

Oracle Intelligent Agent

ユーザース・ガイド

リリース 9.2

2002 年 12 月

部品番号 : J06325-02

ORACLE®

Oracle Intelligent Agent ユーザーズ・ガイド, リリース 9.2

部品番号 : J06325-02

原本名 : Oracle Intelligent Agent User's Guide, Release 9.2.0.2

原本部品番号 : A96676-02

Copyright © 1996, 2002, Oracle Corporation. All rights reserved.

Printed in Japan.

制限付権利の説明

プログラム（ソフトウェアおよびドキュメントを含む）の使用、複製または開示は、オラクル社との契約に記された制約条件に従うものとします。著作権、特許権およびその他の知的財産権に関する法律により保護されています。

当プログラムのリバース・エンジニアリング等は禁止されております。

このドキュメントの情報は、予告なしに変更されることがあります。オラクル社は本ドキュメントの無謬性を保証しません。

* オラクル社とは、Oracle Corporation（米国オラクル）または日本オラクル株式会社（日本オラクル）を指します。

危険な用途への使用について

オラクル社製品は、原子力、航空産業、大量輸送、医療あるいはその他の危険が伴うアプリケーションを用途として開発されておりません。オラクル社製品を上述のようなアプリケーションに使用することについての安全確保は、顧客各位の責任と費用により行ってください。万一かかる用途での使用によりクレームや損害が発生いたしましても、日本オラクル株式会社と開発元である Oracle Corporation（米国オラクル）およびその関連会社は一切責任を負いかねます。当プログラムを米国国防総省の米国政府機関に提供する際には、『Restricted Rights』と共に提供してください。この場合次の Notice が適用されます。

Restricted Rights Notice

Programs delivered subject to the DOD FAR Supplement are "commercial computer software" and use, duplication, and disclosure of the Programs, including documentation, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement. Otherwise, Programs delivered subject to the Federal Acquisition Regulations are "restricted computer software" and use, duplication, and disclosure of the Programs shall be subject to the restrictions in FAR 52.227-19, Commercial Computer Software - Restricted Rights (June, 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065.

このドキュメントに記載されているその他の会社名および製品名は、あくまでその製品および会社を識別する目的にのみ使用されており、それぞれの所有者の商標または登録商標です。

目次

はじめに	vii
1 Intelligent Agent の概要	
Oracle Intelligent Agent: 概要	1-2
特長	1-2
SNMP のサポート	1-3
2 インストール、構成および使用方法	
Intelligent Agent のインストール	2-2
Windows NT システムでの Intelligent Agent の運用制御	2-2
Windows NT 上での Intelligent Agent の起動	2-2
Windows NT 上での Intelligent Agent の停止	2-3
ジョブ実行のための Windows NT ユーザー・アカウントの作成	2-3
新規 Windows NT ユーザー・アカウントの作成	2-4
既存の NT ユーザー・アカウントへの権限の割当て	2-4
Windows 2000 の新規ユーザー・アカウントの作成	2-4
ドメイン・ユーザーを Intelligent Agent のユーザーとして構成	2-5
Windows NT および Windows 2000 での SNMP の構成	2-6
UNIX システムでの Intelligent Agent の運用制御	2-7
root.sh シェル・スクリプトの実行	2-7
UNIX プラットフォーム上での Intelligent Agent の起動および停止	2-8
ブラックアウト	2-10
ブラックアウトの定義	2-10
ブラックアウトのコマンドライン・インタフェース	2-10
コマンドラインの例	2-11

UNIX での SNMP の構成	2-12
複数のネットワーク・カード (NIC) で使用するための Intelligent Agent の構成	2-13
複数のネットワーク・カード使用時の Intelligent Agent の動作	2-14
Oracle Intelligent Agent と Oracle Names	2-15
Intelligent Agent に必要なロールとユーザー	2-16
自動検出	2-17
自動検出のための前提条件	2-18
サービス検出プロセス	2-19
Intelligent Agent の検出プロセス (Windows NT)	2-19
Intelligent Agent の検出プロセス (UNIX)	2-20
Real Application Clusters 環境	2-21
リリース 8.0.6、8.1.6 および 8.1.7 の Intelligent Agent のリリース 9.x へのアップグレード	2-21
最適な方法: エージェントのインストールおよび構成	2-22
Agent のインストール	2-22
Intelligent Agent の構成	2-25
Intelligent Agent の互換性	2-26
一般的な Agent-Enterprise Manager のアップグレード・プロセス	2-26

3 ジョブ・スクリプトとイベント・スクリプト

スクリプト言語	3-2
Tcl 言語の説明	3-2
OraTcl の説明	3-4
サーバー・メッセージとエラー情報	3-5
oramsg の要素	3-5
修正ジョブの Tcl 配列に対するイベント	3-8
trigevent の要素	3-8
Intelligent Agent での Tcl の使用	3-10
NLS の問題とエラー・メッセージ	3-11
OraTcl の関数とパラメータ	3-12
共通パラメータ	3-13
convertin	3-14
convertout	3-14
msgtxt	3-15
msgtxt1	3-15
oraautocom	3-16
oracancel	3-16

oraclose	3-17
oracols	3-17
oracommit	3-18
oradbsnmp	3-18
orafail	3-19
orafetch	3-19
orainfo	3-21
orajobstat	3-21
oralogoff	3-22
oralogon	3-22
oralogon_unreached	3-23
oraopen	3-23
oraplexec	3-23
orareadlong	3-24
orareportevent	3-25
oraroll	3-26
orasleep	3-26
orasnmp	3-27
orasql	3-28
orastart	3-29
orastop	3-29
orertime	3-30
orawritelong	3-30

A Intelligent Agent 構成ファイル

構成ファイル	A-2
snmp_ro.ora	A-2
snmp_rw.ora	A-2
services.ora	A-2
ユーザーが構成可能なパラメータ	A-2
Intelligent Agent のログ・ファイル	A-9

B トラブルシューティング

Intelligent Agent のトラブルシューティング	B-2
簡単なチェック	B-2

Windows NT での Intelligent Agent の簡単なチェック	B-2
UNIX での Intelligent Agent の簡単なチェック	B-5
追加のチェック	B-7
TCP/IP の構成と動作の確認	B-8
DNS 名およびコンピュータ名の一致の確認 (Windows NT)	B-10
Oracle Net 構成ファイルの確認	B-10
Oracle Net の動作確認	B-11
Intelligent Agent の ping	B-12
Intelligent Agent への接続のテスト	B-12
Intelligent Agent の正常起動の確認	B-12
Intelligent Agent がノード上の全インスタンスに接続していることの確認	B-14
Intelligent Agent が正しい許可で動作していることの確認 (UNIX)	B-15
オペレーティング・システム・ユーザーが存在し、正しい許可を持っていることの確認 (Windows NT)	B-15
エラーの有無の確認	B-15
Intelligent Agent のエラー・メッセージと解消	B-15
Intelligent Agent 全般	B-16
Windows NT での Intelligent Agent	B-18
UNIX での Intelligent Agent	B-20
リリース 9.x の Intelligent Agent のトレース	B-22
Windows NT/2000 での Intelligent Agent の起動時エラーの理解	B-28
Oracle Intelligent Agent - Windows のイベント・ログ・メッセージ	B-28
UNIX での一般的な Intelligent Agent 設定の問題	B-32
Data Gatherer の理解およびトラブルシューティング	B-34
Data Gatherer のリカバリ能力	B-36
その他の Data Gatherer の問題	B-36
Intelligent Agent の初期化	B-37
UNIX での Intelligent Agent の検出エラーの診断	B-40
UNIX で検出時に使用されるファイル	B-41
UNIX での検出問題の診断	B-42
Window NT での Intelligent Agent の検出エラーの診断	B-46
検出時に使用されるファイル	B-47
NT での検出問題の診断	B-49

用語集

索引

はじめに

このマニュアルの目的

このマニュアルには、Oracle Intelligent Agent の構成情報およびトラブルシューティングに関する重要な疑問への回答が記載されています。『Oracle Intelligent Agent ユーザーズ・ガイド』は、バージョン 2 とリリース 9.x の Oracle Enterprise Manager の利用者、および Intelligent Agent を介して Oracle データベースと通信する、その他のサポート対象システム管理フレームワークの利用者を対象としています。

このマニュアルの対象読者

このマニュアルは、UNIX または Windows NT プラットフォーム上で Oracle Intelligent Agent のインストール、構成またはトラブルシューティングを行う、すべての人を対象としています。ほとんどの状況下において、Oracle Intelligent Agent が必要とする構成およびメンテナンスの手段はごくわずかです。このため、『Oracle Intelligent Agent ユーザーズ・ガイド』は、最初から順番に読むのではなくリファレンスとして使用してください。

このマニュアルの構成

[第 1 章「Intelligent Agent の概要」](#)

Intelligent Agent および検出サービスの機能の概要について説明します。

[第 2 章「インストール、構成および使用方法」](#)

Intelligent Agent のインストールおよび構成の手順について説明します。

[第 3 章「ジョブ・スクリプトとイベント・スクリプト」](#)

Tcl を使用した、ジョブおよびイベントのスクリプト記述について説明します。

[付録 A「Intelligent Agent 構成ファイル」](#)

Oracle Enterprise Manager によって使用される、必要な構成ファイルについて説明します。

付録 B 「トラブルシューティング」

Intelligent Agent のトラブルシューティングのガイドラインを提供し、その手順について説明します。

ドキュメント

Oracle Enterprise Manager リリース 9.2 のドキュメントには、次のものがあります。

- Oracle Enterprise Manager のリリース・ノートには、ソフトウェアの更新その他の最新情報に関する重要な注意や、製品の動作とドキュメントでの記述との違いが記載されています。
- 『Oracle Enterprise Manager 構成ガイド』には、Oracle Enterprise Manager システムの構成に関する情報が記載されています。
- 『Oracle Enterprise Manager 概要』には、Enterprise Manager システムの概要が記載されています。
- 『Oracle Enterprise Manager 管理者ガイド』では、Oracle Enterprise Manager システムのコンポーネントおよび機能が記載されています。
- 『Oracle Intelligent Agent ユーザーズ・ガイド』には、Oracle Intelligent Agent の管理方法が記載されています。
- 『Oracle Enterprise Manager メッセージ・マニュアル』には、Oracle Enterprise Manager でのエラーについて、考えられる原因および推奨される処置が記載されています。

Oracle Enterprise Manager ドキュメントに加えて、Oracle Enterprise Manager のコンポーネントには、詳細なオンライン・ヘルプが用意されています。

リリース・ノートまたはインストール・マニュアルをダウンロードするには、OTN-J (Oracle Technology Network Japan) の次の URL を参照してください。

<http://otn.oracle.co.jp/document/>

関連文書

詳細は、次を参照してください。

- 『Oracle Enterprise Manager 構成ガイド』
- 『Oracle SNMP サポート・リファレンス・ガイド』

リリース・ノート、インストール・マニュアル、ホワイト・ペーパーまたはその他の関連文書は、OTN-J (Oracle Technology Network Japan) に接続すれば、無償でダウンロードできます。OTN-J を使用するには、オンラインでの登録が必要です。次の URL で登録できます。

