assurer par le Listener :

- écouter sur plusieurs réseau (TCP/IP, NetBios...),
- router les communication vers l'instance cible.



On retrouvera donc ces deux aspects dans le fichier Listener.ora. Une parti pour écouter les différents réseaux, l'autre pour router vers une instance ou l'autre. le contenu de fichier Listener.ora est construit dynamiquement en fonction des protocoles réseau présents sur votre machine lors de l'installation de Net8 Server. Cela explique pourquoi certaines sections du fichier Listener.ora ne seront pas présentes sur votre machine. Comme le Listener écoute successivement sur chaque « entrée » réseau, vous minimiserez son travail en installant uniquement les protocoles réseau strictement nécessaire, sur votre machine.

Pour accéder à une nouvelle base, nous avons fait figurer en gras les lignes à ajouter). Les deux paramètres à changer sont le nom du serveur et de l'instance écoutée par le Listener.
Il est indispensables d'arrêter puis de relancer le service NT pour en compte les modifications apportées au fichier Listener.ora.

```

#####
# Filename .....: Listener.ora
# Node .....: local.world
# Date .....: 24-MAY-95 13:25:30
#####
LISTENER =
(ADDRESS_LIST =
  (ADDRES =
    (PROTOCOL = IPC)
    (KEY = oracle.world)
  )
)
  
```

```

        (ADDRESS =
          (PROTOCOL =IPC)
          (KEY = ORCL)
        )
      (ADDRESS =
        (COMMUNITY = NMP.world)
        (PROTOCOL = NMP)
        (SERVER = ORAPIPE)           # écoute sur le pipe
      )
      (ADDRESS =
        (ADDRESS = TCP)
        (HOST = MON_SERVEUR)
        (PORT = 1521)               # écoute sur le port 1521
      )
      (ADDRESS =
        (PROTOCOL = TCP)
        (HOST = MON_SERVEUR)
        (PORT = 1526)               # écoute sur le port 1526
      )
      (ADDRESS =
        (PROTOCOL = TCP)
        (HOST = 127.0.0.1)
        (PORT =1521)
      )
      (ADDRESS =
        (PROTOCOL = SPX)
        (Service = mon_serveur_lsnr) # écoute pour SPX
      )
    )
  )
ATARTUP_WAIT_TIME_LISTENER = 0
CONNECT_TIMEOUT_LISTENER = 10
TRACE_LEVEL_LISTENER = 0
SID_LIST_LISTNER =
  (SID_LIST =
    (SID_DESC =
      (GLOBAL_DBNAME = MON_SERVEUR)
      (SID_NAME = ORBL)
    )
    (SID_DESC =
      (GLOBAL_DBNAME = MON_SERVEUR)
      (SID_NAME = MON_NOUVEAU_SID)
    )
    (SID_DESC =
      (SID_NAME = extroc)          # procédure externe
      (PROGRAM =extproc)          # programme exécutant la procédure externe
    )
  )
PASSWORD_LISTNER = (oracle)

```

II.5 Où placer les fichier Tnsnames, Sqlnet, Listener ?

Le fichier Listener.ora doit être présent uniquement sur le serveur pour lequel il a été conçu.

Les fichiers Tnsnames.ora et Sqlnet.ora sont utilisés par tous les clients. Vous devez en placer une copie sur chaque client, en ORACLE_HOME\NET80\ADMIN. Si votre serveur peut être « client » de lui-même ou d'un autre serveur, il doit disposer de ce fichier.

III Oracle SNMP Agent pour Services d'Oracle

L'Oracle SNMP l'Agent vous permet de configurer un

- OracleSNMPPeerMasterAgent
- OracleSNMPPeer Encapsulator

OracleSNMPPeerMasterAgent

L'OracleSNMPPeerMasterAgent est le processus sur un noeud géré qui accepte des requêtes de la station d'administration et communique avec les sous-agents pour répondre à ces requêtes. Il peut aussi envoyer des traps SNMP indépendamment en réponse aux conditions spécifiques, seulement lorsque un agent de maître peut exister sur chaque noeud géré. N'importe quel noeud qui ne possède pas d'agent ne peut pas être capable de répondre aux requêtes SNMP.

OracleSNMPPeerEncapsulator

Si on emploie actuellement un agent incompatible, comme Microsoft SNMP, on doit installer l'Encapsulator à encapsuler l'agent de maître de Microsoft tel que toutes les demandes de SNMP d'une Station de Gestion de Réseau (NMS) sont envoyées à OracleSNMPPeerMasterAgent, qu'alors en avant les demandes appropriées à l'agent de maître encapsulé par l'encapsulator. Si l'encapsulator n'est pas configuré, Microsoft SNMP ne travaillera pas.

III.1 Configuration d'Oracle SNMP Agent :

Cette section décrit brièvement les tâches principales que nous avons appliquée pour permettre l'exécution l'Agent SNMP d'Oracle. L'Oracle SNMP Agent est installé avec l'Oracle l'Installateur Universel.

Configurer l'Oracle SNMP soutient sur un noeud géré :

- Spécifiez le port où l'agent de maître écoute. Le port est spécifié dans la section de TRANSPORT du fichier MASTER.CFG placé à ORACLE_HOME\NET80\ADMIN\On ajoutez la section suivante au fichier :

TRANSPORTEZ SNMPOVER ordinaire UDP SOCKETAT PORT 161

Notez : la Société d'Oracle recommande au port d'utilisation 161 pour OracleSNMPPeerMasterAgent : puisque c'est le port de défaut pour la communication SNMP. Cependant, on peut spécifier un port différent tant que la demande de gestion sur notre NMS peut être configurée pour envoyer des demandes de SNMP à l'agent de maître écoutant sur ce port.

- Spécifiez l'authentification dans la section de COMMUNAUTÉ du fichier MASTER.CFG :

Public de COMMUNAUTÉ NE PERMET AUCUN CHIFFRAGE À TOUTE
L'UTILISATION D'OPÉRATIONS

Continuons vers l'étape3, l'Encapsulator doit être employé.

- Spécifiez un port inutilisé ou le service Microsoft SNMP (l'agent encapsulé) doit écouter.

Le port est spécifié dans le fichier de SERVICES placé à NT_HOME\SYSTEM32\DRIVERS\ETC. On assure l'existence de la ligne suivante dans le fichier services :

Snmap 1161/udp snmp

Notez : s'il y a déjà eu une entrée pour SNMP dans le fichier, le changement le port de 161 (le numéro{*nombre*} de défaut) à quelque chose autrement qui est disponible (1161 dans cet exemple).

- La quatrième étape consiste à l'édition du fichier de configuration, ENCAPS.CFG, placé à ORACLE_HOME\NET80\ADMIN Vous devez au moins ajouter une entrée d'AGENT, incluant MIB des sous-arbres maniables par NMS, pour l'agent de maître encapsulé.

On doit avoir une section comme la suivante dans le fichier :

AGENT À PORT 1161 AVEC COMMUNAUTÉ public
SUBTREES
1.3.6.1.2.1.1,
1.3.6.1.2.1.2,
1.3.6.1.2.1.3,

```
1.3.6.1.2.1.4,  
1.3.6.1.2.1.5,  
1.3.6.1.2.1.6,  
1.3.6.1.2.1.7,  
1.3.6.1.2.1.8,  
1.3.6.1.2.1.77  
FORWARD TOUS LES PIÈGES;
```

Notez : le port (1161 dans cet exemple) doit correspondre à celui que nous avons spécifié dans la troisième étape.

III.2 Direction de l'Agent de Maître et l'Encapsulator :

III.2.1 Départ de l'Agent de Maître :

Nous pouvons lancer l'agent de maître par une commande de MS-DOS ou de la boîte de dialogue de Services du Panneau de configuration de Windows NT. Toutes les deux voies sont décrites ci-dessous.

Lancer l'agent de maître par une commande de MS-DOS :

Invoquez l'agent de maître comme suit :

C:\> CD ORACLE_HOME\BIN

```
C:\> AGENT CONFIGURATION_FILE TEMPORARY_FILE
```

Où CONFIGURATION_FILE est le nom du fichier de configuration de maître et TEMPORARY_FILE est un fichier provisoire où l'information de stockage doit être stockée. Le nom de fichier de configuration de maître est manqué à ses engagements à MASTER.CFG.

Notez : CONFIGURATION_FILE et TEMPORARY_FILE sont obligatoire.

Lancer l'agent de maître De la fenêtre de Panneau de configuration :

double cliquez sur l'icône Services.

La fenêtre de Services apparaît.

Cherchez le service d'OracleSNMPPeerMasterAgent.

Un blanc dans la colonne de Statut indique que le service s'est arrêté.

Si le maître court, on saute les deux premières étapes, si le maître ne dirige pas, continuons à L'étape 2.

Choisissez le service d'OracleSNMPPeerMasterAgent et le Début de clic.

Les démarrages d'agent de maître.

Dans la fenêtre de Services, choisissez le bouton Proche.

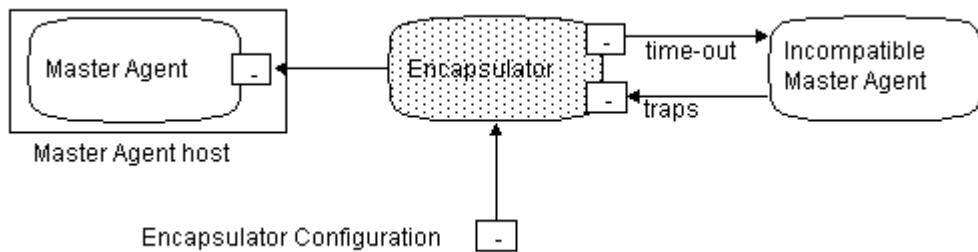
III.2.2 Départ de l'Encapsulator :

Le principe de lancement l'encapsulator est le même que pour le fichier MASTER.CFG par la ligne de commande de MS-DOS ou de la boîte de dialogue de Services du Panneau de configuration de Windows NT. Toutes les deux voies sont décrites ci-dessous.

Notez : si l'Encapsulator doit être employé, l'agent de maître doit être commencé d'abord. Commençons l'encapsulator du commandes MS-DOS :

```
C:\ > CD ORACLE_HOME\BIN
```

```
C:\ > ENCAPS [options]
```



Il y a six options de ligne de commande comme montré ci-dessous. Les options sont :

- T le PORT Indique le port de trap entrant où l'encapsulator écoute pour recevoir des traps de SNMP envoyés par des agents de maître encapsulés. Défaut : aucun trap expédiés.
- S le PORT Indique le port dont l'encapsulator envoie des requêtes SNMP aux agents encapsulés et auquel les agents encapsulés envoient leurs réponses. Défaut : assigné par l'hôte
- H l'HÔTE Indique l'adresse d'hôte de l'agent de maître, c'est-à-dire l'hôte où l'agent de maître réside. Cela peut ou ne peut pas être l'hôte de l'agent encapsulé, indiqué dans le fichier ENCAPS.CFG. Défaut : localhost
- P le PORT Indique l'écoute de l'agent de maître SMUX le port, c'est-à-dire le port auquel l'encapsulator envoie son trafic SMUX. Défaut : 199
- W ATTENDENT Indique le temps mort de réponse SNMP (dans des secondes) de l'agent de maître encapsulé. Défaut : 2 s

- C le FICHIER Indique le fichier de configuration de l'encapsulator's. Ce fichier est manqué à ses engagements à ENCAPS.CFG.

Commencer l'encapsulator de la Fenêtre de Panneau de configuration :

De la fenêtre de Panneau de configuration, double cliquez sur l'icône Services.

La fenêtre de Services apparaît. Cherchez le service d'OracleSNMPPeerEncapsulator. Un blanc dans la colonne de Statut indique que le service est arrêté. Si le maître court, sauter pour l'étape 3. Si le maître ne dirige pas, continuer à l'étape 2.

Choisissez le service d'OracleSNMPPeerEncapsulator et le Début de clic.

L'encapsulator est lancé.

Dans la fenêtre de Services, choisissez le bouton Proche.

III.3 Compréhension du Fichier MASTER.CFG :

Le fichier MASTER.CFG contrôle plusieurs aspects de l'Agent OracleSNMPPeerMaster, incluant :

- Choix de protocoles de transport
- Contrôle d'accès basé sur communauté
- Nom basé sur communauté

Notez : le fichier MASTER.CFG est automatiquement installé sur le serveur à
ORACLE_HOME\NET80\ADMIN

```
#####
# Default Agent Configuration File
#
# This file allows MANAGERS to be specified. This is used to
# specify which managers will be receiving which traps.
#
# Also, COMMUNITYs can be specified. This allows that agent to
# be configured such that it will only accept requests from
# certain managers and with certain community strings.
#
# Syntax
# TRANSPORT <name> SNMP
#           [OVER UDP SOCKET]
#           [AT <addr>]
# COMMUNITY <communityName>
#           ALLOW <op> [<op>]* [OPERATIONS]
#           [AS ENTITY <entityName>]
#           [MEMBERS <addrs> [,<addrs>] ]
# TRANSPORT ordinary SNMP
```

```

OVER UDP SOCKET
AT PORT 161
COMMUNITY public
ALLOW ALL OPERATIONS
USE NO ENCRYPTION

```

Ci-dessous est une description de PARAMÈTRES de communauté et le TRANSPORT.

Le TRANSPORT : Définit une interface sur laquelle l'agent de maître écoute pour des rapports SMUX de demandes de SNMP ou des SOUS-AGENTS.