<http://otn.oracle.co.jp/membership/>

すでに OTN-J のユーザー名およびパスワードを取得済であれば、次の OTN-J Web サイトの文書セクションに直接接続できます。

<http://otn.oracle.co.jp/document/>

マニュアル内にある外部 Web サイトへのリンクの利用について

このマニュアルには、オラクル社が所有または管理していない他の企業や組織の Web サイトへのリンクが記載されている場合があります。オラクル社では、このような Web サイトの利用に関する評価も代弁も行いません。

Intelligent Agent の概要

この章では、Intelligent Agent の概要と特長について簡単に説明します。

- [Oracle Intelligent Agent: 概要](#)
- [特長](#)
- [SNMP のサポート](#)

Oracle Intelligent Agent: 概要

Oracle Intelligent Agent は、ネットワーク内のリモート・ノード上で独立して動作するプロセスです。Intelligent Agent は、サポートするサービスと同じターゲット上に存在し、次の機能を実行します。

- ローカル・サービスを提供する、またはオペレーティング・システム依存のサービスをコールすることにより、管理対象のターゲットとローカルに対話します。
- イベントをチェックし、結果として得られたイベント・レポートを Oracle Enterprise Manager にキューイングします。
- Oracle Enterprise Manager のジョブを実行してその結果および出力を収集し、必要に応じてその結果をキューイングします。
- データ収集を行います。
- コンソールまたはその他のアプリケーションの指示に従って、ジョブまたはイベントを取り消します。
- Intelligent Agent のプラットフォーム上で SNMP がサポートされている場合、イベントの SNMP トラップ送信のリクエストを処理します。

Intelligent Agent の構成の詳細は、ご使用のシステムに適した、プラットフォーム固有の Oracle データベースのインストール・マニュアルを参照してください。

注意： リリース 9.x では、Data Gatherer の機能が Intelligent Agent に統合されたため、独立したアプリケーションではなくなりました。

特長

Intelligent Agent は独立して動作するプログラムであり、その動作の前提としてコンソールまたは Management Server が動作している必要はありません。データベースにサービスする Intelligent Agent は、データベースがダウンしているときも動作できるため、Intelligent Agent によるデータベースの起動またはシャットダウンを行うことができます。Intelligent Agent は、管理者の介入なしに、管理ジョブを常に独立して実行できます。同様に、Intelligent Agent は独自にイベントを検出して、そのイベントに対応できます。このため、Intelligent Agent は、管理者の介入なしにシステムを監視し、修正ジョブを実行して問題を解決できます。

Intelligent Agent はコンソールおよび Management Server とは無関係に稼働し、管理者がコンソールからログアウトしているときでもジョブを実行したり、イベントを監視したりできます。Intelligent Agent は、管理者宛てのすべてのジョブまたはイベント・メッセージをキューイングし、それらを Management Server に通知します。管理者が再びコンソールにログインしたときに、Management Server は保留中のメッセージを現在ログインしている管理者に通知します。それらのファイルには .q の拡張子が付き、`$ORACLE_HOME/network/agent` ディレクトリに格納されます。

ジョブとイベントは Tcl スクリプトとして実装されます。あるイベントに対して Intelligent Agent がジョブまたはテストを実行するとき、Intelligent Agent により適切な Tcl スクリプトが実行されます。

Management Server が、コンソールにログインしている管理者の代理として Intelligent Agent にメッセージを送るとき、管理者の言語およびキャラクタ・セット環境に関する情報も同時に送られます。Intelligent Agent では、管理者の代理としてデータベース管理タスクを実行するとき、NLS 環境情報を使用します。これにより、管理者は自国語でデータベースを管理できます。たとえば、フランスの管理者が、ドイツ語でデータベースを管理し、メッセージをフランス語で受け取ることができます。

Intelligent Agent では、特定のカートリッジを使用して特定の型のデータを収集します。たとえば、オペレーティング・システムまたはデータベースの情報です。Intelligent Agent は、そのノード上で検出されたサービスに関する統計データを収集できます。リクエストがある場合、Oracle Performance Manager および Oracle Capacity Planner とは無関係に収集が開始されます。また、オペレーティング・システムの評価、および Enterprise Manager コンソールで登録されるデータベース固有のイベントに使用される統計データも収集されます。

他の型のデータを収集する場合、インテグレータが独自のカートリッジを記述できます。

SNMP のサポート

Intelligent Agent は SNMP をサポートします。これにより、サード・パーティのシステム管理フレームワークは、SNMP を使用して Intelligent Agent から SNMP トラップを直接受け取ることができます。Intelligent Agent では、Oracle のデータベース MIB 変数にアクセスできます。SNMP をサポートしないプラットフォーム上にデータベースが存在しているときでも、Oracle MIB 変数にアクセスするジョブまたはイベントを送ることができます。SNMP の詳細は、『Oracle SNMP サポート・リファレンス・ガイド』を参照してください。

インストール、構成および使用方法

この章では、Intelligent Agent の基本設定および構成手順について説明します。次の項目について説明します。

- [Intelligent Agent のインストール](#)
- [Windows NT システムでの Intelligent Agent の運用制御](#)
- [Windows NT および Windows 2000 での SNMP の構成](#)
- [UNIX システムでの Intelligent Agent の運用制御](#)
- [UNIX での SNMP の構成](#)
- [複数のネットワーク・カード（NIC）で使用するための Intelligent Agent の構成](#)
- [複数のネットワーク・カード使用時の Intelligent Agent の動作](#)
- [Oracle Intelligent Agent と Oracle Names](#)
- [Intelligent Agent に必要なロールとユーザー](#)
- [自動検出](#)
- [サービス検出プロセス](#)
- [リリース 8.0.6、8.1.6 および 8.1.7 の Intelligent Agent のリリース 9.x へのアップグレード](#)

Intelligent Agent のインストール

Intelligent Agent は Oracle Database および Oracle9i Application Server とともに出荷され、ORACLE_HOME 環境下で管理されるリモート・マシン上にインストールできます。Oracle Universal Installer では、Enterprise Manager のツリー・リストか、データベース・サーバーのツリー・リスト（この場合、データベースおよび Intelligent Agent がインストールされます）のいずれかから Intelligent Agent のインストールを選択できます。Intelligent Agent をスタンドアロンのサービスとしてインストールする場合は、Enterprise Manager のツリー・リストから Intelligent Agent を選択してください。

Windows NT システムでの Intelligent Agent の運用制御

Windows NT システムで Intelligent Agent の運用を制御するには、次の手順に従います。

Windows NT 上での Intelligent Agent の起動

Windows NT 上で Intelligent Agent を起動する手順は、次のとおりです。

1. 「コントロール パネル」フォルダにある「サービス」アイコンをダブルクリックします。
2. 「Oracle<ORACLE_HOME_NAME>Agent」サービスを選択します。

スタートアップの種類が「手動」に設定されている場合、これは、ユーザーが手動で Intelligent Agent を起動するための設定です。システムを起動するたびに Intelligent Agent が自動的に起動されるようにするには、スタートアップの種類を「自動」に設定します。

- a. 「スタートアップ」ボタンをクリックします。「サービス」ダイアログ・ボックスが表示されます。
 - b. 「スタートアップの種類」で「自動」を選択します。
 - c. 「サービス」ダイアログ・ボックスで「OK」をクリックします。
3. 「開始」ボタンをクリックして、Intelligent Agent を起動します。

注意： コマンドラインで次のように入力しても、Intelligent Agent を起動できます。

```
net start oracle<ORACLE_HOME_NAME>agent
```

Intelligent Agent が起動されると、バッチ・ファイル dbsnmpwd.bat が自動的に実行されます。dbsnmpwd とは、Intelligent Agent が停止した場合に Intelligent Agent を自動的に起動する Windows の監視プロセスです。これによって、明示的に停止しないかぎり Intelligent Agent は常に起動状態になります。dbsnmpwd は、バッチ・ファイルを直接編集して構成できます。構成の詳細は、2-8 ページの「[DBSNMPWD](#)」を参照してください。

Windows NT 上での Intelligent Agent の停止

Windows NT 上で Intelligent Agent を停止する手順は、次のとおりです。

1. 「コントロール パネル」フォルダにある「サービス」アイコンをダブルクリックします。
2. 「Oracle<ORACLE_HOME_NAME>agent」サービスを選択します。
3. 「停止」ボタンをクリックして、Intelligent Agent を停止します。

注意： コマンドラインから次のように入力しても、Intelligent Agent を停止できます。

```
net stop oracle<ORACLE_HOME_NAME>agent
```

Intelligent Agent の動作検証

Intelligent Agent が動作していることを検証するには、「コントロール パネル」の「サービス」でそのステータスを調べるか、コマンド・プロンプトで `net start` と入力します。動作しているサービスのリストに「Oracle<ORACLE_HOME_NAME>agent」が表示されます。

「Windows NT タスク マネージャ」で、`dbssnmp` プロセスの情報を確認するという方法もあります。

ジョブ実行のための Windows NT ユーザー・アカウントの作成

管理対象のノード上で Intelligent Agent がジョブを実行するためには、次の条件が満たされている必要があります。

- 高度なユーザー権限、「バッチ ジョブとしてログオン」を持つ Windows NT ユーザー・アカウントが存在すること。権限は既存のローカル・ユーザーまたはドメイン・ユーザー、あるいは新規 Windows NT ユーザーに割り当てることができます。
- Oracle Enterprise Manager コンソールで、ノードに対する優先接続情報リストが前述のユーザーに対して設定されていること。優先接続情報リストの設定の詳細は、『Oracle Enterprise Manager 管理者ガイド』を参照してください。
- Intelligent Agent を起動するユーザーに、TEMP ディレクトリまたは ORACLE_HOME ディレクトリに対する書き込み許可と、`%ORACLE_HOME%\network` ディレクトリに対する読み込み / 書き込み許可が与えられていること。

注意： 「バッチ ジョブとしてログオン」権限が設定されていないと、管理対象のターゲット上でジョブを実行しようとする際に、「ユーザーの認証に失敗しました」というメッセージが表示されます。

- ユーザーがデータベースやリスナーなどの Windows NT サービスを開始および停止するために管理者権限を持っていること。

次の手順のいずれかを実行してください。

新規 Windows NT ユーザー・アカウントの作成

ローカルの Windows NT マシン上で新規 Windows NT ユーザー・アカウントを作成し、そのユーザーに「バッチ ジョブとしてログオン」権限を付与する手順は、次のとおりです。

1. 「管理ツール」プログラム・グループから「ユーザー マネージャ」を選択します。このツールの詳細は、Windows NT のドキュメントを参照してください。
2. 「ユーザー」メニューの「新しいユーザー」を選択して、次の点を確認します。
 - 「ユーザーは次回ログオン時にパスワード変更が必要」オプション・ボックスがチェックされていないこと
 - ユーザー名に SYSTEM または system が使用されていないこと
3. ユーザー・マネージャの「原則」メニューから「ユーザーの権利」を選択します。
4. 「高度なユーザー権利の表示」ボックスをチェックします。
5. 権限のリストから「バッチ ジョブとしてログオン」を選択します。
6. 選択したユーザーにこの権限を付与します。

既存の NT ユーザー・アカウントへの権限の割当て

既存のローカル・ユーザー・アカウントに権限を割り当てる手順は、次のとおりです。

1. 「ユーザー マネージャ」パネル上でユーザーを選択し、次の点を確認します。
 - 「ユーザーは次回ログオン時にパスワード変更が必要」オプション・ボックスがチェックされていないこと
 - ユーザー名に SYSTEM または system が使用されていないこと
2. ユーザー・マネージャの「原則」メニューから「ユーザーの権利」を選択します。
3. 「高度なユーザー権利の表示」ボックスをチェックします。
4. 権限のリストから「バッチ ジョブとしてログオン」を選択します。
5. このユーザーに高度なユーザー権利を追加します。

Windows 2000 の新規ユーザー・アカウントの作成

ローカルの Windows 2000 マシン上で新規ユーザー・アカウントを作成し、そのユーザーに「バッチ ジョブとしてログオン」権限を付与する手順は、次のとおりです。

1. 「コントロール パネル」>「管理ツール」>「ローカル セキュリティ ポリシー」>「ローカル ポリシー」>「ユーザー権利の割り当て」を選択し、右側の画面で「バッチ ジョブとしてログオン」をハイライトします。
2. 「バッチ ジョブとしてログオン」を右クリックし、「セキュリティ」を選択します。
3. 「追加」をクリックし、追加するユーザーを選択します。
4. ユーザーを選択して「追加」をクリックし、「OK」をクリックします。

ドメイン・ユーザーを Intelligent Agent のユーザーとして構成

注意： ドメイン・ユーザーは、リリース 7.3.3 以下の Intelligent Agent ではサポートされていません。

ドメイン・ユーザーを Intelligent Agent のユーザーとして構成する手順は、次のとおりです。

1. ユーザー・マネージャの「原則」メニューから「ユーザーの権利」を選択します。
2. 「高度なユーザー権利の表示」ボックスをチェックします。
3. 権限のリストから「バッチ ジョブとしてログオン」を選択します。
4. 「追加」ボタンをクリックします。
 - a. 「ドメインまたはコンピュータ」フィールドに値を入力します。（ドメインを選択します。）
 - b. 「ユーザーの表示」ボタンをクリックします。
 - c. リスト・ボックスからドメイン・ユーザーを選択します。
 - d. 「追加」をクリックします。
 - e. 「OK」をクリックします。
5. 「ユーザー権利の原則」ダイアログで、「OK」をクリックします。

注意： 同じ名前のローカル・ユーザーとドメイン・ユーザーが存在する場合、ローカル・ユーザーが優先されます。

Windows NT および Windows 2000 での SNMP の構成

Windows NT および Windows 2000 上のサブエージェントをサポートするよう SNMP Master Agent を構成する手順は、次のとおりです。

1. 次のようなサービス・ファイルがあるドライブを選択します。

```
x:\winnt\system32\drivers\etc\services
```

SNMP エントリを次のように変更します。

```
snmp 1161/udp
snmp-trap 1162/udp
```

2. 次のパスにある Peer SNMP Master Agent 構成ファイル、MASTER.CFG を開きます。

```
%ORACLE_HOME%\network\admin
```

この構成ファイルを編集して、次の内容を追加します。

- a. トランスポート・エントリは次のとおりです。

```
TRANSPORT ordinary SNMP
OVER UDP SOCKET
AT PORT 1161
```

- b. SNMP トラップを受け取る必要のあるコミュニティおよびマシン（例：
dlsun1000.us.oracle.com、または IP アドレスを指定する）エントリは、次のとおり
です。

```
COMMUNITY public
ALLOW ALL OPERATIONS
USE NO ENCRYPTION

MANAGER trap_machine.acme.com
SEND ALL TRAPS
WITH COMMUNITY PUBLIC
```

3. 「コントロール パネル」の「サービス」パネルから、Peer SNMP Master Agent および
カプセル化機能を起動します。

カプセル化機能が必要なのは、マシン上に複数のサブエージェントがインストールおよび構成されている場合のみであることに注意してください。Peer SNMP Master Agent 実行可能ファイルは %ORACLE_HOME%\bin\agent.exe、カプセル化機能の実行可能ファイルは %ORACLE_HOME%\bin\encaps.exe です。

4. 「コントロール パネル」の「サービス」パネルからサブエージェント（Oracle Intelligent Agent）を起動します。サブエージェントは、起動するとマスター・エージェントに登録されます。

Oracle の SNMP をテストする場合は、Oracle SNMP Master Agent との通信、および Oracle 固有データの問合せに SNMP を使用する、サード・パーティ製アプリケーションを使用してください。

SNMP をサポートしているサード・パーティ製アプリケーションは、『Oracle SNMP サポート・リファレンス・ガイド』を参照してください。

UNIX システムでの Intelligent Agent の運用制御

UNIX システムで Intelligent Agent の運用を制御するには、次の手順に従います。

root.sh シェル・スクリプトの実行

Intelligent Agent のインストールに成功すると、root.sh の実行を求めるメッセージが Oracle Universal Installer により表示されます。

root.sh は、oratab ファイルの更新または作成を行うシェル・スクリプトです。oratab ファイルは、Intelligent Agent により検出され、Oracle Enterprise Manager により制御される、すべてのデータベースへの参照をユーザーが記述するファイルです。作成されるデータベースごとに、<SID>:<\$ORACLE_HOME>: [Y/N] という形式のエントリがあります。

Intelligent Agent は通常、root.sh により、setuid プログラムとして構成されます。root.sh の実行に成功した場合、setuid root という名前で Intelligent Agent がインストールされます。これにより Intelligent Agent は、ホストに対する優先接続情報リストでその名前とパスワードが与えられるユーザーとして、ジョブを実行できるようになります。

注意： Intelligent Agent が setuid root に設定されることは、root ユーザーが Intelligent Agent を起動することと同じ効果を持つわけではありません。root ユーザーが Intelligent Agent を起動すると、セキュリティ上の問題が生じる可能性があります。setuid プログラムの詳細は、ご使用のプラットフォームのマニュアルを参照してください。

ノード・ジョブを UNIX システムの Intelligent Agent に送るユーザーには、Intelligent Agent の ORACLE_HOME ディレクトリに対する読み込み / 書き込みのアクセス権を付与してください。root.sh に setuid の設定がない場合、Intelligent Agent に送られるすべてのジョブは、Intelligent Agent の実行可能ファイル (dbsnmp.exe) を所有するユーザーの権限のもとで実行されます。root.sh は、Intelligent Agent 上のすべてのジョブに対して、Oracle Enterprise Manager コンソールで優先接続情報リストを設定するようユーザーに強制します。

root.sh の実行成功の検証

root.sh が正しく実行されたことを検証するには、dbsnmp についてファイルの許可を確認します。

1. cd \$ORACLE_HOME/bin と入力します。
- これにより、Intelligent Agent の実行可能ファイルがある、\$ORACLE_HOME/bin にディレクトリが変更されます。
2. ls -al dbsnmp と入力します。
- dbsnmp について、すべての関連詳細がリスト表示されます。

dbsnmp に対する ls -al コマンドの出力は、次のようになります。

```
-rwsr-s---  1 root      g651      1497980 Jun 12 21:04 dbsnmp
```

root は所有者です。dbsnmp は Intelligent Agent の実行可能ファイルです。この例では、グループの名前は g651 です。所有者が root、許可が -rwsr-s--- であれば、root.sh は正しく実行されています。

UNIX プラットフォーム上での Intelligent Agent の起動および停止

UNIX では、Intelligent Agent の起動および停止に agentctl コマンドを使用します。また、dbsnmpwd も実行されます。dbsnmpwd とは、Intelligent Agent が停止した場合に Intelligent Agent を自動的に起動する UNIX の監視スクリプトです。これによって、明示的に停止しないかぎり Intelligent Agent は常に起動状態になります。

関連する agentctl コマンドの一覧を、次の表に示します。

操作	入力するコマンド
Intelligent Agent および dbsnmpwd を起動。	agentctl start agent
Intelligent Agent および dbsnmpwd を停止。	agentctl stop agent
Intelligent Agent のステータスを検証	agentctl status agent
Intelligent Agent を停止してから再起動。このオプションは、Intelligent Agent がすでに稼働中の場合に Intelligent Agent の再起動のみ行います。	agentctl restart agent

DBSNMPWD

dbsnmpwd は、dbsnmp (Intelligent Agent) のプロセスが監視対象のターゲットに常に存在するようにする UNIX の監視スクリプトです。Intelligent Agent が異常終了した、すなわち予期しないリターン・コードで終了した場合、このスクリプトにより Intelligent Agent が再起動されます。

監視スクリプトの動作は、次の環境変数を使用して構成できます。これらの変数は、スクリプト自体の中に記述されています。

注意： 環境変数は、snmp_rw.ora ファイル内では設定しないでください。

DBSNMP_WDLOGFILE: 起動メッセージが記述されるログ・ファイル。デフォルトでは、\$ORACLE_HOME/network/log/dbsnmp.nohup になります。

DBSNMP_RESTART: 自動再起動のメカニズムを無効にする場合、この環境変数を 0 に設定します。デフォルトは 1 です。

DBSNMP_MAX_ABNORMALEXIT および **DBSNMP_TIME_DELTA:** これら 2 つの変数は、Intelligent Agent がスラッシングしている（Intelligent Agent を正常に起動するための初期条件が満たされず、起動直後に停止する状態にある）と想定しても問題のないタイミングを、監視スクリプトが判断する際に機能します。デフォルト値は、それぞれ 3 と 60 です。この場合、Intelligent Agent が 60 秒以内に 4 回以上異常終了すると、Intelligent Agent がスラッシングしていると判断され、監視スクリプトが再起動を行わなくなります。

Intelligent Agent ファイルの暗号化のためのパスワード指定

デフォルトでは、Intelligent Agent が常駐するホストのホスト名を暗号化の鍵またはパスワードとして使用して、Intelligent Agent ファイルが暗号化されます。暗号化のために別のパスワードを指定するには、次の手順で Intelligent Agent を起動します。

```
agentctl start agent password=<new password>
```

例：

```
% agentctl start agent password=mynewpassword
```

ps -efo の実行およびパスワード取得のためのコマンドライン引数の検索を防ぐために、ファイルへのパスワード入力および agentctl へ引数としてファイルを受け渡すオプションがあります。これを使用すると、このファイルは不当なユーザーによる閲覧から保護されます。ファイルの最初の 8 文字がパスワードとして使用されます。使用方法は次のとおりです。

```
% agentctl start agent password_file=<location of password file>
```

例：

```
% agentctl start agent password_file=/u1/myhome/newpassword.txt
```

ブラックアウト

ブラックアウトを使用すると、Enterprise Manager のユーザーは、管理対象である 1 つ以上のターゲットに対する任意またはすべての管理、またはデータ収集アクティビティ（あるいはその両方）を一時停止できます。この機能を利用して、メンテナンスや緊急時の操作を行うことが可能です。

具体的には、ブラックアウトにより次の操作が停止します。

- **イベント**: ターゲットに登録されているすべてのイベントが、ブラックアウトの期間中は評価または起動されません。
- **ジョブ**: ターゲットに発行されたすべてのジョブが、ブラックアウトの期間中はスケジュールまたは実行されません。ブラックアウトの期間中にターゲットに対してスケジュールされる通常間隔のジョブについては、ジョブがスキップされたことを示す通知が Enterprise Manager コンソールに送られます。
- **データ収集**: ターゲットに関する現行の履歴データ収集アクティビティは、すべて停止します。ただし、ブラックアウトの**前に**ターゲットに関して収集されたデータのロードは、データベースが稼働中であるかぎり続行します。新規の収集を発行することはできませんが、ブラックアウトが終了するまで処理は続行されません。