La COMMUNAUTÉ : Fournit une forme d'authentification pour l'accès à l'information.

Notez : Ce paramètre n'est pas le même comme le paramètre de COMMUNAUTÉ dans le fichier TNSNAMES.ORA.

III.4 Compréhension du Fichier ENCAPS.CFG :

Le fichier de configuration ENCAPS.CFG indique que des agents incompatibles doivent être encapsulés et que faire visible à la Station de Gestion de Réseau (NMS).

On montre un échantillon ENCAPS.CFG le fichier de configuration dans la figure ci-dessous :

```

#####
#
# THIS IS AN EXAMPLE OF AN ENCAPSULATOR CONFIGURATION FILE.
#
# USING THIS FILE, ENCAPSULATOR WILL ENCAPSULATE A NON-PEER
# MIB2 AGENT ON THE SAME PROCESSOR, THAT'S LISTENING FOR SNMP
# REQUESTS AT PORT 1161.
#
# IT ALLOWS THE NETWORK MANAGEMENT STATION TO VIEW THE 8 SUBTREES
# MANAGED BY THAT AGENT, AND TO RECEIVE ALL THE TRAPS EMITTED BY
# THAT AGENT. ALL THE REQUESTS AND TRAPS PASS THROUGH THE PEER
# AGENT AND ENCAPSULATOR. AT THE SAME TIME THAT THE PEER AGENT
# IS HANDLING ENCAPSULATOR, IT ALSO HANDLES ALL SUB-AGENTS THAT
# WERE DEVELOPED WITH THE PEER TOOLKIT TO MANAGE OTHER MIBS.
#
# Syntax:
# [
# AGENT [ON HOST <ip>] [AT PORT <port>] [WITH COMMUNITY <community>]
# SUBTREES <treetlist>
# [FORWARD <traplist> TRAPS]
# ;
# ]+
#
AGENT AT PORT 1161 WITH COMMUNITY public

```

SUBTREES

```
1.3.6.1.2.1.1,  
1.3.6.1.2.1.2,  
1.3.6.1.2.1.3,  
1.3.6.1.2.1.4,  
1.3.6.1.2.1.5,  
1.3.6.1.2.1.6,  
1.3.6.1.2.1.7,  
1.3.6.1.2.1.8,  
1.3.6.1.4.1.77  
FORWARD ALL TRAPS;
```

L'AGENT : Définit que des agents de maître incompatibles sont encapsulés. Chaque entrée d'AGENT définit quel port l'agent de maître encapsulé écoute pour l'arrivée des requêtes SNMP et que la communauté tend le Pair d'Oracle SNMP doit employer en demandes de SNMP envoyées à cet agent.

Les SOUS-ARBRES : Identifient lequel des sous-arbres gérés par l'agent encapsulé doit être fait visible au NMS en inscrivant les identificateurs d'objet.

EXPÉDIEZ DES PIÈGES : Définit qui prend au trap envoyé par des agents encapsulés sur un hôte seront expédié par le Pair d'Oracle SNMP à l'agent de maître Encapsulator

IV. Exemple de gestionnaire de réseau :

IV.1 HP Open View:

Open-View est la solution pour gérer les réseaux informatiques constitués de matériels hétérogènes proposé par Hewlett-Packard. C'est une solution complète de gestion de systèmes distribués de base sur lequel peuvent tourner toutes les applications de gestion. Il offre les services de gestion communs et fournit une interface de programmation d'application normalisée. L'ensemble des activités de gestion tel que performance et faute sont proposé par une série d'applications. De même, il y a un ensemble d'utilitaires de fournis permettant l'analyse, le diagnostic et l'étude des réseaux locaux et réseaux longue distance. Il est possible de gérer son adressage de surveiller, contrôler et planifier son environnement.

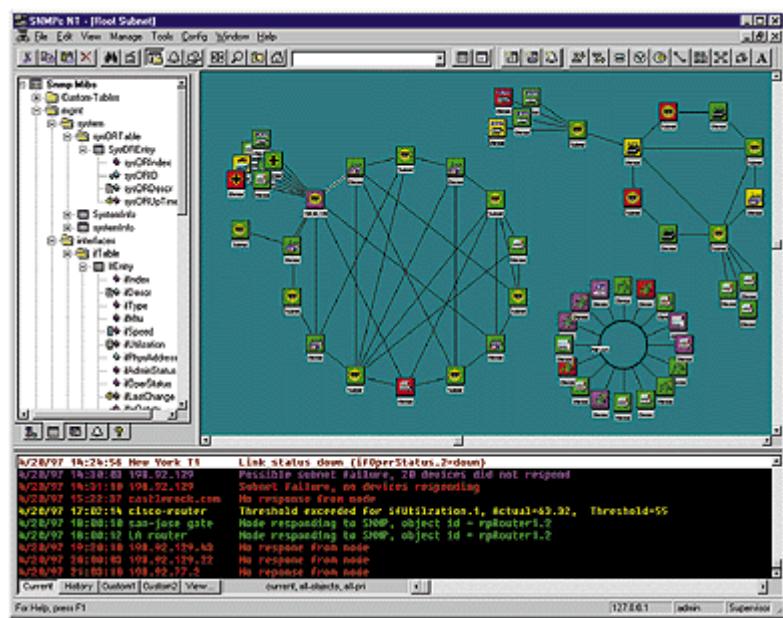
Open-View est décomposé en quatre parties au niveau de son architecture :

- les utilisateurs du réseau : différents intervenants sur le système et leurs droits de décision propre.

- les objets gérés : les objets liés à la transmission, les objets concernant les données réparties, les objets liées aux traitements répartis, les objets liés aux applications.
- les fonctionnalités : elles sont basées sur celles définies dans la normalisation d'ISO.
- les services et supports : ensemble de services et de supports de développement permettant d'adapter les spécificités du système à celles de l'entreprise.

IV.2 Gestionnaire De Réseau SNMPc 5.0 :

Le gestionnaire de réseau de SNMPc 5,0 est une suite scalable et distribuée de gestion pour contrôler petite à de grands réseaux. SNMPc utilise les composants multiples fonctionnant sur différents ordinateurs, et emploie la technologie de base de données répartie pour fournir une plateforme de rendement élevé indépendamment de la taille ou de la configuration de votre réseau.



IV.2.1 Console à distance :

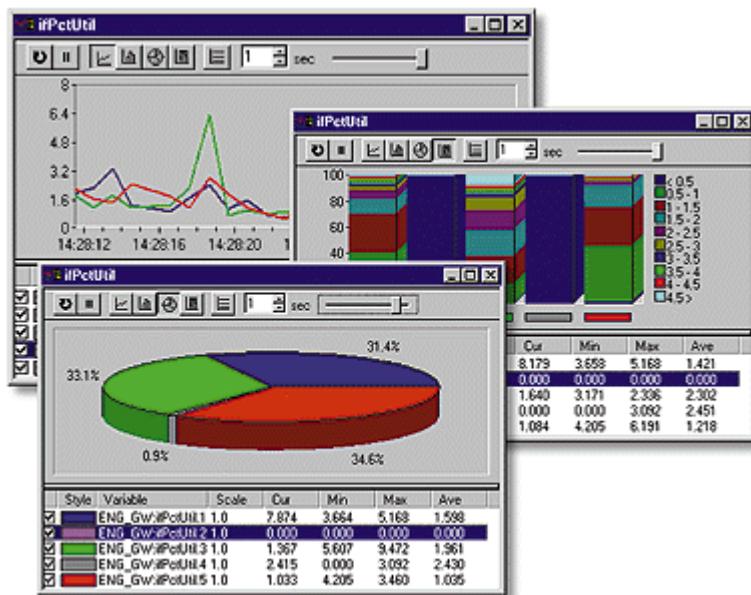
SNMPc fournit l'accès d'utilisateur local dans les éditions de équipe de travail et d'entreprise. Supplémentaire, l'édition d'entreprise supporte l'accès à distance au-dessus d'une connexion élevée de vitesse avec la console à distance. Vous pouvez exécuter l'application à distance de console de n'importe quel poste de travail de Windows en utilisant une connexion locale ou à distance de TCP/IP. Toutes les fonctions d'entreprise de SNMPc sont fournies par les consoles à distance. Chaque console peut avoir accès lecture/écriture complet, indépendant d'autres sessions de console.

Le système de base d'entreprise de SNMPc inclut un permis à distance de console. Avec l'option d'extension d'accès à distance vous pouvez simultanément utiliser un nombre illimité de consoles à distance sur les systèmes multiples.

IV.2.2 Affichages En temps réel De MIB :

SNMPc supporte les affichages en temps réel des données variables de MIB dans le format tabulaire ou graphique. Les affichages de tableau ou de graphique peuvent inclure les dispositifs multiples et les variables de MIBs standard et privé différent, y compris des expressions dérivées telles que l'utilisation, le volume, et les erreurs de pour cent. Vous pouvez choisir toutes les cellules, lignes de table ou colonnes variables, pour l'affichage graphique. Vous pouvez modifier et placer n'importe quelle variable de MIB directement d'un affichage de Tableau en utilisant "l'édition" sur place.

L'affichage de graphique montre des données comme graphe par ligne, diagramme circulaire circulaire, diagramme à barres, ou distribution. L'affichage de graphique montre également le minimum, le maximum, et les valeurs moyennes pour chacun variable.



V. Conclusion :

Ce chapitre définit l'introduction à la réalisation pratique de notre projet. Au premier nous avons configuré les fichiers Tnsnames.ora et Listener.ora pour que les utilisateurs puissent accéder à la base de données d'Oracle. Ensuite pour que l'agent SNMP d'Oracle écoute le réseau et soit près pour recevoir des requêtes ou bien l'envoie d'une trap nous avons configuré les fichiers Master.cfg et Encap.cfg ce dernier c'est pour l'encapsulation des informations.

Après la mise en place de la configuration nous avons passé à l'interfaçage d'un gestionnaire de réseau comme le HP Openview ou SNMPc pour accéder à la base MIB d'Oracle et faire les tests nécessaires pour assurer le bon fonctionnement de notre gestionnaire

CONCLUSION GENERALE

Rappelons que les objectifs de ce projet sont doubles :

- Présenter une étude permettant de faire le point sur l'administration des agents SNMP d'Oracle.
- Mettre en place un système d'administration qui satisfaire nos besoin étudiés.

Après un premier chapitre introduction générale qui indique le cadre de ce projet son cahier de charge et une présentation de société d'accueil IRSIT, La description des différents fonctions de la gestion d'un réseau informatique a fait l'objet de deuxième chapitre.

Le troisième chapitre est consacrée pour l'étude de protocole de gestion SNMP qui assure l'échange des données administratifs entre le poste d'administration et les agents SNMP d'Oracle.

Le quatrième chapitre qui interprète les variables MIB privés d'Oracle, leur classification dans des tables, la composition de l'OID de chaque variable et son format. Puisque Oracle Enterprise Manager ne peut pas exploiter ces variables on a besoin d'appeler un gestionnaire de réseau, cette procédure fait l'objet de cinquième chapitre. Avant l'interfaçage entre l'agent SNMP d'Oracle et un gestionnaire de réseau il faut qu'il doit être configuré pour l'écoute de réseau et l'encapsulation des données. Ensuite nous avons présenté le principe de fonctionnement d'un gestionnaire de réseau en cas de l'envoie des requêtes ou la capture des traps. Ce chapitre sera détaillé en pratique le jour de soutenance.

La réalisation de ce projet nous a permis de viser essentiellement deux objectifs. Le premier est professionnel pour la familiarisation dans le domaine de recherche et la pratique, en touchant l'environnement Oracle ainsi que l'administration de leurs agents SNMP. Le deuxième objectif est scolaire pour la préparation d'une bonne mémoire de fin d'étude qui réponde au cahier de charge.

BIBLIOGRAPHIE

Ouvrage :

[1] Abdelaziz Abdellatif, Mohamed Limame, Abdelmalek Zeroual

“ Oracle 7 “, Edition EYROLLES 1996

[2] Marshall Rose

“ Gestion des réseaux ouverts”, Edition INTEREDITIONS

[3] Gilles Briard

« Oracle8 pour windows NT », Edition EYROLLES 1998

[4] Jean Pierre et Michelle Claude

« Gestion des réseaux informatiques », Edition EYROLLES 1993

Mémoire :

- Administration des réseaux hétérogènes Application au réseau de ESPTT.
1993/1994
- Mise en œuvre d'un gestionnaire de réseaux pour le réseau de l'IRSIT.
1998/1999

Documentation :

<http://www.oracle.com>

<http://www.castlerock.com>

<http://www.epita.com>

<http://www.oracle.fr/services/education/catalogue/techno/reseau/net8.html>

<http://testdb.athenet.net:88/oranti/doc/server.815/a68694/Output/apg.htm#625131>

<http://itlab01.bus.iastate.edu/faqOracle.htm>

Annexe

A. L'environnement réseau d' Oracle :

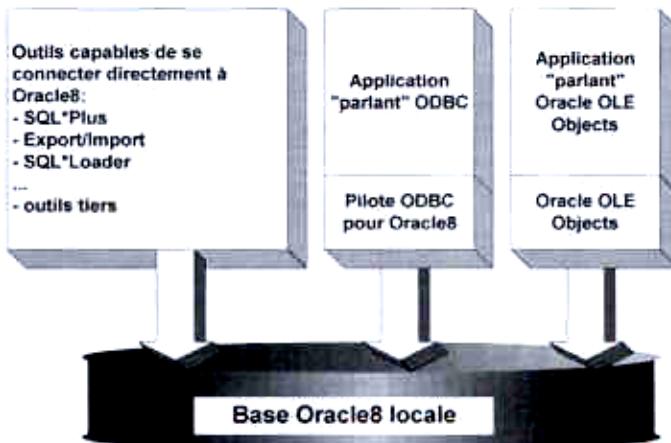
A.1. Qu'est-ce que le middleware Net8 ?

La majorité des experts s'accorde à définir le middleware comme regroupement de logiciels et de matériels mis en œuvre entre votre application cliente et votre serveur de données.

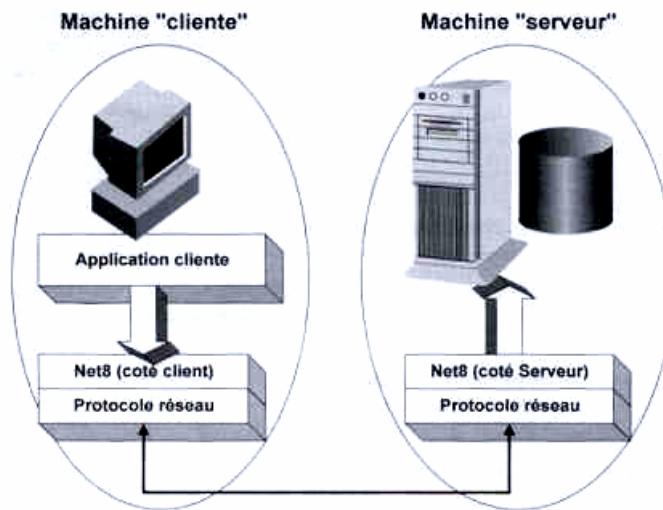
La partie matérielle n'est pas de la compétence d'Oracle, vous choisissez le type de réseau, son câblage, les cartes de communication, etc. Néanmoins, Oracle s'interface avec les logiciels que vous aurez choisis ou qui seront déjà installés sur votre poste.

Pour les produit Oracle, le middleware se compose de Net8 ainsi que des outils de communication comme Oracle Objects pour OLE et le pilote ODBC pour Oracle, ou encore les composants JDBC pour Java.

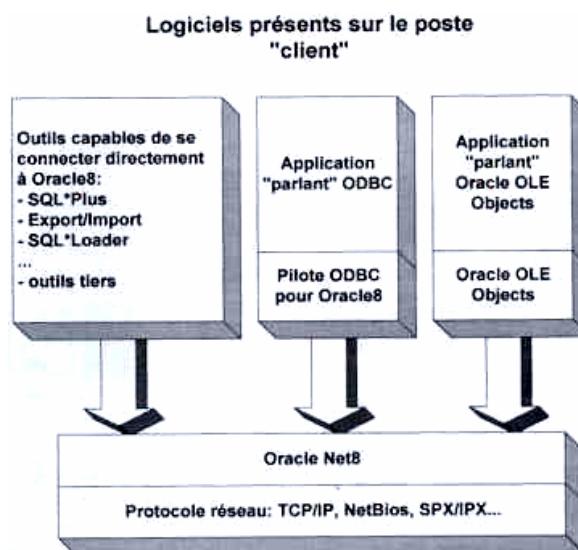
Si votre base de données et votre application sont sur la même machine, les logiciels de middleware « s'accrochent » directement à la base de données.



Si votre base de données et votre application ne sont pas sur la même machine, vous avez besoin d'un réseau entre le poste client et le serveur. Comme Net8 s'appuie sur le réseau que vous utilisez, il faut disposer de Net8 adapté au protocole réseau aussi sur le poste client que sur le serveur.

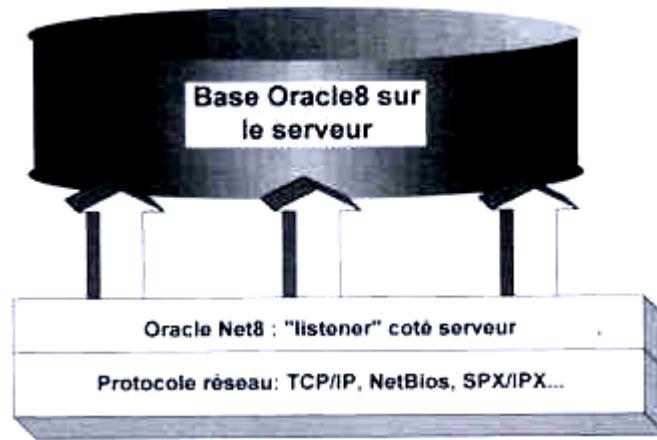


Dans le cas d'une architecture client-serveur, les logiciels de middleware comprennent en plus Net8 et le protocole de communication. La partie client et la partie serveur de Net8 doivent être installées et configurées séparément.

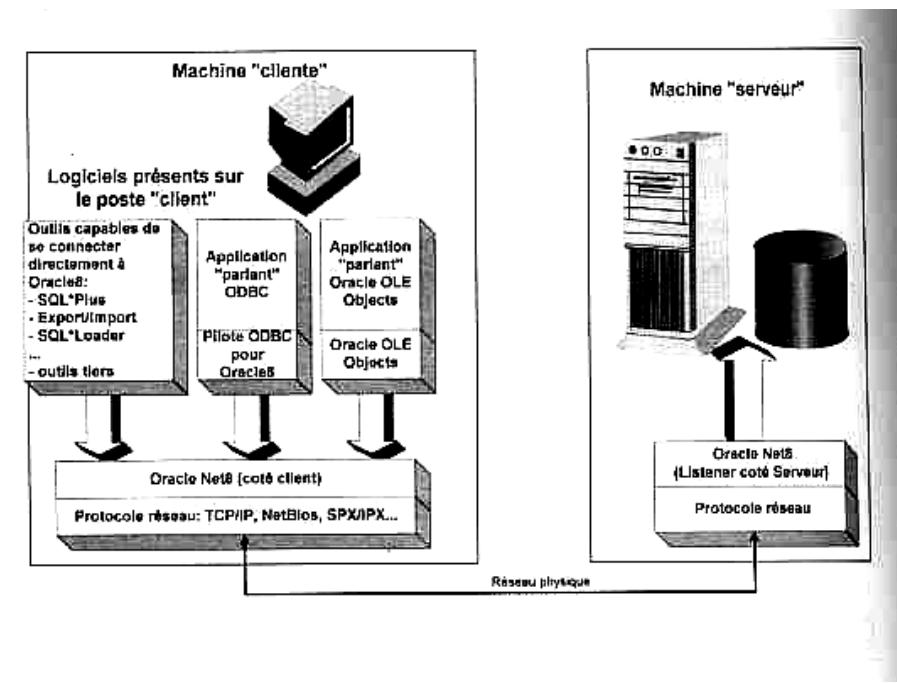


Tous les outils destinés à fonctionner en client-serveur s'interfacent à Net8. Certains proviennent d'Oracle, d'autre sont développés par des sociétés tierces. C'est le cas de la majorité des outils et des applications présents sur le marché.

Des logiciels peuvent accéder à Oracle en utilisant le pilote ODBC ou l'interface OLE2. Un tableur comme Excel a le choix d'utiliser l'un ou l'autre ! C'est à vous de choisir le chemin à emprunter, chaque interface ayant ses avantages et ses inconvénients. (ce point est détaillé en annexe en illustrant des exemples avec Excel et WORD).



Le schéma technique positionnant l'ensemble des couches logicielles mises en œuvre lors d'une relation client-serveur est donc :

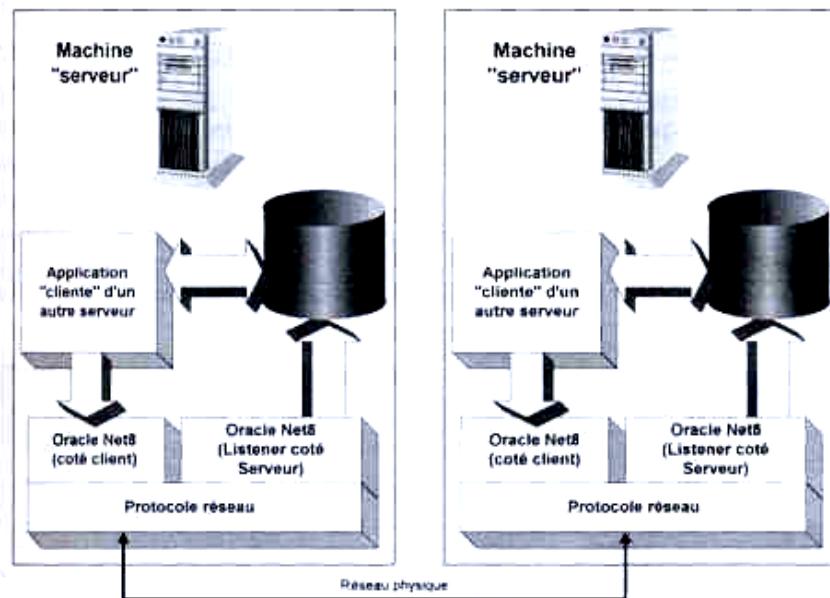


Ce schéma vous permet de bien comprendre le rôle « pivot » de Net8 dans une relation client-serveur.

A.1.1 Comment dialoguer entre plusieurs serveur Oracle :

Ce sont exactement les même couches logicielles qui sont mise en œuvre dans le cadre d'une relation serveur/serveur.

Dans une relation serveur/serveur, il faut que, sur chaque serveur, soient installés à la fois Oracle Net8 client et Oracle Net8 serveur et que chaque serveur soit configuré à la fois comme serveur, mais aussi comme client. Il y a aucune configuration spécifique à dialogue serveur/serveur à prévoir. A un instant donné, le serveur se composant comme client d'un autre serveur. Tout ceci est parfaitement transparent lors de l'utilisation.



A.1.2 Oracle Net8 :

Net8 assure un lien transparent entre les applications situées sur le poste client et la base de données située sur le serveur. Si le poste client et la base de données sont sur la même machine, vous n'avez pas besoin d'installer Net8. C'est le cas de Personal Oracle, mais toute application client-serveur utilisant un réseau a besoin de Net8.

Pour se connecter à Oracle8, il faut trois paramètres :

- Le nom d'utilisateur (déclaré dans la base Oracle cible) ;
- Le mot de passe (déclaré dans la base Oracle cible) ;
- Un alias pour se connecter à la base cible.

C'est cet alias qui force à utiliser Net8. Il regroupe sous son nom tout un ensemble de renseignement :

- Le protocole réseau utilisé pour accéder à la machine cible (TCP/ IP, NetBios, SPX/ IPX ...) ;
- La machine cible où se situe la base (pour le protocole utilisé) ;
- Comme la machine cible peut accueillir plusieurs bases, le SID de la base cible;

D'autres paramètres dépendent du protocole utilisé.

B MIB RMON :

Le principe de Remote Monitoring fut une extension inestimable pour le protocole SNMP. En effet, avec la MIB-II on a des informations purement locales, il est difficile d'avoir une vue globale du trafic par exemple. RMON informe le manager sur les couches basses du réseau en utilisant des sondes. RMON travail de façon passive, il peut faire des analyses sur tous les paquets d'un réseau et émettre des statistiques ou des rapports sur le nombre de collisions, d'erreurs, etc.

Le RMON est implémenté soit comme un système à part entière, soit comme une fonction sur une entité. Ainsi un PC sur un sous-réseau peut supporter l'agent RMON, mais les fonctionnalités sont alors bridées. En générale, il y a un agent RMON par sous-réseau. Le rôle de RMON est de détecter de façon passive, sans polling, des erreurs comme les congestions. S'il y a une coupure de ligne avec la NMS, RMON prend le relais pour sauver les informations administratives nécessaires pour les statistiques. RMON est aussi capable de faire des opérations plus évoluées, comme des analyses spécifiques à un type de donnée etc. Donc un agent RMON peut être autonome ou associé à une entité manageante et contrôlé par celle-ci à distance.

Défini dans la RFC 1271, une MIB RMON permet de standardiser les variables et les contrôles, c'est une sous branche de la MIB-II son identifiant est 1.3.6.1.2.1.16. Un agent Remote

Monitoring a besoin d'être configuré pour un type de donné particulier qui est dicté par le MIB RMON. Elle est organisée en neuf groupes fonctionnels :

Statistics : informations sur l'utilisation du niveau bas (physique) avec des statistiques d'erreurs et d'utilisation du sous-réseau que la sonde analyse.

History : enregistre périodiquement des extractions de statistiques de la table précédente.

Alarm : permet à l'opérateur de positionner un intervalle et un seuil déclencheur pour une valeur ou un compteur sélectionné.

Hosts : contient des compteurs de types variés sur le trafic entrant et sortant des hosts du segment.

HostTopN : contient des statistiques tirées sur les hosts selon les critères provenant du groupe précédent.

Matrix : sous forme matricielle sont présentées les erreurs d'utilisations associées à leurs origines.

Filter : permet d'enregistrer des paquets suivant un filtre. Tous les paquets passant sur le réseau peuvent être enregistrés, où ils servent juste pour l'établissement de statistiques.

Capture : détermine comment les données sont envoyées vers la station d'administration.