ブラックアウトの定義

ブラックアウトは、ターゲットレベルで作成する、つまり Intelligent Agent が常駐しているノード上で定義する必要があります。ブラックアウトは、コマンドライン・インタフェースで制御できます。ブラックアウトのサブシステムによって、コマンドラインでの要求はすべて、CLI ユーザーと呼ばれる特別なタイプの Intelligent Agent ユーザーに関連付けられます。即時ブラックアウトを設定できるのは、一度に 1 つのみです。ターゲット・ブラックアウトは、同時に複数存在できます。

一度有効になったブラックアウトは変更できません。特定のブラックアウトのステータスを変更する場合は、既存のブラックアウトを削除してから、変更した新規のブラックアウトを再作成してください。

重要: ブラックアウトの取消しは、それを設定したユーザーによってのみ可能です。

ブラックアウトのコマンドライン・インタフェース

Intelligent Agent が常駐するノード上には、ブラックアウトのコマンドライン・ツールがあり、管理者はこれを使用してブラックアウトの設定および取消しを行うことができます。ブラックアウトを設定するには、Intelligent Agent が稼働している必要があります。ユーザー・コマンドは、次のとおりです。

表 2-1 ブラックアウト・コマンド

ブラックアウトの操作	コマンド
ブラックアウトを定義	<code>agentctl start blackout [-d [DD] HH:MM] [<target name>]</code> -d オプションでは、DD HH:MM の形式で期間を指定します。 DD は日数を表します（オプション）。 HH は時間を表します。 MM は分数を表します。
ブラックアウトを削除	<code>agentctl stop blackout [<target name>]</code>
ブラックアウトのステータスを表示（マシン名、ターゲット・タイプおよび期間）	<code>agentctl status blackout</code>
サブシステム（ジョブ、イベント、履歴収集）ごとのブラックアウト	<code>agentctl start blackout <target> -d<uration> -s<ubsystem/s></code> デフォルトでは、すべてのサブシステム（ジョブ、イベント、収集）がブラックアウトされます。

コマンドラインの例

次の例は、状況に応じたブラックアウト・コマンドライン・ユーティリティの使用方法和、その結果を示したものです。

状況例 1: ターゲット vnukal-pc.world 上で、10 分間のブラックアウトを開始する場合。

```
$ agentctl start blackout -d 0:10 vnukal-pc.world
Blackout registered on vnukal-pc.world database
```

状況例 2:（CLI ユーザーにより）ターゲット vnukal-pc.world 上に設定されたブラックアウトを停止する場合。

注意：他の Intelligent Agent ユーザーにより設定されたブラックアウトは、取り消せません。

```
$ agentctl stop blackout vnukal-pc.world
Blackout canceled on vnukal-pc.world database
```

状況例 3: 管理対象のすべてのターゲット上で、無期限のブラックアウトを開始する場合。これは、Intelligent Agent 全体のブラックアウトと同じです。

```
$ agentctl start blackout
Do you wish to blackout the entire agent (Y/N) ? [N] y
All targets on the agent are blacked out.
```

状況例 4: 登録されているすべてのターゲット上で、ブラックアウトを停止する場合。

```
$ agentctl stop blackout
Do you wish to cancel blackout on all targets (Y/N) ? [N] y
Blackout canceled on all targets.
```

状況例 5: あるターゲット上でのブラックアウトのステータスを確認する場合。

```
$ agentctl status blackout
vnukal-pc.world is blacked out. Blackout will end in 2 hours.
```

状況例 6: タイプの異なる別のターゲットと名前が一致するターゲット上で、ブラックアウトを設定する場合。この場合、コマンドライン・インタフェースでは、対話形式で任意のターゲットを選択できます。

```
$ agentctl start blackout payroll
Following targets matching "payroll" have been found.
1. payRoll ( Database )
2. payRoll ( Listener )
Choose the target to blackout [1] : 2
Blackout registered on "payroll" listener
```

状況例 7: ターゲット mytarget 上のイベントを、30 分間ブラックアウトする場合。

```
$ agentctl start blackout mytarget -d 0:30 -s events
```

UNIX での SNMP の構成

UNIX システムでは、Oracle Intelligent Agent (IA) のインストール時に、SNMP ファイルもインストールされます。Enterprise Manager のみの環境を実行している場合、SNMP は不要です。ただし、サード・パーティ製の SNMP 監視システムを実行している場合、SNMP Master Agent または Peer カプセル化機能と通信するには、ご使用のプラットフォームに応じて Intelligent Agent を構成する必要があります。SNMP を構成する手順は、次のとおりです。

1. Intelligent Agent をインストールします。

次のコマンドで、SNMP マスター・エージェント (snmpd プロセス) が稼働していないことを確認します。

```
ps -ef | grep snmp
snmpd (Solaris では snmpd.cfddi) プロセスが稼働している場合は、それを停止します。
```

2. `cd $ORACLE_HOME/network/snmp/peer`
3. `config.master` スクリプトで、同じディレクトリ内の SNMP Console を実行しているマシンの IP アドレスを編集し、`start_peer` スクリプトが存在することを確認します。

4. `su root`
5. `start_peer -a`（Master Peer Agent、カプセル化機能およびシステム固有の SNMP デーモンを起動）
6. root ユーザーからログアウトします。
7. Enterprise Manager コンソールで、イベントを登録し、「外部サービス（エージェントによる SNMP トラップ）への通知を使用可能にする」のチェックボックスをチェックします。トリガーされる各 OEM イベントについて、トラップが SNMP Master コンソールに送られます。

注意：

- SNMP の構成はプラットフォームごとに異なります。各プラットフォームのドキュメントを確認してください（インストールおよび構成のガイドに、この情報が記載されています）。
 - 一部の UNIX プラットフォームでは、SNMP がサポートされていません。そのプラットフォームのドキュメントを確認してください。
-
-

サード・パーティのシステム管理アプリケーションは、SNMP Master Agent を使用して Intelligent Agent と通信します。Oracle Intelligent Agent が SNMP を使用して Master Agent と通信できるようにするには、SNMP Master Agent および Oracle Intelligent Agent を正しく構成する必要があります。

Oracle データベースおよび Management Server に対して SNMP を構成する一般的な手順については、『Oracle SNMP サポート・リファレンス・ガイド』を参照してください。

より広範な構成情報については、ご使用のプラットフォームのインストール・ガイドまたは構成ガイドを参照してください。SNMP の構成は、プラットフォームによって異なります。

複数のネットワーク・カード（NIC）で使用するための Intelligent Agent の構成

リリース 8.1.7 の Intelligent Agent と同様に、リリース 9.x の Intelligent Agent ユーザーは、複数のネットワーク・カードを持つ 1 台のマシン上にある Intelligent Agent の構成において、3 つのオプションを選択できます。デフォルトでは、Intelligent Agent はそのマシン上の主 NIC（UNIX プラットフォームでは `le0`、Windows NT では `network0`）にバインドされます。他の 2 つのオプションは、次のとおりです。

- a. ユーザーが指定した NIC にバインドできます。

- b. マシン上のすべての NIC にバインドできます。このオプションは、すべての NIC 上でその Intelligent Agent にリスニングをさせることが好ましくない場合は、使用しないでください。

注意： Intelligent Agent は、EM ジョブ、イベントおよびデータ収集の実行を求めて送られてくるすべてのリクエストをリスニングするために、IP アドレスにバインドされ、そのアドレスを使用します。

Intelligent Agent では、Intelligent Agent で使用されている IP アドレス /NIC と異なる IP アドレス /NIC でリスニング中のサービス（リスナーなど）を検出することもできるようになります。

複数のネットワーク・カード使用時の Intelligent Agent の動作

リスニング・アドレスが指定されていない場合

snmp_rw.ora に明示的なリスニング・アドレス・ディレクティブがない場合、Intelligent Agent は、Intelligent Agent マシンの主ネットワーク・インタフェース・カードを介した接続をリスニングします。

リスニング IP アドレスが指定されている場合

snmp_rw.ora で明示的なリスニング・アドレスが指定されている場合、Intelligent Agent はそのアドレスでの接続のみをリスニングします。

主ネットワーク・カード以外の特定のネットワーク・インタフェース・カードにリリース 9.x の Intelligent Agent をバインドする手順は、次のとおりです。

1. dbsnmp.hostname パラメータを設定します。

```
dbsnmp.hostname=<IPaddress of the network card>
```
2. コマンドラインで次のように入力し、Intelligent Agent を起動します。

```
>agentctl start agent
```

ホスト名が指定されている場合

snmp_rw.ora でホスト名が指定されている場合、Intelligent Agent はマシンのすべてのネットワーク・インタフェース・カードでの接続をリスニングします。

リリース 8.1.7 より前のリリースの Intelligent Agent（リリース 8.1.5 以上）では、これが Intelligent Agent のデフォルトの動作です（Intelligent Agent で使用するネットワーク・レイヤー・コードのデフォルトの動作です）。

リリース 9.x の **Intelligent Agent** をホスト上のすべてのネットワーク・インタフェース・カードにバインドする手順は、次のとおりです。

1. `dbsnmp.hostname` を設定します。

```
dbsnmp.hostname=<name of host>
```

2. **Intelligent Agent** を起動します。

Windows NT の Fail Safe 構成が使用されている場合

Windows NT の Fail Safe 構成では、**Intelligent Agent** は Fail Safe Agent の Windows NT レジストリに格納されている IP アドレスでの接続をリスニングします。

Intelligent Agent では、ターゲットの構成ファイルで使用されているマシン名または IP アドレスに関係なく、マシン上の各ターゲットを検出します。

Oracle Intelligent Agent と Oracle Names

Oracle Intelligent Agent の管理対象となるマシン上で **Oracle Names** を実行している場合、データベースがすでに **Names Server** で登録されており、データベースの別名が `listener.ora` ファイルの `GLOBAL_DBNAME` パラメータによって定義されていることが想定されます。

Intelligent Agent では、管理対象のサービスを検出する際に **Oracle Names** を使用しません。リスナーがどのデータベースにサービスするかは、`listener.ora` ファイルの `GLOBAL_DBNAME` パラメータによって判断されます。この名前は、管理対象のデータベース名として **Enterprise Manager** コンソールのナビゲータに表示されます。

`GLOBAL_DBNAME` パラメータには一般に、**Names Server** で登録されているものと同一のデータベース名が記述されます。たとえば、データベース初期化パラメータ・ファイルで与えられている、データベースの名前およびドメインが記述されます。`GLOBAL_DBNAME` パラメータの値は一意である必要があります。

この環境でジョブを実行している、またはイベントを監視しているとき、**Intelligent Agent** でのデータベース別名の解決に **Oracle Names** は使用されません。**Intelligent Agent** は **Bequeath** プロトコルを使用して、独自の TNS 接続文字列を生成します。

注意： 同じノード上で複数の **Oracle** データベースを管理しようとしている場合、`listener.ora` ファイルの `GLOBAL_DBNAME` パラメータがデータベースごとに異なっていることを確認してください。