Event : table des événements générés par l'agent RMON.

Dans chaque groupe se trouve un certain nombre de table de contrôle et de données. Lors de la configuration de l'agent, on va spécifier le type de donnée à surveiller dans la table des contrôles. Ces tables sont mises à jour par affectation des valeurs, mais aussi par modification de la structure par ajout ou suppression de colonnes.

Il se peut qu'un agent RMON soit partagé par plusieurs managers ce qui peut entraîner des situations de conflits dans le cas de requêtes concurrentes. Ainsi une station de management peut mobiliser un agent au détriment des autres. Pour limiter ces problèmes lors du partage d'un agent RMON, à chaque colonne est associée une sorte d'étiquette marquant son appartenance à un propriétaire.

Face au succès de RMON, on s'est demandé pourquoi ne pas étendre ses fonctionnalités. En effet RMON est limité aux couches basses du modèle OSI, les deux premières. L'analyse se fait au niveau des adresses MAC donc limité au sous-réseau où se situe l'agent. RMON-II permet

d'étudier les couches supérieures, en déterminant par exemple le type de protocoles, le trafic entre deux serveurs distants etc

C .ODBC, Oracle Objets pour OLE, JDBC ET SQLJ :

Votre base Oracle est installée, votre middleware réseau Net8 également. Vos utilisateurs ont certainement besoin d'accéder, de récupérer et de manipuler les données contenues dans la base de données en les rapatriant dans leur outils familières, Excel et word par exemple.

Nous mettrons en œuvre les deux outils accès possibles livrés en standard avec Oracle8 : par le biais du pilotes ODBC pour Oracle OLE Objets. L'accès ODBC est important car un très grand nombre de logiciels utilisent cette technique pour se connecter à une bases Oracle 8. Nous aborderons ensuite la sécurité liée à ce type d'accès et nous aborderons ensuite la sécurité liée à ce type d'accès et nous proposons des critères pour choisir entre ODBC et OLE en fonction de vos besoins.

Enfin, Java et de plus en plus présent. Ce langage nécessite des moyens d'accès à Oracle8 : C'est l'objectif proposé par JDBC et SQLJ. Les pilotes JDBC sont livrés avec toutes les versions d'Oracle pour Windows NT.

C.1. Configuration d'un pilote ODBC :

◆ Présentation techniques :

L'emplacement d'un pilote ODBC et d'Oracle OLE Objects dans une chaîne de liaison avec Oracle8 a été présenté dans le chapitre 5 (NET8 , middleware Oracle). Pour mettre en œuvre ODBC et Oracle OLE Objects, il faut que Net8 soit installé et testé entre le poste client et votre serveur oracle8.

Il faut par ailleurs que les logiciels Oracle et/ou Oracle Objects soient installés sur votre poste client. Vous pouvez le vérifier à l'aide de l'Installer Oracle.

◆ Les pilotes ODBC :

Il y a quelque années, il était possible de configurer manuellement les pilotes ODBC en accédant directement aux fichiers ODBC.INI. Cela n'est plus possible maintenant dans les environnement 32 bits (Windows NT, Windows 95) car la configuration ODBC doit être synchronisée entre le fichier « .INI » et la Registry du système (certains outils clients utilisent l'un

ou l'autre). Les deux configurations sont gérées en même temps par l'administrateur ODBC de Microsoft.

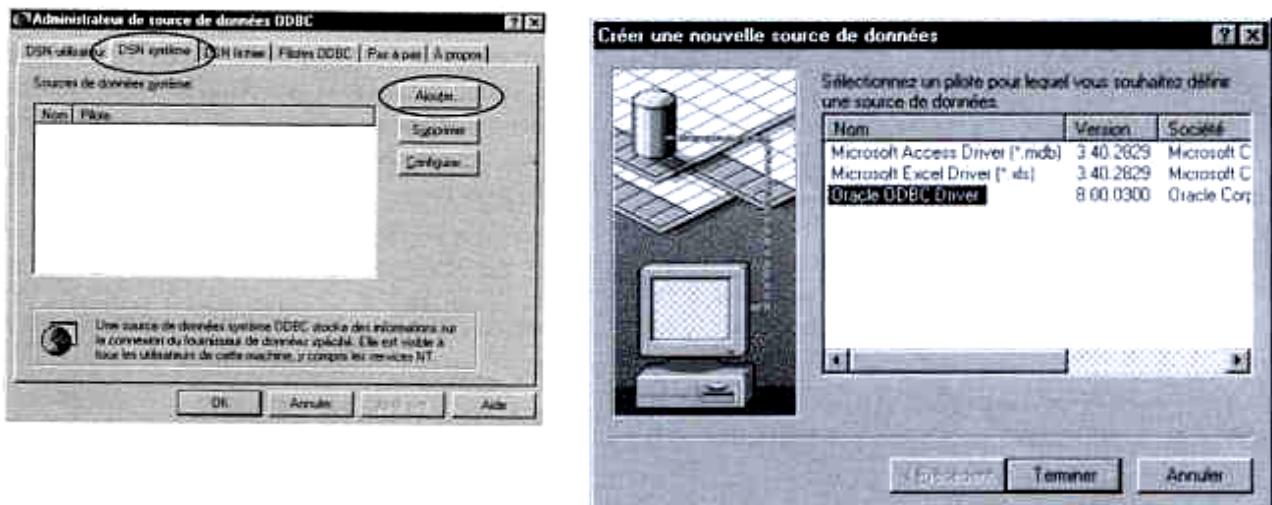
Oracle fournit un point d'entrée pour accéder à l'administrateur ODBC et un utilisateur de test pour valider la liaison ODBC.

Pour commencer, il faut créer un Data Source Name, c'est-à-dire un alias ODBC qui permet à toute liaison ODBC de déterminer :

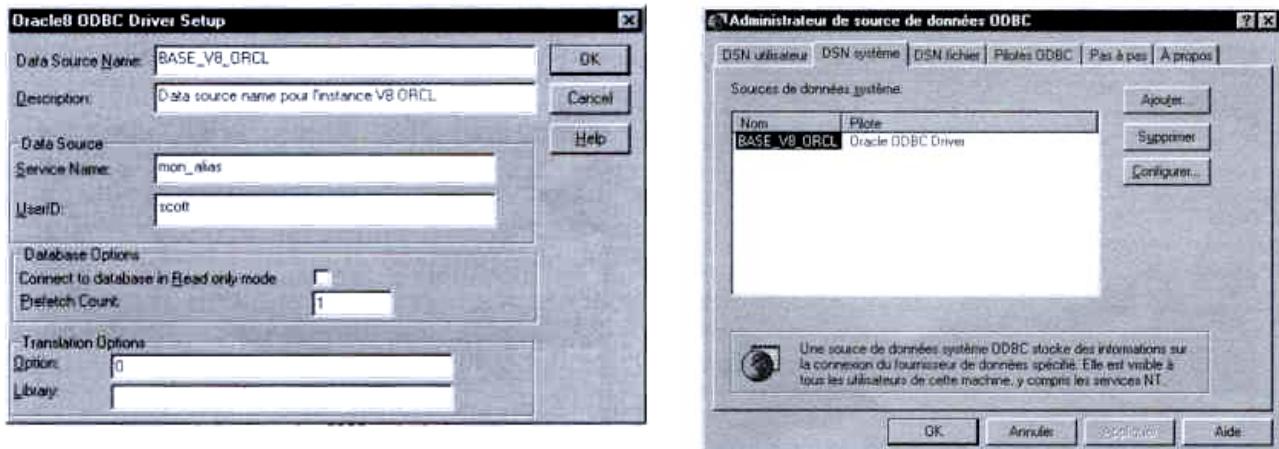
- vers quelle fournisseur de base de données il pointe : ORACLE ;
- quel alias Net8 il va utiliser (le protocole, la machine cible, l'instance cible sont alors connus).

Lancer l'administrateur ODBC : Démarrer>Programmes>Oracle pour Windows NT>Microsoft ODBC Administrateur. Cliquer sur l'onglet DSN système puis sur Ajouter.

Parmi la liste de pilotes proposés, choisissez celui pour Oracle8.

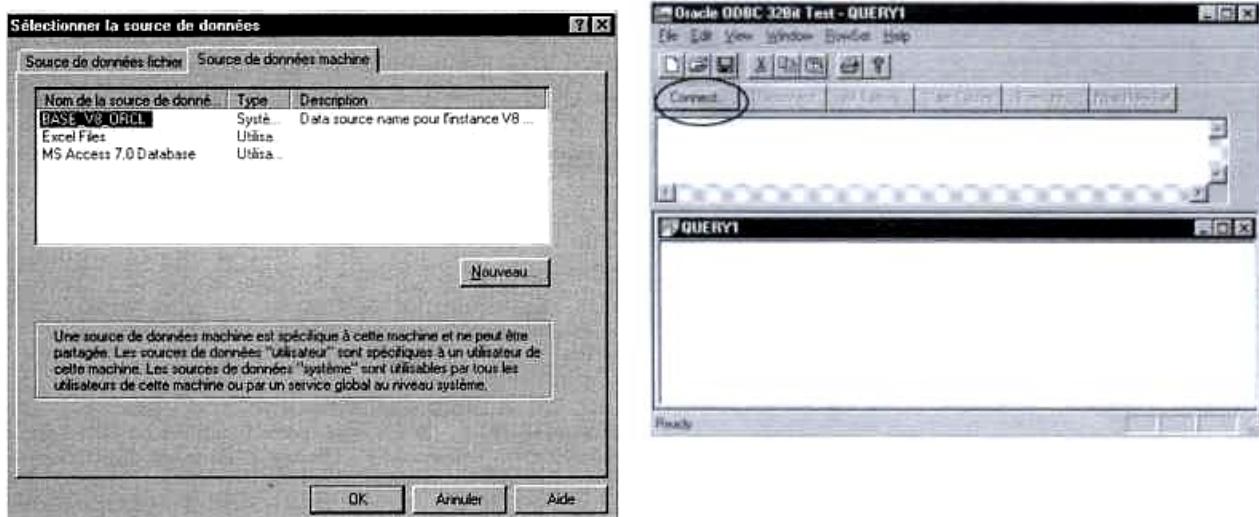


L'étape suivante précise l'alias Net8 que nous allons utiliser, puis un nom d'utilisateur Oracle qui sera proposé par défaut.

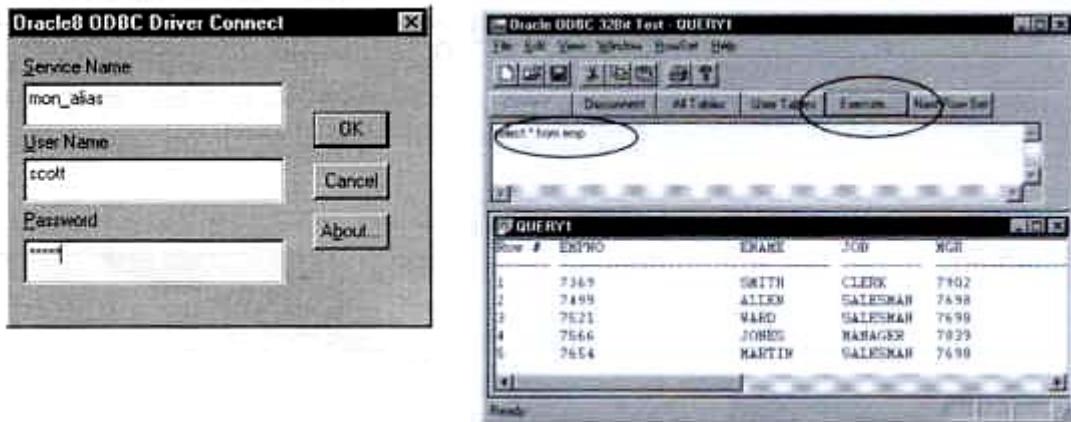


Votre DNS (Data Source Name) est maintenant configuré, vous allez le tester avec l'utilisateur fourni par Oracle. Pour y accéder, lancer : Démarrer>Programmes>Oracle pour WindowsNT>Oracle ODBCTest.

Cliquez sur Connect et choisissez le DSN que vous venez de créer



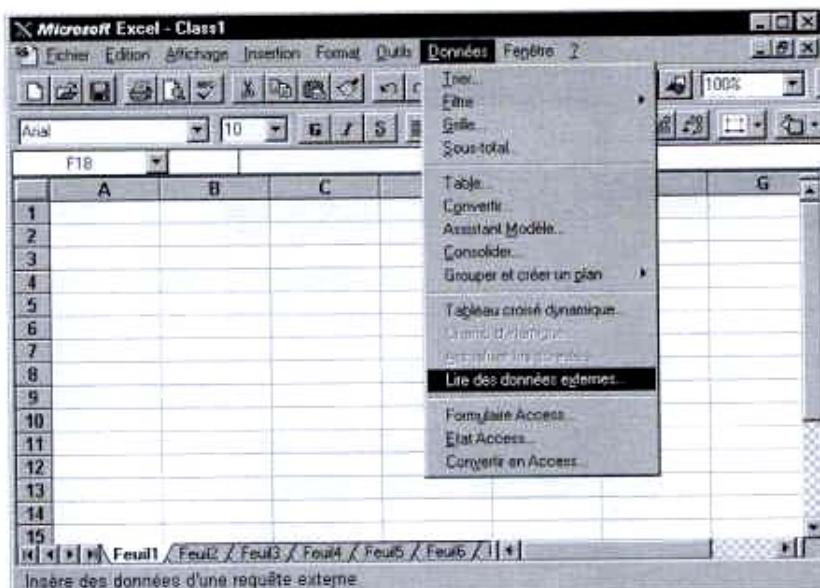
Saisissez le mot de passe de l'utilisateur Scott (mot de passe Tiger). Vous êtes maintenant connecté à la base cible. Vérifiez si le fonctionnement est correct en saisissant un ordre SQL dans la fenêtre supérieure puis en cliquant sur Execute.