Intelligent Agent に必要なロールとユーザー

Intelligent Agent に対するデフォルト・データベースのユーザー名 / パスワードは、`dbsnmp/dbsnmp` です。`catsnmp.sql` スクリプトが、データベースとともにインストールされます。データベースのロールおよび権限は、`catsnmp.sql` スクリプトを使用して Intelligent Agent ユーザーに割り当てられます。Oracle データベースのインストール時には、`catalog.sql` によって `catsnmp.sql` スクリプトが自動的に実行されます。

Intelligent Agent のデータベース・ログイン用のユーザー名およびパスワードを、ユーザーが変更する必要がある場合があります。ユーザー名およびパスワードを `dbsnmp/dbsnmp` 以外のものに変更するには、`snmp_rw.ora` を編集して、次のパラメータを追加する必要があります。

```
snmp.connect.<svcname>.name = <username>
snmp.connect.<svcname>.password = <password>
```

注意： `<svcname>` は、Enterprise Manager コンソールまたは `snmp_ro.ora` ファイルに表示されるデータベース・サービス名です。

新規の Intelligent Agent ユーザーに必須のロールおよびユーザー権限を付与するには、新規 Intelligent Agent ユーザーを参照する `catsnmp.sql` で指定されている個々の SQL コマンドを実行します。Server Manager または SQL*Plus を使用できます。このタスクを実行するために、`catsnmp.sql` スクリプトを直接編集しないでください。

SNMPAGENT のロールがデータベースに存在するかどうかを確認するには、次の SQL コマンドを入力します。

```
SELECT * FROM dba_roles;
```

SNMPAGENT ロールが表示されない場合、データベースに対して `catsnmp.sql` スクリプトを実行します。

複数のバージョンのデータベースがすでに動作している場合、Intelligent Agent がコンタクトする必要のあるすべての付与およびビューに対して正しい設定を行うには、各データベース上で `catsnmp.sql` スクリプトを実行する必要があります。

スクリプトを実行するには、SYS でログインする必要があります。

注意： `catsnmp.sql` の場所は、動作しているデータベースのバージョンおよびプラットフォームによって異なります。たとえば、Windows NT 上の Oracle9i Database では、スクリプトは次の場所にあります。

```
%ORACLE_HOME%\rdbms\admin
```

自動検出

Intelligent Agent は、ビルトインの自動検出機能を備えています。これは、管理対象のサービスに関する情報が含まれる必要な構成ファイルを、プロセスの起動時に毎回、自動的に生成する機能です。検出プロセスの間に、次の 3 つのファイルが作成または追加されます。

- `$ORACLE_HOME/network/admin/snmp_ro.ora`

`snmp_ro.ora` ファイルは、Intelligent Agent によって作成される読み込み専用ファイルです。このファイルには、Intelligent Agent によって監視されるサービスに関する情報が含まれています。

- `$ORACLE_HOME/network/admin/snmp_rw.ora`

`snmp_rw.ora` ファイルには、管理対象のサービスの索引情報が含まれています。この情報は Intelligent Agent によって内部的に使用されます。また、ユーザーがトレースなどの変数を指定できるようにする働きもあります。

- `$ORACLE_HOME/network/agent/services.ora`

`services.ora` ファイルには、Intelligent Agent が監視する必要のあるすべてのサービスの別名が含まれています。このファイルにリストされたサービスのみが、Intelligent Agent により監視されます。このファイルの内容はその後、検出の間にコンソールに送られます。リリース 9.x では、Intelligent Agent が稼働している環境のオペレーティング・システムの種類およびバージョンに関する情報も、このファイルに含まれます。

注意： これらのファイルで使用されるパラメータの詳細は、[付録 A 「Intelligent Agent 構成ファイル」](#) を参照してください。

Intelligent Agent が起動すると、自動検出プロセスは次のソースから構成パラメータを読み込みます。

- `oratab` (UNIX ノード上)
- Windows NT レジストリ (Windows NT ノード上)
- `listener.ora`
- `tnsnames.ora` (存在する場合)

検出プロセスにより、ノード上にインストールされているサービスが抽出され、前述の構成ファイル群がコンパイルされます。

Intelligent Agent は各 `ORACLE_HOME` に対する SID 情報を、`ORATAB` ファイル (UNIX の場合) または Windows NT レジストリから集めます。Intelligent Agent は次に、`listener.ora` ファイルを解析して、関連する SID およびリスナー情報を探します。`listener.ora` ファイルに `GLOBAL_DBNAME` セクションがある場合、Intelligent Agent はデータベース・サービス名を `GLOBAL_DBNAME` 変数に設定します。変数が存在しない場合、Intelligent Agent は SID に対する有効なサービス名を含む `tnsnames.ora` を、そのマシン上で探します。

該当するファイルを Intelligent Agent が見つけれられない場合、<SID>_<HOSTNAME> というサービス名が各 SID に対して作成されます。

注意： tnsnames.ora で、同じインスタンスに対して複数の別名が存在する場合、Intelligent Agent は最初にリストされている別名を使用します。それ以外の別名を使用するには、tnsnames.ora のエントリの順序を入れ替えて、Intelligent Agent を再起動します。

注意： 1 台のマシン上に複数のデータベース・インスタンスが存在し、listener.ora ファイルの GLOBAL_DBNAME パラメータを使用している場合、これらのインスタンスは listener.ora に一意の GLOBAL_DBNAME を持つ必要があります。listener.ora を手動で編集する必要がある場合もあります。

自動検出のための前提条件

- Oracle Net 構成が存在し、必要なファイル群が Intelligent Agent の起動に先立って作成されている必要があります。必要な SQL*Net（または Oracle Net）ファイルは listener.ora のみですが、特定のサービスを検出するには、tnsnames.ora および sqlnet.ora を正しく構成してください。Intelligent Agent はこれらのファイルを、\$ORACLE_HOME/network/admin ディレクトリで検索します。
- Intelligent Agent による検出における TNS_ADMIN 変数の使用方法は次のとおりです。

（UNIX の場合）すべてのバージョンの UNIX 検出スクリプトで、TNS_ADMIN 変数を使用して入力構成ファイル（listener.ora および tnsnames.ora）の場所を特定できます。この環境変数が設定されている場合、リリース 7.3.4 または 8.0.3 より後のリリースでのみ、出力ファイル（snmp_ro.ora および snmp_rw.ora）を TNS_ADMIN に正しく書き込むことができます。TNS_ADMIN 変数が設定されていない場合、Intelligent Agent は出力ファイルを \$ORACLE_HOME/network/admin ディレクトリに書き込みます。

（Windows NT の場合）リリース 8.0.5 以上では、前述のものに加えて、検出スクリプトにより TNS_ADMIN の値が NT レジストリから読み込まれます。この変数は、次の場所にあります。

 - 「コントロールパネル」>「システム」>「環境」タブ内の TNS_ADMIN 変数
 - Windows NT レジストリ内の TNS_ADMIN キー

サービス検出プロセス

Intelligent Agent が起動時に最初に実行する操作は、Intelligent Agent が監視するノード上にどのようなサービスが存在するかを検出することです。次の検出アルゴリズムに関するドキュメントでは、Intelligent Agent が動作する最も一般的な 2 つのプラットフォームでのサービス検出プロセスについて説明します。

Intelligent Agent の検出プロセス（Windows NT）

Intelligent Agent を起動すると、スクリプトが実行され、Windows NT レジストリ、listener.ora ファイルおよび tnsnames.ora ファイル（存在する場合）から構成パラメータが読み込まれます。

Intelligent Agent は、インストールされているマシン上で新規のサービスを検出し、その構成ファイルである snmp_ro.ora、snmp_rw.ora および services.ora を作成、再書き込みまたは追加します。

Intelligent Agent では、次の検出アルゴリズムを使用して、そのマシン上で使用可能なサービス（Intelligent Agent により管理されるサービス）を判断します。

1. Intelligent Agent は、Windows NT レジストリから読み込まれた各データベース・サービスの ORACLE_SID および ORACLE_HOME 情報を記録します。
2. Windows NT レジストリの値を基に、Intelligent Agent は listener.ora ファイルを読み込み、どのリスナーがどのデータベースにサービスするかを判断します。構成ファイル listener.ora の場所は、Oracle Net の構成ファイルの場所に基づきます。たとえば、環境変数 TNS_ADMIN と、%ORACLE_HOME%\network\ ディレクトリの場所は、Windows NT レジストリから読み込まれた ORACLE_HOME 情報を基にして決まります。
3. 検出されたデータベースの名前は、そのデータベースの listener.ora ファイルで定義されている GLOBAL_DBNAME パラメータに基づきます。
4. listener.ora に GLOBAL_DBNAME パラメータが存在しない場合、Intelligent Agent は listener.ora ファイルを検索したときと同じ方法で tnsnames.ora ファイルを検索します。
5. tnsnames.ora ファイルが見つからない場合、データベース別名 <SID>_<hostnames> がデータベース・サービスに割り当てられます。サービスはこの別名で Intelligent Agent に認識され、Oracle Enterprise Manager コンソールにその名前で表示されます。

注意： TNSNAMES.ORA ファイルで、同じインスタンスに対して複数の別名が存在する場合、Intelligent Agent は最初にリストされている別名を使用します。それ以外の別名を使用するには、TNSNAMES.ORA ファイルのエントリの順序を入れ替えて、Intelligent Agent を再起動します。

Intelligent Agent が存在するノード上に、データベースまたはその他の新しいサービスがインストールされた場合、Intelligent Agent を再起動して、新しいサービスを Intelligent Agent の構成ファイルに追加する必要があります。この手順は、UNIX での Intelligent Agent についても適用されます。

追加サービスの検出

データベース・サービスおよびリスナー・サービスとともに、Intelligent Agent はその他のサービス（コンカレント・マネージャ、Forms Server、Application Server など）も検出します。これらのサービスは、nmiconf.lst ファイルで定義される各検出スクリプトを通して検出されます。nmiconf.tcl は nmiconf.lst を解析して、基本的なサービスのみがサーバーにインストールされていることを確認し、このファイルにリストされているすべての検出スクリプトを実行します。

Intelligent Agent の検出プロセス（UNIX）

起動時に Intelligent Agent は、インストールされているマシン上で新しいサービスを検出し、その構成ファイル snmp_ro.ora、snmp_rw.ora および services.ora を作成します。

Intelligent Agent は次の検出アルゴリズムを使用して、そのマシン上で使用可能なサービス（Intelligent Agent により管理されるサービス）を判断します。

1. Intelligent Agent は oratab ファイルから、すべての Oracle ホームおよび SID の値を読み込みます。oratab ファイルが次のどちらの場所に格納されるかは、プラットフォームによって異なります。
 - /etc
 - /var/opt/oracle
2. Intelligent Agent は、次の要領で listener.ora ファイルを検索します。
 - 環境変数 tns_admin が設定されている場合、listener.ora ファイルはこのディレクトリ内で検索されます。
 - 環境変数 tns_admin が設定されていない場合、listener.ora ファイルは /etc または /var/opt/oracle 内で検索されます。
 - どちらのディレクトリ（/etc または /var/opt/oracle）も存在しない場合、listener.ora ファイルは oratab ファイルに一覧で記載されている ORACLE_HOME の /network/admin ディレクトリ内で検索されます。

3. 検出されたデータベースの名前は、そのデータベースの listener.ora ファイルで定義されている GLOBAL_DBNAME パラメータに基づきます。
4. listener.ora に GLOBAL_DBNAME パラメータが存在しない場合、Intelligent Agent は listener.ora ファイルを検索したときと同じ方法で tnsnames.ora ファイルを検索します。
5. tnsnames.ora ファイルが見つからない場合、データベース別名 <SID>_<hostnames> がデータベース・サービスに割り当てられます。サービスはこの別名で Intelligent Agent に認識され、Oracle Enterprise Manager コンソールにその名前で表示されます。

Real Application Clusters 環境

Real Application Clusters 環境では、Intelligent Agent（各ノードに 1 つ）の複数のインスタンスは 1 つの ORACLE_HOME を共有する必要があります。この構成をサポートするには、そのエージェントによって作成されるファイル（キュー・ファイル、ログおよびトレース、CS リカバリ・ファイル、構成ファイル）を、ORACLE_HOME の外側の代替のディレクトリに格納します。

デフォルトの ORACLE_HOME の位置をオーバーライドするには、環境変数 ORA_OEMAGENT_DIR を ORACLE_HOME の外側の新しいディレクトリに設定します。

例（UNIX 環境の場合）：

```
setenv ORA_OEMAGENT_DIR /private/agentstate
```

リリース 8.0.6、8.1.6 および 8.1.7 の Intelligent Agent のリリース 9.x へのアップグレード

各リリースの Intelligent Agent により、Agent のパフォーマンス、機能性および信頼性が向上します。そのため、ご使用のプラットフォームで利用できる最新バージョンに Intelligent Agent をアップグレードすることをお薦めします。より新しい Intelligent Agent に移行する際、Enterprise Manager の既存のジョブ、イベントおよびデータ収集を維持するには、\$ORACLE_HOME/bin にある NMUMIGR8 ユーティリティを使用します。このユーティリティでは、既存のジョブ、イベントおよびデータ収集を、リリース 9.x の Intelligent Agent で認識されるフォーマットに移行できます。

使用方法

```
nmumigr8 [-source_home <source ORACLE_HOME>] [-verbose]
```

パラメータ

-source_home

既存の Intelligent Agent キューおよびデータ収集ファイルがあるソース Oracle ホーム。
source_home を指定しない場合は、環境変数 ORACLE_HOME の値である宛先 Oracle ホームになります。

-verbose

Intelligent Agent の \$ORACLE_HOME/network/trace ディレクトリにあるトレース・ファイル nmumigr8.trc に、詳細な移行情報を書き込むかどうかを指定するフラグ。このフラグを設定しない場合は、要約情報のみがトレース・ファイルに書き込まれます。

旧リリースの Intelligent Agent からリリース 9.x の Intelligent Agent にアップグレードするユーザーは、次の項で説明するガイドラインに従ってください。

最適な方法 : エージェントのインストールおよび構成

1 つのマシンに、異なるコア・バージョンのデータベース製品を複数インストールでき、これらに対して個別にアップグレードおよびパッチを実行することができます。Intelligent Agent は複数の製品を管理できるため、Agent の実装、アップグレードおよびメンテナンスを最適な方法で実行することにより、Agent のパフォーマンスを最適化し、トラブルのないスムーズな操作を保証します。

Agent のインストール

1 つのマシンに 1 つの Agent のインストール

1 つのマシンに Intelligent Agent の複数のバージョンをインストールすることも可能ですが、アクティブにできるバージョンは常に 1 つのみです。このため、1 つのマシンに 1 つのバージョンのみの Intelligent Agent をインストールすることをお勧めします。1 つのマシンに複数のバージョンの Intelligent Agent が存在すると、間違った Agent を起動して、すべてのジョブおよびイベントの登録と通知を正しくないバージョンの Agent に送信してしまう可能性があります。これは、Management Server および Intelligent Agent 間の同期の問題を引き起こします。この問題に対する唯一の解決方法は、ナビゲータからノードを削除し、情報を再同期するために Intelligent Agent を完全起動することです。詳細は、B-37 ページの「Intelligent Agent の初期化」を参照してください。

異なる Agent のユーザーの作成

Agent をインストールおよび制御（起動および停止）するオペレーティング・システム・レベルのユーザーを作成します。ユーザー名は、管理されているノード上ですべて同じである必要があります（例、oramon）。パスワードはこのかぎりではありません。Agent の所有者は、ノード上で監視されているすべてのデータベースに対する dba グループのメンバーである必要がありますが、そのプライマリ・グループである必要はありません。

Agent の所有者のプロファイルでは、次の環境変数を設定します。リスト 1 の例では、csh を Agent 所有者のデフォルト・シェルであると仮定し、.cshrc ファイルの内容をリストしています。Bourne Shell または Korn Shell が Agent 所有者のデフォルト・シェルとして使用されている場合、.profile ファイルで同様の構造を使用します。

例 2-1 ORAMON オペレーティング・システム・ユーザーのプロファイル

```
umask 002
# provide the appropriate value for ORACLE_HOME
setenv ORACLE_HOME /oracle
setenv PATH ${ORACLE_HOME}/bin:/usr/ccs/bin:${PATH}
setenv TCL_LIBRARY ${ORACLE_HOME}/network/agent/tcl

# unset ORACLE_HOME, LD_LIBRARY_PATH and NLS_LANG
unsetenv ORACLE_SID
unsetenv LD_LIBRARY_PATH
unsetenv NLS_LANG

# if a central location for net8 config files exist,
# set it here otherwise, make sure TNS_ADMIN is not set
# setenv TNS_ADMIN /usr/local/oracle/admin
unsetenv TNS_ADMIN

# point ORATAB to an OEM-specific oratab file if such exists
if ( -e ${ORACLE_HOME}/network/agent/oratab.oem ) then
    setenv ORATAB ${ORACLE_HOME}/network/agent/oratab.oem
endif
```

環境変数 TNS_ADMIN は、Oracle Net 構成ファイルの集まっている場所を指している必要があります（その場所が存在する場合）。存在しない場合、TNS_ADMIN 変数を設定しないようにします。

独自の ORACLE_HOME への Intelligent Agent のインストール

様々なソフトウェアのメンテナンス・アップグレードの際に起こる競合を回避し、各製品を個別にアップグレードできるようにするため、Intelligent Agent を独自の ORACLE_HOME にインストールすることをお勧めします。

さらに、使用環境のすべてのマシンに Agent をインストールする際には、標準オペレーティング・システム・ユーザーである oramon を使用することをお勧めします。ディレクトリの所有者、およびそのディレクトリが属しているグループの所有者の両方を変更する必要があります。これを行うには、次の操作を実行します。

1. 新しい ORACLE_HOME ディレクトリを作成します。

```
> mkdir oemagent_9.2.0
```

2. 所有者を oramon に変更します。

```
> chown oramon oemagent_9.2.0
```
3. グループを oramon に変更します。

```
> chgrp oramon oemagent_9.2.0
```

Oracle 8.1.5 以降では、Oracle ソフトウェアのインストールには Universal Installer を使用しています。Universal Installer を使用して Intelligent Agent をインストールするには、次の手順に従います。

1. Installer の「ファイルの場所」ウィンドウで、新しい ORACLE_HOME 名および物理的な場所を指定します。
2. 「カスタム」インストール・タイプを実行します。
3. 製品リストで次の製品を選択します。

Oracle Enterprise Manager Products -> Oracle Intelligent Agent

4. 他のすべての製品を選択解除します。製品リストのすべてのエレメントを拡張し、Intelligent Agent のみがインストール対象になっていることを確認します。製品のトップ・レベルのエントリのみを選択解除しても、それ以下のレベルのエレメントが選択解除されているとはかぎりません。
5. インストールを終了し、オペレーティング・システム固有のインストール後のタスク (UNIX での root.sh スクリプトなど) を実行します。

注意： Oracle データベース・サーバーを持つ Oracle ホームに Intelligent Agent をインストールする場合、Agent のバージョンがデータベースのバージョンに適合していることを確認します。Intelligent Agent のインストール中にデータベースをシャットダウンする必要はありません。Agent を削除する場合、これによって Oracle データベース・サーバーが依存しているコンポーネントが削除されることはありません。

Intelligent Agent の Agent Home へのシンボリック・リンクの使用

すべてのマシンで標準の Agent の ORACLE_HOME を使用する必要があり、ORACLE_HOME は現在の Agent の場所へのシンボリック・リンクになります。このように、Agent が他のバージョンにアップグレードされる際、リンクは最新の Agent の場所を示します。これにより、各マシンで常に Agent の一定の ORACLE_HOME を維持できます。

たとえば、Intelligent Agent のリリース 9.2.0 を使用する場合、ディレクトリ /u01/app/oracle/product/oemagent_9.2.0 へのシンボリック・リンクであるディレクトリ /u01/app/oracle/product/oemagent を指すように ORACLE_HOME を設定します。

```
ln -s /u01/app/oracle/product/oemagent /u01/app/oracle/product/oemagent_9.2.0
```


Intelligent Agent の構成

未使用の Intelligent Agent の無効化

古い Intelligent Agent を誤って起動することを防ぐために、Agent のすべての古いバージョンの DBSNMP の実行可能ファイルの名前を変更します。これにより、正しくない Intelligent Agent を誤って起動することを防ぎます。

マシン起動時の Agent の自動起動

マシンの起動および再起動時に、Agent が自動的に起動するように設定する必要があります。次にリストされているようなスクリプトの 1 つをブート・アップ・プロシージャに含め、マシンの起動時に Intelligent Agent が自動的に起動するようにします。

例 2-2 サンプルの Agent ブート時のスタートアップ・スクリプト (UNIX)

```
#!/bin/sh

PATH=/usr/bin:/usr/sbin:/sbin; export PATH

# modify the next line to point to the agent's ORACLE_HOME
ORACLE_HOME="/u01/app/oracle/product/oemagent"

# find the agent owner by looking at file ownership
USERID=`ls -ld $ORACLE_HOME/bin | awk '{print $3}'`

case $1 in
'start'| '')
    su - $USERID -c "$ORACLE_HOME/scripts/oemagent start"
    "Starting OEM agent"
    ;;
'stop')
    su - $USERID -c "$ORACLE_HOME/scripts/oemagent stop"
    "Stopping OEM agent"
    ;;
*)
    echo "usage: $0 {start|stop}"
    ;;
esac
```

Intelligent Agent の互換性

Intelligent Agent、Enterprise Manager、RDBMS 間の互換性を保持し、Enterprise Manager フレームワークの特定のバージョンで利用可能なすべての機能にアクセスするため、フレームワークのバージョンに適合する Intelligent Agent を使用することをお勧めします。次の表では、Agent のバージョンの互換性を示しています。