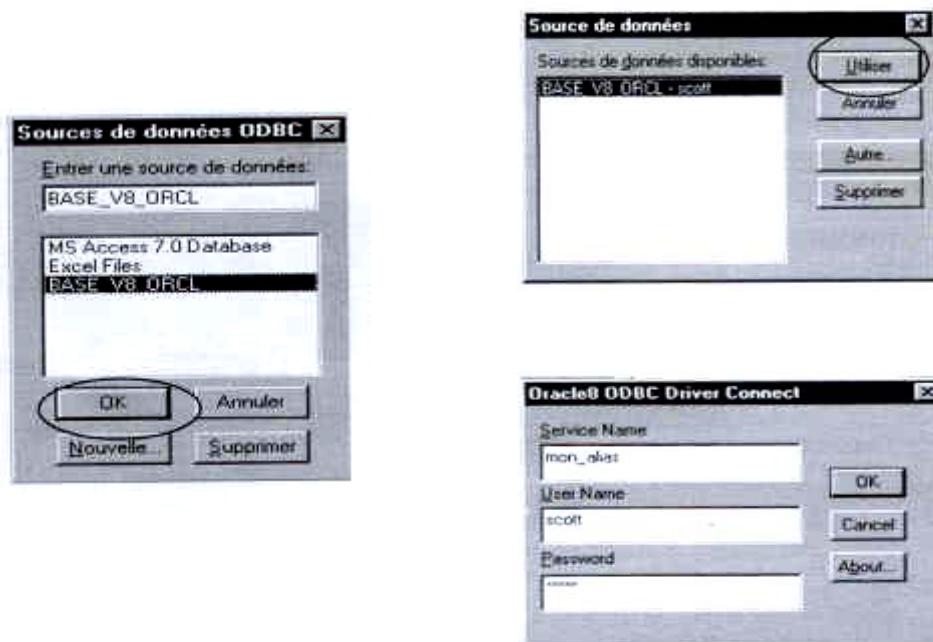
Votre configuration ODBC fonctionne. Vous pouvez réutiliser ce DSN à partir de tous les outils qui utilisent ODBC pour accéder à Oracle. C'est le cas dans l'exemple suivant où nous allons faire le lien entre Excel et Oracle8.

C.2. Lien ODBC entre Excel et Oracle8 :

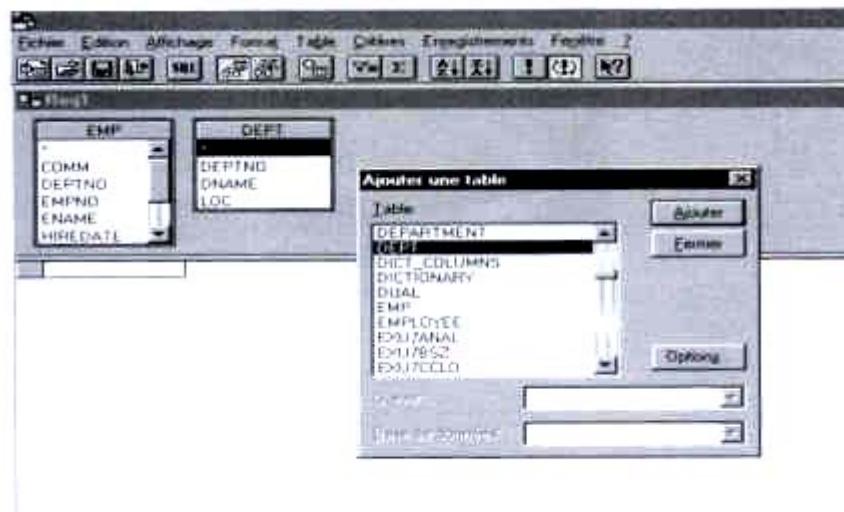
Cet exemple vous montre comment rapatrier dans Excel7 des données provenant d'Oracle8. Tous les produits utilisés sont inclus dans Microsoft Office. Lancez Excel7. Au préalable, il faut que Net8 et le pilote ODBC pour Oracle soient installés et fonctionnent. A partire d'Excel7, exécutez Lire des données externes. Cela lance le produit Microsoft Query (qui est exécutable séparément d' Excel7).



Choisissez le Data Source Name pour la base comme source de données ODBC, puis connectez-vous au compte Oracle Scott (mot de passe Tiger).



Vous êtes maintenant connectez à Oracle8 et l'ensemble des tables, vues, synonymes de l'utilisateur Scott sont affichés. Sélectionnez la table EMP et DEPT.

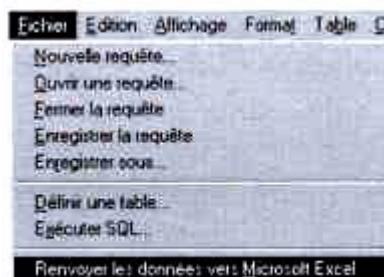


Effectuez une jointure entre les deux tables en effectuant un glisser-déplacer de la colonne DEPTNO de table DEPT. Cliquez ensuite sur les colonnes pour afficher les données.

* Reql

EMP		DEPT		
ENAME	JOB	SAL	COMM	DNAME
SMITH	CLERK	800		RESEARCH
ALLEN	SALESMAN	1600	300	SALES
WARD	SALESMAN	1250	500	SALES
JONES	MANAGER	2975		RESEARCH
MARTIN	SALESMAN	1250	1400	SALES
BLAKE	MANAGER	2850		SALES
CLARK	MANAGER	2450		ACCOUNTING
SCOTT	ANALYST	3000		RESEARCH
KING	PRESIDENT	5000		ACCOUNTING
TURNER	SALESMAN	1500	0	SALES
ADAMS	CLERK	1100		RESEARCH
JAMES	CLERK	950		SALES
FORD	ANALYST	3000		RESEARCH
MILLER	CLERK	1300		ACCOUNTING

Quittez l'environnement Microsoft Query pour retourner vers Excel7.



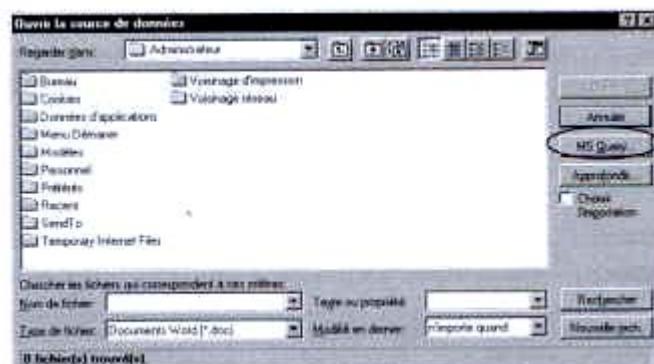
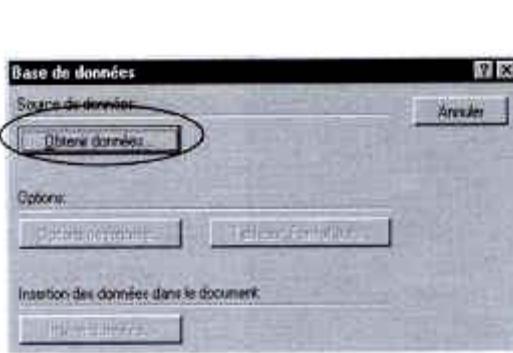
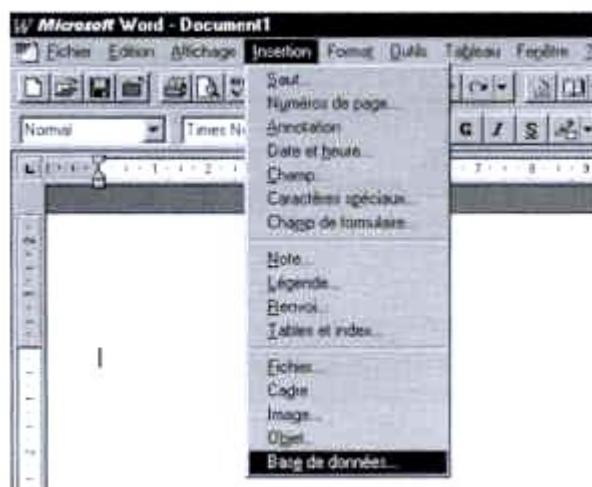
Les données sont récupérées dans votre environnement familier.

Microsoft Excel - Class1

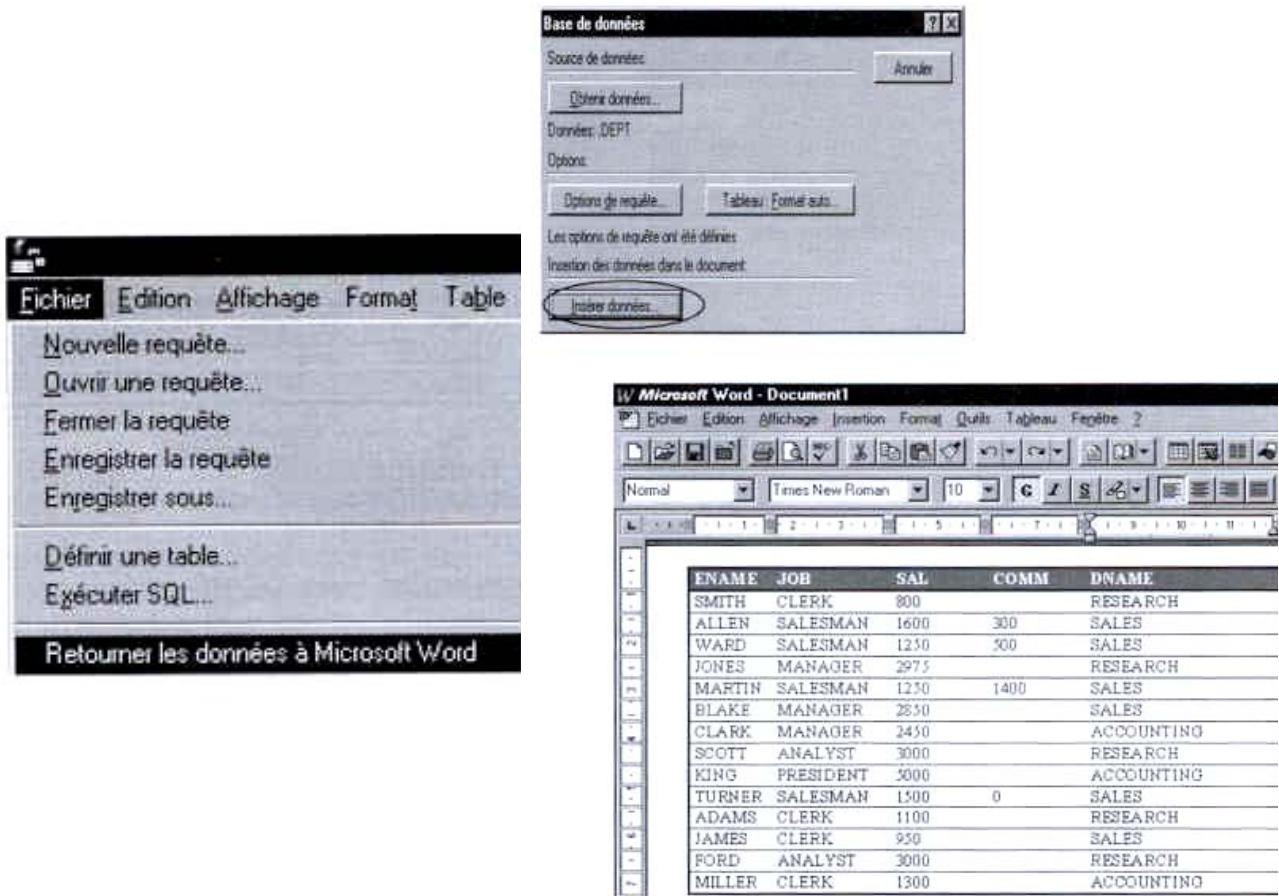
A	B	C	D	E	F
1 ENAME	JOB	SAL	COMM	DNAME	
2 SMITH	CLERK	800		RESEARCH	
3 ALLEN	SALESMAN	1600	300	SALES	
4 WARD	SALESMAN	1250	500	SALES	
5 JONES	MANAGER	2975		RESEARCH	
6 MARTIN	SALESMAN	1250	1400	SALES	
7 BLAKE	MANAGER	2850		SALES	
8 CLARK	MANAGER	2450		ACCOUNTING	
9 SCOTT	ANALYST	3000		RESEARCH	
10 KING	PRESIDENT	5000		ACCOUNTING	
11 TURNER	SALESMAN	1500	0	SALES	
12 ADAMS	CLERK	1100		RESEARCH	
13 JAMES	CLERK	950		SALES	
14 FORD	ANALYST	3000		RESEARCH	
15 MILLER	CLERK	1300		ACCOUNTING	
16					
17					
18					

C .3. Lien ODBC entre Word et Oracle8 :

Nous allons réaliser le même exemple à partir de Word7. la démarche est tout à fait la similaire et les prérequis d'installation sont les même que pour Excel7. Suivez-nous pas à pas, à partir de Word7 :



Jusqu'au retour dans Word7, les étapes dans Microsoft Query sont exactement les mêmes que dans l'exemple précédent.



A partire vos données sont rapatriées dans Word7.

D. Interprétation des variables MIB d'Oracle :

- D.1 Interprétation des Variables de la Base de Données d'Oracle MIB :**
- ♦ Les variables d'oraDbSysTable et leurs correspondances IDs :

Noms de variables	Identificateur ID
OraDbSysConsistentChanges	1.3.6.1.4.1.111.4.1.1.1.1
OraDbSysConsistentGets	1.3.6.1.4.1.111.4.1.1.1.2
OraDbSysDbBlockChanges	1.3.6.1.4.1.111.4.1.1.1.3
OraDbSysDbBlockGets	1.3.6.1.4.1.111.4.1.1.1.4
oraDbSysFreeBufferInspected	1.3.6.1.4.1.111.4.1.1.1.5
oraDbSysFreeBufferRequested	1.3.6.1.4.1.111.4.1.1.1.6
OraDbSysParseCount	1.3.6.1.4.1.111.4.1.1.1.7

OraDbSysPhysReads	1.3.6.1.4.1.111.4.1.1.1.8
OraDbSysPhysWrites	1.3.6.1.4.1.111.4.1.1.1.9
OraDbSysRedoEntries	1.3.6.1.4.1.111.4.1.1.1.10
oraDbSysRedoLogSpaceRequests	1.3.6.1.4.1.111.4.1.1.1.11
OraDbSysredoSyncWrites	1.3.6.1.4.1.111.4.1.1.1.12
OraDbSysSortsDisk	1.3.6.1.4.1.111.4.1.1.1.13
OraDbSysSortsMemory	1.3.6.1.4.1.111.4.1.1.1.14
OraDbSysSortsRows	1.3.6.1.4.1.111.4.1.1.1.15
OraDbSysTableFetchRowid	1.3.6.1.4.1.111.4.1.1.1.16
oraDbSysTableFetchContinuedRow	1.3.6.1.4.1.111.4.1.1.1.17
OraDbSysTableScanBlocks	1.3.6.1.4.1.111.4.1.1.1.18
OraDbSysTableScanRows	1.3.6.1.4.1.111.4.1.1.1.19
OraDbSysTableScansLong	1.3.6.1.4.1.111.4.1.1.1.20
OraDbSysTableScansShort	1.3.6.1.4.1.111.4.1.1.1.21
OraDbSysUserCalls	1.3.6.1.4.1.111.4.1.1.1.22
OraDbSysUserCommits	1.3.6.1.4.1.111.4.1.1.1.23
OraDbSysUserRollbacks	1.3.6.1.4.1.111.4.1.1.1.24
OraDbSysWriteRequests	1.3.6.1.4.1.111.4.1.1.1.25

♦ Les variables oraDbTablespaceTable et leurs correspondances IDs :

Noms de variables	Identificateur ID
OraDbTablespaceIndex	1.3.6.1.4.1.111.4.1.2.1.1
OraDbTablespaceName	1.3.6.1.4.1.111.4.1.2.1.2
OraDbTablespaceSizeAllocated	1.3.6.1.4.1.111.4.1.2.1.3
OraDbTablespaceSizeUsed	1.3.6.1.4.1.111.4.1.2.1.4
OraDbTablespaceState	1.3.6.1.4.1.111.4.1.2.1.5
OraDbTablespaceLargestAvailableChunk	1.3.6.1.4.1.111.4.1.2.1.6

♦ Les variables d'oraDbDataFileTable et leurs correspondances IDs :

Noms de variables	Identificateur ID
OraDbDatafileIndex	1.3.6.1.4.1.111.4.1.3.1.1
OraDbDatafileName	1.3.6.1.4.1.111.4.1.3.1.2
OraDbDatafileSizeAllocated	1.3.6.1.4.1.111.4.1.3.1.3
OraDbDatafileDiskReads	1.3.6.1.4.1.111.4.1.3.1.4
OraDbDatafileDiskWrites	1.3.6.1.4.1.111.4.1.3.1.5
OraDbDatafileDiskReadBlocks	1.3.6.1.4.1.111.4.1.3.1.6
OraDbDatafileDiskWrittenBlocks	1.3.6.1.4.1.111.4.1.3.1.7
OraDbDatafileDiskReadTimeTicks	1.3.6.1.4.1.111.4.1.3.1.8
OraDbDatafileDiskWriteTimeTicks	1.3.6.1.4.1.111.4.1.3.1.9

♦ Les variables d'oraDbLibraryCacheTable et leurs correspondances IDs :

Noms de variables	Identificateur ID
OraDbLibraryCacheIndex	1.3.6.1.4.1.111.4.1.4.1.1
OraDbLibraryCacheNameSpace	1.3.6.1.4.1.111.4.1.4.1.2
OraDbLibraryCacheGets	1.3.6.1.4.1.111.4.1.4.1.3
OraDbLibraryCacheGetHits	1.3.6.1.4.1.111.4.1.4.1.4
OraDbLibraryCachePins	1.3.6.1.4.1.111.4.1.4.1.5
OraDbLibraryCachePinHits	1.3.6.1.4.1.111.4.1.4.1.6
OraDbLibraryCacheReloads	1.3.6.1.4.1.111.4.1.4.1.7
OraDbLibraryCacheInvalidations	1.3.6.1.4.1.111.4.1.4.1.8

OaRepPending2PhaseCommits	1.3.6.1.4.1.111.4.3.1.1.1
OaRepDisabledTriggers	1.3.6.1.4.1.111.4.3.1.1.2
OaRepInvalidPackages	1.3.6.1.4.1.111.4.3.1.1.3
OaRepSchemas	1.3.6.1.4.1.111.4.3.1.1.4

- ◆ Les variables d'oraRepSchemaTable et leurs correspondances IDs :

Noms de variables	Identificateur ID
OaRepSchemaIndex	1.3.6.1.4.1.111.4.3.2.1.1
OaRepSchemaName	1.3.6.1.4.1.111.4.3.2.1.2
OaRepSchemaType	1.3.6.1.4.1.111.4.3.2.1.3
OaRepSchemaStatus	1.3.6.1.4.1.111.4.3.2.1.4
OaRepSchemaLinks	1.3.6.1.4.1.111.4.3.2.1.5

- ◆ Les variables d'oraRepMasterSchemaTable et leurs correspondances IDs :

Noms de variables	Identificateur ID
OaRepMasterSchemaRequestErrors	1.3.6.1.4.1.111.4.3.3.1.1
OaRepMasterSchemaResolvedConflicts	1.3.6.1.4.1.111.4.3.3.1.2
OaRepMasterSchemaOtherMasters	1.3.6.1.4.1.111.4.3.3.1.3

- ◆ Les variables d'oraRepSnapshotSchemaTable et leurs correspondances IDs:

Noms de variables	Identificateur ID
OaRepSnapshotSchemaMaster	1.3.6.1.4.1.111.4.3.4.1.1
OaRepSnapshotSchemaSnapshots	1.3.6.1.4.1.111.4.3.4.1.2

- ◆ Les variables d'oraRepLinkTable et leurs correspondances IDs :

Noms de variables	Identificateur ID
OaRepLinkIndex	1.3.6.1.4.1.111.4.3.5.1.1
OaRepLinkName	1.3.6.1.4.1.111.4.3.5.1.2
OaRepLinkDeferredTransactions	1.3.6.1.4.1.111.4.3.5.1.3
OaRepLinkErrors	1.3.6.1.4.1.111.4.3.5.1.4
OaRepLinkAdminRequests	1.3.6.1.4.1.111.4.3.5.1.5

D.2 Interprétation des Variables MIB de Listener d'Oracle :

- ◆ Les variables d'oraListenerTable et leurs correspondances IDs :

Noms de variables	Identificateur ID
OraListenerIndex	1.3.6.1.4.1.111.5.1.1.1.1
OraListenerName	1.3.6.1.4.1.111.5.1.1.1.2
OraListenerVersion	1.3.6.1.4.1.111.5.1.1.1.3
OraListenerStartDate	1.3.6.1.4.1.111.5.1.1.1.4
OraListenerUptime	1.3.6.1.4.1.111.5.1.1.1.5
OraListenerTraceLevel	1.3.6.1.4.1.111.5.1.1.1.6
OraListenerSecurityLevel	1.3.6.1.4.1.111.5.1.1.1.7
OraListenerParameterFile	1.3.6.1.4.1.111.5.1.1.1.8
OraListenerLogFile	1.3.6.1.4.1.111.5.1.1.1.9
OraListenerTraceFile	1.3.6.1.4.1.111.5.1.1.1.10
OraListenerState	1.3.6.1.4.1.111.5.1.1.1.11
OraListenerNumberOfServices	1.3.6.1.4.1.111.5.1.1.1.12
OraListenerContact	1.3.6.1.4.1.111.5.1.1.1.13

- ◆ Les variables d'oraSIDTable et leurs correspondances IDs :

Noms de variables	Identificateur ID
OraSIDListenerIndex	1.3.6.1.4.1.111.5.1.5.1.1
OraSIDName	1.3.6.1.4.1.111.5.1.5.1.2
OraSIDCurrentConnectedClients	1.3.6.1.4.1.111.5.1.5.1.3
OraSIDReservedConnections	1.3.6.1.4.1.111.5.1.5.1.4

- ◆ Les variables d'oraDedicatedSrvTable et leurs correspondances IDs :

Noms de variables	Identificateur ID
OraDedicatedSrvIndex	1.3.6.1.4.1.111.5.1.2.1.1
OraDedicatedSrvEstablishedConnections	1.3.6.1.4.1.111.5.1.2.1.2
OraDedicatedSrvRejectedConnections	1.3.6.1.4.1.111.5.1.2.1.3

- ◆ Les variables d'oraDispatcherTable et leurs correspondances IDs :

Noms de variables	Identificateur ID
OraDispatcherIndex	1.3.6.1.4.1.111.5.1.3.1.1
OraDispatcherEstablishedConnections	1.3.6.1.4.1.111.5.1.3.1.2
OraDispatcherRejectedConnections	1.3.6.1.4.1.111.5.1.3.1.3
OraDispatcherCurrentConnections	1.3.6.1.4.1.111.5.1.3.1.4
OraDispatcherMaximumConnections	1.3.6.1.4.1.111.5.1.3.1.5
OraDispatcherState	1.3.6.1.4.1.111.5.1.3.1.6
OraDispatcherProtocolInfo	1.3.6.1.4.1.111.5.1.3.1.7

- ◆ Les variables d'oraPrespawnedSrvTable et leurs correspondances IDs :

Noms de variables	Identificateur ID
OraPrespawnedSrvIndex	1.3.6.1.4.1.111.5.1.4.1.1
OraPrespawnedSrvEstablishedConnections	1.3.6.1.4.1.111.5.1.4.1.2
OraPrespawnedSrvRejectedConnections	1.3.6.1.4.1.111.5.1.4.1.3
OraPrespawnedSrvCurrentConnections	1.3.6.1.4.1.111.5.1.4.1.4
OraPrespawnedSrvMaximumConnections	1.3.6.1.4.1.111.5.1.4.1.5
OraPrespawnedSrvState	1.3.6.1.4.1.111.5.1.4.1.6
OraPrespawnedSrvProtocolInfo	1.3.6.1.4.1.111.5.1.4.1.7
OraPrespawnedSrvProcessorID	1.3.6.1.4.1.111.5.1.4.1.8

- ◆ Les variables d'oraListenAddressTable et leurs correspondances IDs :

Noms de variables	Identificateur ID
OraListenAddressIndex	1.3.6.1.4.1.111.5.1.6.1.1
OraListenAddress	1.3.6.1.4.1.111.5.1.6.1.2

- ◆ Les variables d'oraListenerTraps et leurs correspondances IDs :

Noms de variables	Identificateur ID
OraListenerStateChange	1.3.6.1.4.1.111.5.2.1.1.1

D.3 Interprétation des Variables MIB d'Oracle names :

- ◆ Les variables d'oraNamesTNSTable et leurs correspondances IDs :

Noms de variables	Identificateur ID
OraNamesTNSStartDate	1.3.6.1.4.1.111.6.1.1.1.1

OraNamesTNSTraceLevel	1.3.6.1.4.1.111.6.1.1.1.2
OraNamesTNSSecurityLevel	1.3.6.1.4.1.111.6.1.1.1.3
OraNamesTNSParameterFile	1.3.6.1.4.1.111.6.1.1.1.4
OraNamesTNSLogFile	1.3.6.1.4.1.111.6.1.1.1.5
OraNamesTNSTraceFile	1.3.6.1.4.1.111.6.1.1.1.6
OraNamesTNSState	1.3.6.1.4.1.111.6.1.1.1.7
OraNamesTNSContact	1.3.6.1.4.1.111.6.1.1.1.8
OraNamesTNSListenAddresses	1.3.6.1.4.1.111.6.1.1.1.9
OraNamesTNSFailedListenAddresses	1.3.6.1.4.1.111.6.1.1.1.10
OraNamesTNSReload	1.3.6.1.4.1.111.6.1.1.1.11
OraNamesTNSRunningTime	1.3.6.1.4.1.111.6.1.1.1.12