表 2-2 Intelligent Agent の互換性

データベースのバージョン	Enterprise Manager のバージョン	Intelligent Agent のバージョン
7.3.4.x	1.3.4	7.3.4.x
8.0.5.x、7.3.4.x	1.6.0	8.0.5.x
8.0.6.x、8.0.5.x、7.3.4.x	1.6.5	8.0.6.x
8.1.5.x、8.0.6.x、8.0.5.x、7.3.4.x	2.0.4	8.1.5.x
8.1.6.x、8.1.5.x、8.0.6.x、8.0.5.x、7.3.4.x	2.1	8.1.6.x
8.1.7.x、8.1.6.x、8.1.5.x、8.0.6.x、8.0.5.x、7.3.4.x	2.2	8.1.7.x
9.2.0、9.0.1、8.1.7.x	9.2.0	9.2.0

Intelligent Agent のインストールの完了後、Intelligent Agent の SQL*Net/Oracle Net ファイル (SQLNET.ORA および TNSNAMES.ORA) を作成 / 更新して、Agent が管理する必要のあるすべてのデータベースに接続できるようにする必要があります。さらに、Intelligent Agent はデフォルトでそのマシンの ORATAB ファイルを使用して (環境変数 TNS_ADMIN が設定されていないかぎり)、ORACLE_HOME を検出してサービスを見つけます。Intelligent Agent で ORACLE_HOME のサブセットを検出し、対応するサービスを見つけようとする場合、このサブセットを含む別のファイルを作成することをお勧めします。Intelligent Agent を起動する前に、環境変数 ORATAB を Agent の環境で設定し、このファイルを指すようにする必要があります。

一般的な Agent-Enterprise Manager のアップグレード・プロセス

1. 最新の Intelligent Agent を新しい Oracle ホームの下にインストールします。「最適な方法 : エージェントのインストールおよび構成」を参照してください。
2. 保存しておくジョブまたはイベントが、それぞれジョブ・ライブラリまたはイベント・ライブラリに保存されていることを確認します。ジョブ / イベントをジョブ / イベント・ライブラリに追加するには、ジョブ / イベント・ペインから該当のジョブ / イベントを選択し、マウスの右ボタンを使用して目的のエントリをクリックして、コンテキスト依存メニューから「ジョブのライブラリへのコピー」を選択します。

3. イベントのアラートをイベントの履歴に移動します。履歴ペインの内容は保存または消去できます。

注意： 複数のターゲットに対して登録されているイベントがある場合、「類似作成」メニュー・オプションを使用して、各ターゲットについて個々のイベントを作成し、これらのイベントをイベント・ライブラリに保存します。

4. Enterprise Manager コンソールから、既存のイベントの登録を解除し、Intelligent Agent をアップグレードするノードに対してスケジュールされているアクティブ・ジョブを削除します。
5. 古い Intelligent Agent をシャットダウンします。
6. 新しい Intelligent Agent を起動します。
7. OEM コンソールから、ナビゲータ上のノードをリフレッシュします。
8. 新しい Intelligent Agent に保存したジョブおよびイベントを再送します。

ジョブ・スクリプトとイベント・スクリプト

この章では、次の項目について説明します。

- スクリプト言語
- サーバー・メッセージとエラー情報
- 修正ジョブの Tcl 配列に対するイベント
- Intelligent Agent での Tcl の使用
- NLS の問題とエラー・メッセージ
- OraTcl の関数とパラメータ

スクリプト言語

ジョブ・スクリプトおよびイベント・スクリプトの記述には、**OraTcl** 拡張機能付き **Tcl** 言語を使用します。**Tcl** は次の要件を満たすことから、スクリプト用の言語として使用されます。

- ファイルおよびデバイスの処理、プログラムの起動、およびオペレーティング・システム関数の実行のためのホスト・システム・アクセス
- RDBMS アクセスのための SQL および PL/SQL 関数
- RDBMS 管理用関数
- **Intelligent Agent** 自体がサポートするデータベース **MIB** 変数、および、ホストのサービスまたは **SNMP** 上で実現されるその他のサービスなどの外部 **MIB** の両方への **SNMP** アクセス
- **Oracle Intelligent Agent**、および **Oracle Trace** などその他の **Oracle** ソフトウェアとの通信
- 次のことを実現するジョブ・スクリプトおよびイベント・スクリプトを記述するための構文
 - ユーザー・インタフェースの駆動
 - ジョブまたはイベントの性質、および任意の入出力に関する情報の提供
 - グローバリゼーション・サポートのための **Oracle** メッセージ・ファイル・システムへのアクセス

Tcl 言語の説明

Tcl は、カリフォルニア州立大学バークレー校の **John Ousterhout** 博士によって開発されました。**Tcl** は **Tool Command Language** の略称で、現行のリリースのバージョンは 7.5 です。

Tcl は言語とライブラリの両方を指します。**Tcl** はシンプルなテキスト言語で、テキスト・エディタ、デバッガ、イラスト作成ツール、シェルなどの対話形式プログラムにコマンドを発行することがその主な用途です。**Tcl** はシンプルな構文を持ち、言語の拡張が容易です。**Tcl** の利用者は、独自のコマンド・プロシージャを記述して、ビルトイン・セットに備わっているものよりもさらに強力なコマンドを組み込むことができます。

Tcl は、アプリケーション・プログラムに埋込み可能なライブラリ・パッケージでもあります。**Tcl** ライブラリは、**Tcl** 言語のパパーサー、**Tcl** のビルトイン関数を実装するルーチン、および各アプリケーションに固有の追加コマンドで **Tcl** を拡張するためのプロシージャから構成されます。アプリケーション・プログラムは **Tcl** コマンドを生成し、実行のためにそれらを **Tcl** パパーサーに渡します。コマンドは、入力ソースから文字を読み込むことによって生成できます。また、アプリケーションのメニュー・エントリ、ボタン、キーストロークなどのユーザー・インタフェースに、コマンド文字列を関連付けることによって生成できます。**Tcl** ライブラリはコマンドを受け取ると、コマンドをコンポーネント・フィールドに解析し、ビルトイン・コマンドを直接実行します。アプリケーションによって独自に実装されるコマンドについては、**Tcl** はアプリケーションにコール・バックしてコマンドを実行します。多

くの場合、コマンドに実行のための追加の文字列を渡すことによって、Tcl インタプリタが再帰的に起動されます。プロシージャ、ループ・コマンドおよび条件コマンドは、すべてこのようなくみで動作します。

アプリケーション・プログラムのコマンド言語に Tcl を使用することには、様々な利点があります。

- Tcl は標準的な構文を提供します。Tcl をいったん学習すれば、Tcl ベースのアプリケーションに容易にコマンドを送ることができます。
- Tcl は言語拡張性を備えています。すべての Tcl アプリケーションに必要なのは、アプリケーションに固有の低レベル・コマンドをいくつか実装することです。Tcl は多くのユーティリティ・コマンドに加えて、複雑なコマンド・プロシージャを構築するための一般的なプログラミング・インタフェースを提供します。Tcl を使用すれば、これらの機能をアプリケーションの側で繰り返し実装する必要はありません。
- Tcl の拡張機能は、Tcl コマンドの送受信によってアプリケーション間で通信を行うための機構を提供します。Tcl 言語の共通的なフレームワークは、アプリケーション間の通信を容易にします。

Tcl はもともと、大規模なソフトウェア・システムの設計には複数の言語を使用する必要がある、という思想のもとに設計されたものです。これは、複雑な内部データ構造を操作する、パフォーマンスを重要視するなどの各用途に応じて最適な言語を使い分けることを意味します。Tcl については、C プログラムの断片をつなぎ合わせたり、拡張のためのフックを提供したりするための短いスクリプトを記述する、などの用途があります。Tcl スクリプトの場合、パフォーマンスや、複雑なデータ構造およびアルゴリズムを扱うための機能性よりも、学習、プログラミングおよび統合の簡便さが重視されます。Tcl は、より低いレベルでの実現に適したタスクに遭遇した際に、それをより低レベルの言語に移植する容易さを考慮して設計されています。このように、基本的なコア機能は小さなまま保つことができ、特定の目的または要求に必要な断片を持ち寄って組み合わせる作業に集中することができます。Tcl/Tk の詳細は、次の Web サイトを参照してください。

- <http://www.sun.com/960710/cover/testimonials.html>
- <http://www.neosoft.com/tcl>
- <http://www.scriptics.com/resource/doc/papers/>

注意： Web サイトの場所は頻繁に変化するため、これらのアドレスには将来アクセスできなくなる可能性があります。

OraTcl の説明

Intelligent Agent のジョブとイベント・スクリプトはともに、ファイルとデバイスの処理、プログラムの起動、オペレーティング・システム関数の実行、および Oracle データベースへのアクセスのためのホスト・システム・アクセスを必要とします。OraTcl は、Oracle の用途および SNMP アクセスのために Tcl を拡張する目的で開発されました。OraTcl の機能のカテゴリは、次のように分かれています。

- SQL および PL/SQL 関数
- RDBMS 管理用関数
- SNMP アクセス
- Intelligent Agent およびその他の Oracle ソフトウェアとの通信
- キャラクタ・セット変換およびエラー処理用の関数
- 汎用ユーティリティ関数

OraTcl の関数および変数の説明は、3-12 ページの「[OraTcl の関数とパラメータ](#)」を参照してください。

例： OraTcl スクリプト

次の例は、OraTcl の基本的な使用方法を示しています。

```
#
# monthly_pay.Tcl
#
# usage: monthly_pay.Tcl [connect_string]
# or    Tcl -f monthly_pay.Tcl [connect_string]
#
# sample program for OraTcl
# Tom Poindexter
#
# example of sql, pl/sql, multiple cursors
# uses Oracle demo table SCOTT.EMP
# uses id/pass from command line,
# or "scott/tiger" if not specified
#
# this example does not illustrate efficient sql!
# a simple report is produced of the monthly payroll
# for each jobclass
#
global oramsg
set find_jobs_sql { select distinct job from SCOTT.EMP }
set monthly_pay_pl {
begin
select sum(sal) into :monthly
```



```

from SCOTT.EMP
where job like :jobclass;
end;
}
set idpass $argv
if {[string length $idpass] == 0} {
    set idpass "scott/tiger"
}
set lda [oralogon $idpass]
set cur1 [oraopen $lda]
set cur2 [oraopen $lda]
orasql $cur1 $find_jobs_sql
set job [orafetch $cur1]
while {$oramsmsg(rc) == 0} {
    set total_for_job [lindex [oraplexec $cur2 $monthly_pay_pl :monthly ""
:jobclass "$job"] 0]
    puts stdout "Total monthly salary for job class $job = ¥$$total_for_job"
    set job [orafetch $cur1]
}
oraclose $cur1
oraclose $cur2
oralogoff $lda
exit

```

サーバー・メッセージとエラー情報

OraTcl は、Tcl グローバル配列 `oramsmsg` を作成、維持して、Oracle データベース・メッセージのフィードバックを提供します。`oramsmsg` は、OraTcl インタフェース・ルーチンと通信して、NULL の戻り値と LONG の制限を指定するためにも使用されます。NULLVALUE と MAXLONG を除くすべての場合、任意の OraTcl コマンドの起動時に各要素が NULL にリセットされ、コマンドの影響を受ける任意の要素が設定されます。`oramsmsg` 配列は、オープンされているすべての OraTcl ハンドルの間で共有されます。

注意： `oramsmsg` は、それを必要とする任意の Tcl プロシージャの中で、グローバル文を使用して定義してください。

oramsmsg の要素

`oramsmsg` の要素を次に示します。

oramsmsg (agent_characterstet)