◆ Les variables d'oraNamesConfigTable et leurs correspondances IDs :

Noms de variables	Identificateur ID
OraNamesConfigAdminRegion	1.3.6.1.4.1.111.6.1.2.1.1
OraNamesConfigAuthorityRequired	1.3.6.1.4.1.111.6.1.2.1.2
OraNamesConfigAutoRefreshExpire	1.3.6.1.4.1.111.6.1.2.1.3
OraNamesConfigAutoRefreshRetry	1.3.6.1.4.1.111.6.1.2.1.4
OraNamesConfigCacheCheckpointFile	1.3.6.1.4.1.111.6.1.2.1.5
OraNamesConfigCacheCheckpointInterval	1.3.6.1.4.1.111.6.1.2.1.6
OraNamesConfigConfigCheckpointFile	1.3.6.1.4.1.111.6.1.2.1.7
OraNamesConfigDefaultForwarders	1.3.6.1.4.1.111.6.1.2.1.8
OraNamesConfigDefaultForwardersOnly	1.3.6.1.4.1.111.6.1.2.1.9
OraNamesConfigDomainCheckpointFile	1.3.6.1.4.1.111.6.1.2.1.10
OraNamesConfigDomainHints	1.3.6.1.4.1.111.6.1.2.1.11
OraNamesConfigDomains	1.3.6.1.4.1.111.6.1.2.1.12
OraNamesConfigForwarding-Available	1.3.6.1.4.1.111.6.1.2.1.13
OraNamesConfigForwardingDesired	1.3.6.1.4.1.111.6.1.2.1.14
OraNamesConfigLogDirectory	1.3.6.1.4.1.111.6.1.2.1.15
OraNamesConfigLogStatsInterval	1.3.6.1.4.1.111.6.1.2.1.16
OraNamesConfigLogUnique	1.3.6.1.4.1.111.6.1.2.1.17
OraNamesConfigMaxOpenConnections	1.3.6.1.4.1.111.6.1.2.1.18
OraNamesConfigMaxReforwards	1.3.6.1.4.1.111.6.1.2.1.19
OraNamesConfigMessagePool-StartSize	1.3.6.1.4.1.111.6.1.2.1.20
OraNamesConfigNoModifyRequests	1.3.6.1.4.1.111.6.1.2.1.21
OraNamesConfigNoRegionDatabase	1.3.6.1.4.1.111.6.1.2.1.22
OraNamesConfigResetStatsInterval	1.3.6.1.4.1.111.6.1.2.1.23
OraNamesConfigServerName	1.3.6.1.4.1.111.6.1.2.1.24
OraNamesConfigTopologyCheckpointFile	1.3.6.1.4.1.111.6.1.2.1.25
OraNamesConfigTraceDirectory	1.3.6.1.4.1.111.6.1.2.1.26
OraNamesConfigTraceFunc	1.3.6.1.4.1.111.6.1.2.1.27
OraNamesConfigTraceMask	1.3.6.1.4.1.111.6.1.2.1.28
OraNamesConfigTraceUnique	1.3.6.1.4.1.111.6.1.2.1.29

◆ Les variables d'oraNamesServerTable et leurs correspondances IDs :

Noms de variables	Identificateur ID
oraNamesServerQueriesReceived	1.3.6.1.4.1.111.6.1.3.1.1
OraNamesServerLastNnamesNotFound	1.3.6.1.4.1.111.6.1.3.1.2
oraNamesServerQueriesTotalTime	1.3.6.1.4.1.111.6.1.3.1.3
OraNamesServerDeletesReceived	1.3.6.1.4.1.111.6.1.3.1.4
OraNamesServerDeletesRefused	1.3.6.1.4.1.111.6.1.3.1.5
OraNamesServerDeletesTotalTime	1.3.6.1.4.1.111.6.1.3.1.6
OraNamesServerRenamesReceived	1.3.6.1.4.1.111.6.1.3.1.7
OraNamesServerRenamesRefused	1.3.6.1.4.1.111.6.1.3.1.8

OraNamesServerRenamesTotalTime	1.3.6.1.4.1.111.6.1.3.1.9
OraNamesServerUpdatesReceived	1.3.6.1.4.1.111.6.1.3.1.10
OraNamesServerUpdatesRefused	1.3.6.1.4.1.111.6.1.3.1.11
OraNamesServerUpdatesTotalTime	1.3.6.1.4.1.111.6.1.3.1.12
OraNamesServerCorruptMessagesReceived	1.3.6.1.4.1.111.6.1.3.1.13
oraNamesServerResponsesSent	1.3.6.1.4.1.111.6.1.3.1.14
OraNamesServerErrorResponsesSent	1.3.6.1.4.1.111.6.1.3.1.15
OraNamesServerAliasLoopsDetected	1.3.6.1.4.1.111.6.1.3.1.16
OraNamesServerLookupsAttempted	1.3.6.1.4.1.111.6.1.3.1.17
OraNamesServerCreatedOnLookup	1.3.6.1.4.1.111.6.1.3.1.18
OraNamesServerLookupFailures	1.3.6.1.4.1.111.6.1.3.1.19
OraNamesServerExactMatches	1.3.6.1.4.1.111.6.1.3.1.20
OraNamesServerForwardFailures	1.3.6.1.4.1.111.6.1.3.1.21
OraNamesServerForwardTimeouts	1.3.6.1.4.1.111.6.1.3.1.22
OraNamesServerResponsesReceived	1.3.6.1.4.1.111.6.1.3.1.23
OraNamesServerErrorResponsesReceived	1.3.6.1.4.1.111.6.1.3.1.24
OraNamesServerRequestsForwarded	1.3.6.1.4.1.111.6.1.3.1.25
OraNamesServerLastReload	1.3.6.1.4.1.111.6.1.3.1.26
OraNamesServerReloadCheck-Failures	1.3.6.1.4.1.111.6.1.3.1.27
OraNamesServerLastCheckpoint	1.3.6.1.4.1.111.6.1.3.1.28
OraNamesServerName	1.3.6.1.4.1.111.6.1.3.1.29
OraNamesServerAdminRegion	1.3.6.1.4.1.111.6.1.3.1.30

D.4 Interprétation des Variables MIB du RDBMS Public :

- ♦ Les variables de rdbmsDbTable et leurs correspondances IDs :

Noms de variables	Identificateur ID
RdbmsDbIndex	1.3.6.1.2.1.39.1.1.1.1
RdbmsDbPrivateMIBOID	1.3.6.1.2.1.39.1.1.1.2
RdbmsDbVendorName	1.3.6.1.2.1.39.1.1.1.3
RdbmsDbName	1.3.6.1.2.1.39.1.1.1.4
RdbmsDbContact	1.3.6.1.2.1.39.1.1.1.5

- ♦ Les variables de rdbmsDbInfoTable et leurs correspondances IDs :

Noms de variables	Identificateur ID
RdbmsDbInfoProductName	1.3.6.1.2.1.39.1.2.1.1
RdbmsDbInfoVersion	1.3.6.1.2.1.39.1.2.1.2
RdbmsDbInfoSizeUnits	1.3.6.1.2.1.39.1.2.1.3
RdbmsDbInfoSizeAllocated	1.3.6.1.2.1.39.1.2.1.4
RdbmsDbInfoSizeUsed	1.3.6.1.2.1.39.1.2.1.5
RdbmsDbInfoLastBackup	1.3.6.1.2.1.39.1.2.1.6

- ♦ Les variables de rdbmsDbSrvTable et leurs correspondances IDs :

Noms de variables	Identificateur ID
RdbmsSrvPrivateMIBOID	1.3.6.1.2.1.39.1.5.1.1
RdbmsSrvVendorName	1.3.6.1.2.1.39.1.5.1.2
RdbmsSrvProductName	1.3.6.1.2.1.39.1.5.1.3
RdbmsSrvContact	1.3.6.1.2.1.39.1.5.1.4

- ♦ Les variables de rdbmsDbSrvInfoTable et leurs correspondances IDs :

- ◆ Les variables d'oraDBLibraryCacheSumTable et leurs correspondances IDs :

Noms de variables	Identificateur ID
OaDbLibraryCacheSumGets	1.3.6.1.4.1.111.4.1.5.1.1
OaDbLibraryCacheSumGetHits	1.3.6.1.4.1.111.4.1.5.1.2
OaDbLibraryCacheSumPins	1.3.6.1.4.1.111.4.1.5.1.3
OaDbLibraryCacheSumPinHits	1.3.6.1.4.1.111.4.1.5.1.4
OaDbLibraryCacheSumReloads	1.3.6.1.4.1.111.4.1.5.1.5
OaDbLibraryCacheSumInvalidations	1.3.6.1.4.1.111.4.1.5.1.6

- ◆ Les variables d'oraDbSGATable et leurs correspondances IDs :

Noms de variables	Identificateur ID
OaDbSGAFixedSize	1.3.6.1.4.1.111.4.1.6.1.1
OaDbSGAVariableSize	1.3.6.1.4.1.111.4.1.6.1.2
OaDbSGADatabaseBuffers	1.3.6.1.4.1.111.4.1.6.1.3
OaDbSGARedoBuffers	1.3.6.1.4.1.111.4.1.6.1.4

- ◆ Les variables d'oraDbConfigTable et leurs correspondances IDs :

Noms de variables	Identificateur ID
OaDbConfigDbBlockBuffers	1.3.6.1.4.1.111.4.1.7.1.1
OaDbConfigDbBlockCkptBatch	1.3.6.1.4.1.111.4.1.7.1.2
OaDbConfigDbBlockSize	1.3.6.1.4.1.111.4.1.7.1.3
OaDbConfigDbFileSimWrites	1.3.6.1.4.1.111.4.1.7.1.4
OaDbConfigDbMultiBlockReadCount	1.3.6.1.4.1.111.4.1.7.1.5
OaDbConfigDbWriters	1.3.6.1.4.1.111.4.1.7.1.6
OaDbConfigDistLockTimeout	1.3.6.1.4.1.111.4.1.7.1.7
OaDbConfigDistRecoveryConnectHold	1.3.6.1.4.1.111.4.1.7.1.8
OaDbConfigDistTransactions	1.3.6.1.4.1.111.4.1.7.1.9
OaDbConfigLogArchiveBufferSize	1.3.6.1.4.1.111.4.1.7.1.10
OaDbConfigLogArchiveBuffers	1.3.6.1.4.1.111.4.1.7.1.11
OaDbConfigLogBuffer	1.3.6.1.4.1.111.4.1.7.1.12
OaDbConfigLogCheckpointInteval	1.3.6.1.4.1.111.4.1.7.1.13
OaDbConfigLogCheckpointTimeout	1.3.6.1.4.1.111.4.1.7.1.14
OaDbConfigLogFiles	1.3.6.1.4.1.111.4.1.7.1.15
OaDbConfigLogSmallEntryMax	1.3.6.1.4.1.111.4.1.7.1.16
OaDbConfigMaxRollbackSegments	1.3.6.1.4.1.111.4.1.7.1.17
OaDbConfigMaxDispatchers	1.3.6.1.4.1.111.4.1.7.1.18
OaDbConfigMTSMaxServers	1.3.6.1.4.1.111.4.1.7.1.19
OraDbConfigMTSServers	1.3.6.1.4.1.111.4.1.7.1.20
OaDbConfigOpenCursors	1.3.6.1.4.1.111.4.1.7.1.21
OaDbConfigOpenLinks	1.3.6.1.4.1.111.4.1.7.1.22
OaDbConfigOptimizerMode	1.3.6.1.4.1.111.4.1.7.1.23
OaDbConfigProcesses	1.3.6.1.4.1.111.4.1.7.1.24
OaDbConfigSerializable	1.3.6.1.4.1.111.4.1.7.1.25
OaDbConfigSessions	1.3.6.1.4.1.111.4.1.7.1.26
OaDbConfigSharedPool	1.3.6.1.4.1.111.4.1.7.1.27
OaDbConfigSortAreaSize	1.3.6.1.4.1.111.4.1.7.1.28
OaDbConfigSortAreaRetainedSize	1.3.6.1.4.1.111.4.1.7.1.29
OaDbConfigTransactions	1.3.6.1.4.1.111.4.1.7.1.30
OaDbConfigTransactionsPerRollback	1.3.6.1.4.1.111.4.1.7.1.31

- ◆ Les variables d'oraRepTable et leurs correspondances IDs :

Noms de variables	Identificateur ID

Noms de variables	Identificateur ID
RdbmsSrvInfoStartupTime	1.3.6.1.2.1.39.1.6.1.1
RdbmsSrvInfoFinishedTransactions	1.3.6.1.2.1.39.1.6.1.2
RdbmsSrvInfoDiskReads	1.3.6.1.2.1.39.1.6.1.3
RdbmsSrvInfoLogicalReads	1.3.6.1.2.1.39.1.6.1.4
RdbmsSrvInfoDiskWrites	1.3.6.1.2.1.39.1.6.1.5
RdbmsSrvInfoLogicalWrites	1.3.6.1.2.1.39.1.6.1.6
RdbmsSrvInfoPageReads	1.3.6.1.2.1.39.1.6.1.7
RdbmsSrvInfoPageWrites	1.3.6.1.2.1.39.1.6.1.8
RdbmsSrvInfoDiskOutOfSpaces	1.3.6.1.2.1.39.1.6.1.9
RdbmsSrvInfoRequestsHandled	1.3.6.1.2.1.39.1.6.1.10
RdbmsSrvInfoRequestsRecvs	1.3.6.1.2.1.39.1.6.1.11
RdbmsSrvInfoRequestSends	1.3.6.1.2.1.39.1.6.1.12
RdbmsSrvInfoHighwaterIn-boundAssociations	1.3.6.1.2.1.39.1.6.1.13
RdbmsSrvInfoMaxInboundAssociations	1.3.6.1.2.1.39.1.6.1.14

- ♦ Les variables de rdbmsDbSrvParamTable et leurs correspondances IDs :

Noms de variables	Identificateur ID
RdbmsSrvParamName	1.3.6.1.2.1.39.1.7.1.1
RdbmsSrvParamSubIndex	1.3.6.1.2.1.39.1.7.1.2
RdbmsSrvParamID	1.3.6.1.2.1.39.1.7.1.3
RdbmsSrvParamCurrValue	1.3.6.1.2.1.39.1.7.1.4
RdbmsSrvComment	1.3.6.1.2.1.39.1.7.1.5

- ♦ Les variables de rdbmsRelTable et leurs correspondances IDs :

Noms de variables	Identificateur ID
RdbmsRelState	1.3.6.1.2.1.39.1.9.1.1
RdbmsRelActiveTime	1.3.6.1.2.1.39.1.9.1.2

- ♦ Les variables de rdbmsTraps et leurs correspondances IDs :

Noms de variables	Identificateur ID
RdbmsStateChange	1.3.6.1.2.39.2.1.1.1

D.5 Interprétation des Variables MIB de Network Services :

- ♦ Les variables d'applTable et leurs correspondances IDs :

Noms de variables	Identificateur ID
ApplIndex	1.3.6.1.2.1.27.1.1.1
ApplName	1.3.6.1.2.1.27.1.1.2
ApplDirectoryName	1.3.6.1.2.1.27.1.1.3
ApplVersion	1.3.6.1.2.1.27.1.1.4
ApplUptime	1.3.6.1.2.1.27.1.1.5
ApplOperStatus	1.3.6.1.2.1.27.1.1.6
ApplLastChange	1.3.6.1.2.1.27.1.1.7
ApplInboundAssociations	1.3.6.1.2.1.27.1.1.8
ApplOutboundAssociations	1.3.6.1.2.1.27.1.1.9
ApplAccumulatedInboundAssociations	1.3.6.1.2.1.27.1.1.10
ApplAccumulatedOutboundAssociations	1.3.6.1.2.1.27.1.1.11
ApplLastInboundActivity	1.3.6.1.2.1.27.1.1.12
ApplLastOutboundActivity	1.3.6.1.2.1.27.1.1.13
ApplRejectedInboundAssociations	1.3.6.1.2.1.27.1.1.14

App1FailedOutboundAssociations	1.3.6.1.2.1.27.1.1.15
--------------------------------	-----------------------

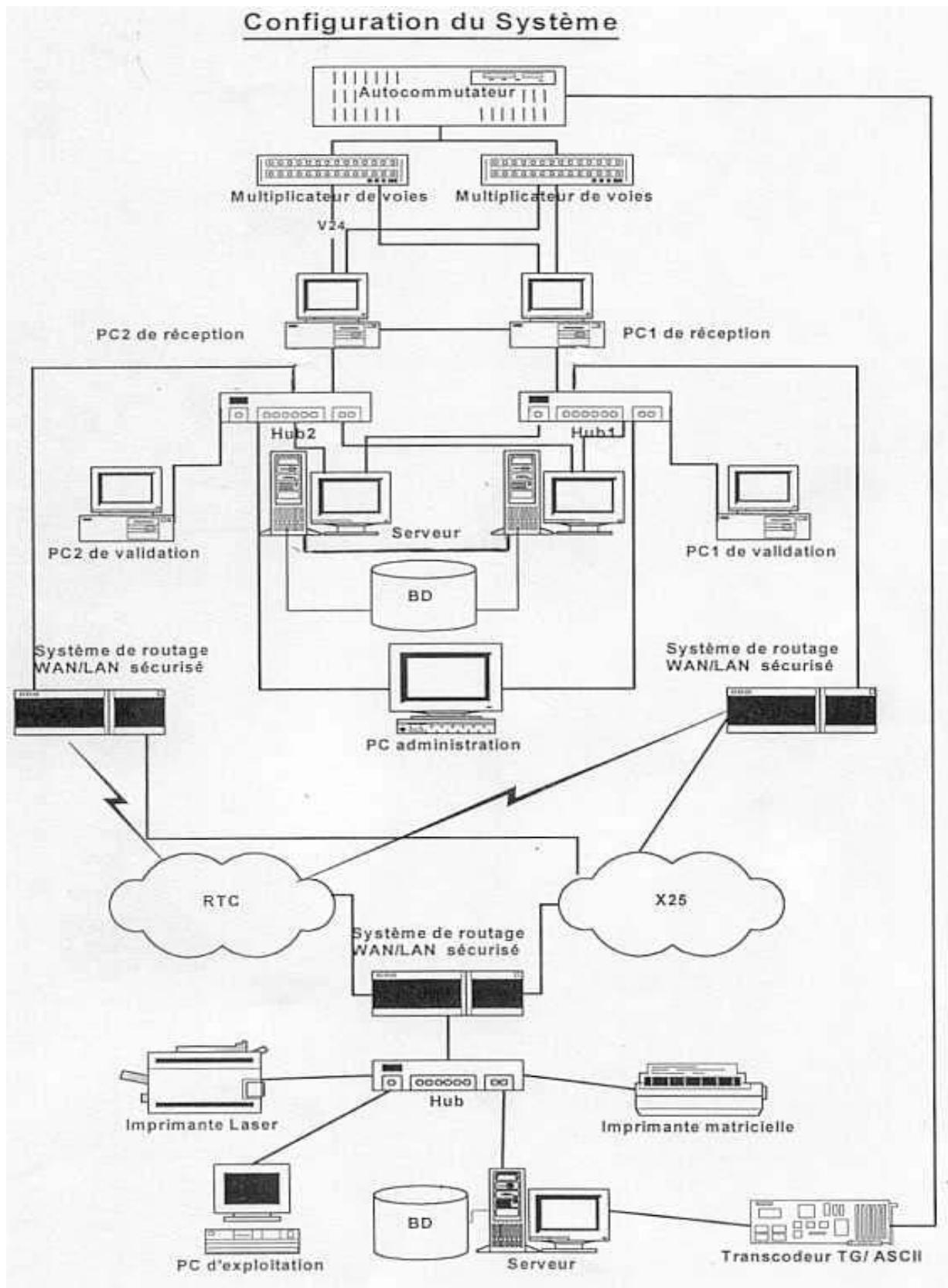
D.6 Interprétation des Variables MIB d'Enterprise Manager :

- ♦ Les variables d'oraAgentEventTable et leurs correspondances IDs :

Noms de variables	Identificateur ID
OraAgentEventIndex	1.3.6.1.4.1.111.12.1.1.1.1
OraAgentEventName	1.3.6.1.4.1.111.12.1.1.1.2
OraAgentEventID	1.3.6.1.4.1.111.12.1.1.1.3
OraAgentEventService	1.3.6.1.4.1.111.12.1.1.1.4
OraAgentEventTime	1.3.6.1.4.1.111.12.1.1.1.5
OraAgentEventSeverity	1.3.6.1.4.1.111.12.1.1.1.6
OraAgentEventUser	1.3.6.1.4.1.111.12.1.1.1.7
OraAgentEventAppID	1.3.6.1.4.1.111.12.1.1.1.8
OraAgentEventMessage	1.3.6.1.4.1.111.12.1.1.1.9
OraAgentEventArguments	1.3.6.1.4.1.111.12.1.1.1.10
OraAgentEventResults	1.3.6.1.4.1.111.12.1.1.1.11

- ♦ Les variables d'oraAgentTraps et leurs correspondances IDs :

Noms de variables	Identificateur ID
OraAgentEventOcc	1.3.6.1.4.1.111.12.2.2



Glossaire

ASN.1	Abstract Syntax Notation One, qui est un langage permettant de définir des types sans ambiguïté. Il se localise au niveau de la couche Présentation du modèle ISO.
BER	Basic Encoding Rules
Client-serveur	Modèle primaire d'interaction dans un système distribué où le programme est situé sur un autre site et attend la réponse. Le programme demandeur est appelé client et le programme qui traite la commande est appelé serveur.
IETF	Internet Engineering Task Force
IEEE	Institute of Electronical and Electronics Engineer
IP	Internet Protocol
ISO	International Standard Organisation
LAN	Local Area Network
MIB	Management Information Base Les objets qui composent la MIB sont organisés suivant une structure arborescente. Ils décrivent l'activité d'un élément du réseau.
NOS	Network Operating System
OSI	Open Systems Interconnection
PDU	Protocol Data Unit
RFC	Request For Comment - Appels à commentaires Série de documents qui contiennent des synthèses, mesures, idées, observations, propositions et normes TCP/IP. Les RFCs sont éditées, mais ne constituent pas des références garanties. Ils sont distribués par le centre d'information réseau (NIC)
RFC	Request For Comment - Appels à commentaires Série de documents qui contiennent des synthèses, mesures, idées, observations, propositions et normes TCP/IP. Les RFCs sont éditées, mais ne constituent pas des références garanties. Ils sont distribués par le centre d'information réseau (NIC)
RMON	Remote network MONitoring
SMI	Structure and identification of Management Information
SNMP	Simple Network Management Protocol Le protocole SNMP est un outil d'administration qui permet de manipuler les objets de la MIB localement ou à travers le réseau.
TCP	Transmission Control Protocol
TNS	Transparent Network Substrate
UDP	User datagram Protocol
WAN	Wide Area Network
OID	Object Identificateur

RFCs

Full Standards	
RFC1155	Structure and Identification of Management Information for TCP/IP-based internets
RFC1157	SNMP
RFC1213	MIB II
Draft Standards	
RFC1212	Concise MIB definitions
Proposed Standards	
RFC1243	Apple Talk MIB
RFC1284	Definitions of Managed Objects for the Ethernet-like Interface Types
RFC1286	Bridges MIB
RFC1317	Definitions of Managed Objects for RS-232-like Hardware Devices
RFC1318	Definitions of Managed Objects for Parallel-printer-like Hardware Devices
1.1 Autres	
RFC1156	Management Information Base for Network Management of TCP/IP-based internets
RFC1214	OSI Internet Management : MIB
RFC1215	A Convention for Defining Traps for use with the SNMP
RFC1231	Token Ring MIB
RFC1270	SNMP Communications Services
RFC1283	SNMP over OSI
RFC1284	Definitions of Managed Objects for the Ethernet-like Interface Types
RFC1285	FDDI Management Information Base
RFC1315	MIB for Frame Relay DTEs
RFC1316	Definitions of Managed Objects for Character Stream Devices
RFC1351	SNMP Administrative Model
RFC1352	SNMP Security Protocols
RFC1353	Definition of Managed Objects for Administration of SNMP Parties
RFC1354	IP forwarding table MIB
RFC1389	RIP version 2 MIB extension
RFC1414	Identification MIB
RFC1441	Introduction to version 2 of the Internet-standard Network Management Framework
RFC1442	Structure of Management Information for SNMPv2
RFC1443	Textual Conventions for SNMPv2
RFC1444	Conformance Statements for SNMPv2
RFC1445	Administrative Model for SNMPv2
RFC1446	Security Protocols for SNMPv2
RFC1447	Party MIB for SNMPv2
RFC1448	Protocols Operations for SNMPv2
RFC1449	Transport Mappings for SNMPv2
RFC1450	Management Information Base for SNMPv2
RFC1451	Manager-to-Manager Management Information Base
RFC1512	FDDI MIB
RFC1513	Token Ring Extensions to the Remote Monitoring MIB

RFC1514	Host Resources MIB
RFC1515	Definitions of Managed Objects
RFC1516	Definitions of Managed Objects for IEEE 802.3 Repeater Devices
RFC1628	UPS MIB
RFC1757	Remote Network Monitoring Management Information Base

CONFIGURATION D'UN AGENT SNMP D'ORACLE ET INTERFACAGE AVEC UN GESTIONNAIRE DE RESEAU

Projet de fin d'études réalisé par :
Garouachi Sami & Labiadh Boulbaba
TS- Télécommunication 2000-2001

Mots clés :

Oracle, SNMP, Variables MIB, Trap, Configuration, Gestionnaire de réseau

Résumé :

La configuration d'un Agent SNMP d'Oracle permet aux différents administrateurs de réseaux la gestion de la MIB Oracle et la consultation de ces variables en envoyant des requêtes ou bien en recevant des alertes sous formes de trap.

La démarche à suivre pour la réalisation de ce travail est :

Au début on doit interpréter les différentes variables MIB d'Oracle et leurs organisation dans des tables, ensuite on doit configurer l'agent SNMP d'Oracle, les fichiers de cette configuration sont Listener.ore, Tnsnames.ora, Master.cfg, Encap.cfg

Enfin avec un gestionnaire de réseau installé sur un poste d'administration, ce gestionnaire capte les traps envoyés par l'agent SNMP pour que l'administrateur puisse faire l'analyse l'intervention dans un temps très court pour assurer le bon fonctionnement de la base .