Intelligent Agent のキャラクタ・セットで、US7ASCII のような値をとります。`convertin` および `convertout` 関数によるキャラクタ・セットの変換に使用されます。3-14 ページの「[convertin](#)」および 3-14 ページの「[convertout](#)」を参照してください。

oramsg (db_characterset)

データベースのキャラクタ・セットで、US7ASCII などの値をとります。convertin および convertout 関数によるキャラクタ・セットの変換に使用されます。3-14 ページの「convertin」および 3-14 ページの「convertout」を参照してください。この変数は、Tcl スクリプト内では oralogon 関数の後でのみ使用できます。

oramsg (collengths)

oracols によって返される列の長さの Tcl リストです。collengths は、oracols によってのみ設定されます。

oramsg (colprecs)

oracols によって返される数値列の精度の Tcl リストです。colprecs は、oracols によってのみ設定されます。数値以外の列については、リスト・エントリは NULL 文字列になります。

oramsg (colscopes)

oracols によって返される数値列のスケールの Tcl リストです。colscopes は、oracols によってのみ設定されます。数値以外の列については、リスト・エントリは NULL 文字列になります。

oramsg (coltypes)

oracols によって返される列の型の Tcl リストです。coltypes は、oracols によってのみ設定されます。返される型には、CHAR、VARCHAR2 (バージョン 7)、NUMBER、LONG、rowid、DATE、RAW、LONG_RAW、MLSLABEL、RAW_MLSLABEL または不明の型などがあります。

oramsg (errortxt)

rc に関連付けられるメッセージ・テキストです。oraplexec 関数によって複数の SQL 文が実行されることがあるため、サーバーから複数のメッセージが受け取られる可能性があります。

oramsg (handle)

最後に実行された OraTcl 関数のハンドルを指します。ハンドルは、コマンドを追跡するために使用されるメモリー内でのマッピングのことで、無効なハンドルが使用されている場合を除き、OraTcl コマンドの実行のたびに設定されます。

oramsg (jobid)

現行のジョブのジョブ ID です。ジョブ・スクリプトに対してのみ定義されます。

oramsg (language)

コンソールの NLS 言語で、AMERICAN_AMERICA.US7ASCII のように設定します。

oramsg (maxlong)

プログラマが設定できる要素で、orafetch によって返される LONG または LONG RAW データの量を制限します。デフォルトは 32KB です。上限値は 64KB (バージョン 6) または

2147483647 バイト (バージョン 7) です。ゼロ以下の値はすべて無視されます。変更された `maxlong` の値は、次に `orasql` をコールしたときに有効になります。`orafetch` の、`MAXLONG` の使用に関する注記を参照してください。

orams (nullvalue)

プログラマが設定できる要素で、NULL の結果に対して返される文字列値を指定します。`orams (nullvalue)` を `DEFAULT` に設定すると、`INTEGER`、`FLOAT` および `MONEY` のような数値 NULL データ型に対しては 0 が返され、その他のすべてのデータ型に対しては NULL 文字列が返されます。NULLVALUE は、初期値では `default` に設定されています。

orams (oraobject)

スクリプトの動作対象オブジェクトを示します。イベント・スクリプトに対してのみ定義されます。

orams (orahome)

ORACLE_HOME ディレクトリです。

orams (oraindex)

構成ファイル `snmp.ora` から読み込まれた SNMP 索引値の Tcl リストです。

orams (orainput)

ジョブの入力ファイルの名前を示す Tcl リストです。ほとんどのジョブは入力ファイルを必要としませんが、SQL スクリプトで `SQL*Plus` を起動するジョブ、または仕様ファイルで `Export` を起動するジョブは、入力ファイル機能を使用します。ジョブ・スクリプトに対してのみ定義されます。

orams (rc)

最後の SQL コマンドと、後続の `orafetch` 関数による処理の結果を示します。`rc` は `orasql`、`orafetch`、`oraplexec` によって設定され、`OraTcl` コマンドによって最後にコールされた OCI ライブラリ関数からの数値リターン・コードが入ります。

詳細は『Oracle9i データベース・エラー・メッセージ』を参照してください。一般的な値は表 3-1「エラー・メッセージ」に一覧で示されています。

表 3-1 エラー・メッセージ

エラー	意味
0000	関数は正常に終了。エラーはありません。
0900 - 0999	無効な SQL 文、キーワード不足、無効な列名など。
1000 - 1099	プログラム・インタフェース・エラー。SQL 文の欠如、ログイン拒否、必要な権限不足など。
1400 - 1499	実行エラーまたはフィードバック。
1403	orafetch コマンドの実行時にデータの終端に到達。
1406	orafetch によってフェッチされる列が切り捨てられた。 LONG または LONG RAW データのフェッチ時に、 maxlong の値が実際のデータ・サイズよりも小さい場合に 発生する可能性があります。

oramsg (rows)

orasql コマンドでの挿入、更新または削除によって変化する行の数、または orafetch によってフェッチされる累計の行数です。

oramsg (starttime)

ジョブが開始するようにスケジューリングされた時刻です。ジョブに対してのみ定義されます。

修正ジョブの Tcl 配列に対するイベント

OraTcl は、Tcl グローバル配列 trigevent を作成、維持して、Tcl 配列を Enterprise Manager の修正ジョブに渡します。trigevent 配列は、修正ジョブ自体からも使用できません。

trigevent の要素

trigevent の要素を次に示します。

trigevent (name)

修正ジョブを起動するイベントの名前です。

trigevent (object)

ネットワークのターゲットまたはノードです。

trigevent (arguments)

トリガー・イベントへの引数です。イベントにより異なります。

trigevent (results)

イベントの結果です。たとえば、この要素は CPU 使用率イベント・テストの使用率が 35% であることを示す数値 35 を返します。

trigevent (severity)

イベントの重大度です。-1（安全）、1（警告）、2（限界）があります。

警告： trigevent は、修正ジョブでのみ機能します。この配列が修正ジョブ以外に渡されると、trigevent は未定義（NULL）になりジョブは失敗します。このため、trigevent 配列は、その要素の参照が解除される前に必ずチェックする必要があります。

例

次の例は、2 つの異なるタスクを実装する修正ジョブを示しています。

- trigevent を使用して修正ジョブに配列を渡す Tcl スクリプト

```
global trigevent
if {[info exists trigevent]} {
    puts "Event name: $trigevent(name)"
    puts "Event object: $trigevent(object)"
    puts "Event arguments: $trigevent(arguments)"
    puts "Event results: $trigevent(results)"
    puts "Event severity: $trigevent(severity)"
} else {
    puts "Not a fixit"
}
```

- オペレーティング・システム・コマンドの実行

```
top -dl -ocpu
```

次の例の修正ジョブは、特定のレベルの CPU の使用率を監視する CPUUTIL イベント・テストと対応付けられています。このイベント・テストの場合、次のようなパラメータが設定されます。

- 限界のしきい値 = 20
- 警告のしきい値 = 10
- 修正ジョブ = プロパティ・シートで選択したオプション

CPUUTIL イベント・テストを含むイベントが引き起こされると、対応する修正ジョブが実行されます。修正ジョブでは、次のような出力が生成されます。

実行された最初のタスク : trigevent 配列の情報が表示されます。

```
Event name: /oracle/host/perf/cpuutil
Event object: aholser-sun
Event arguments: {1} {20} {10}
Event results: 63
Event severity: 2
```

実行された 2 番目のタスク : オペレーション・システム・コマンド top -d1 -ocpu が実行されます。

```
last pid: 26420; load averages:  0.64,  0.56,  0.48    13:16:00
111 processes: 110 sleeping, 1 on cpu
```

```
Memory: 128M real, 7080K free, 89M swap in use, 912M swap free
```

PID	USERNAME	THR	PRI	NICE	SIZE	RES	STATE	TIME	CPU	COMMAND
727	root	1	30	0	128M	19M	sleep	17:22	10.54%	Xsun
25915	aholser	4	31	0	12M	5032K	sleep	4:47	10.32%	db snmp
823	aholser	1	34	0	10M	4368K	sleep	2:11	7.37%	dtterm
26402	aholser	1	34	0	976K	872K	sleep	0:03	3.57%	find
26415	aholser	4	34	0	11M	4304K	sleep	0:00	3.11%	db snmp
26403	aholser	1	23	0	976K	872K	sleep	0:02	2.60%	grep
25914	aholser	4	34	0	11M	4832K	sleep	0:56	2.56%	db snmp
26419	aholser	1	-5	0	1576K	1368K	cpu	0:00	1.64%	top
894	aholser	1	33	0	5904K	3456K	sleep	7:07	0.86%	view_server
159	root	5	23	0	3176K	1976K	sleep	6:21	0.29%	automountd
26418	aholser	1	25	0	920K	760K	sleep	0:00	0.28%	sh
1007	root	3	33	0	1840K	1400K	sleep	0:40	0.06%	cache fsd

Intelligent Agent での Tcl の使用

Tcl スクリプトは、Intelligent Agent でジョブまたはイベントを扱うために使用されます。同じ Tcl スクリプトでも、Intelligent Agent とユーザー・インタフェースではその扱いが異なります。

ジョブとは、1 回または複数回実行されるようにスケジューリングされるスクリプトのことです。ジョブは一般に、データベースを起動する、バックアップを実行する、または puts コマンドを通して画面に出力を送るなどの副次作用を伴い、実行には長い時間を要することがあります。イベント・スクリプトと異なり、ジョブには SQL スクリプトなどの入出力ファイルを与えることができます。UNIX、DOS 上では、出力ファイルが stdout でリダイレクトされることに注意してください。

一方イベント・スクリプトは、例外の検出にその目的を限って使用されます。Tcl イベント・スクリプトは様々な手段を利用して、データベース、ホスト・システムまたは Oracle Net サービスを監視できます。特定の条件が満たされたと判断すると、スクリプトはイベントの重大度を示すリターン・コードを **Intelligent Agent** に送ります。イベント・スクリプトはジョブよりも頻繁に実行される傾向があり、実行時間は比較的短いことが想定されます。また、イベント・スクリプトには副次作用が伴わないことも想定されます。

ジョブとイベントはともに Tcl を使用してタスクを遂行する一方で、実行環境が異なるなど、両者の性質は大きく異なっています。特に UNIX システム上では、イベントは通常 **Intelligent Agent** コードとともにインラインで実行される一方で、ジョブは独立したプロセスへと分岐します。

インラインのイベント・スクリプトにかぎり、Tcl インタプリタの状態が実行の間に保存され、Tcl グローバル変数の値が保持されます。これらは仮想プロセスの存在を装うための動作です。これにより、イベント・スクリプトは履歴を維持して、イベントが何度も呼び出されるのを防ぐことができます。たとえば、ある値が 90 を超えたことをコンソールに通知した後で、次に値が 80 を下回り再び 90 を超えるまでの間、再度の通知を抑えることができます。oralogon 関数を使用しているデータベース接続は、インラインの全イベント・スクリプトにまたがってキャッシュされます。これにより、同じ接続文字列を使用して繰り返し実行されるイベント・スクリプトが、同じ接続を利用できます。

ジョブとイベントの両方に対して、すべてのコマンドとグローバル変数が使用可能なわけではありません。スクリプトの動作対象となっているサービスの種類をイベントに知らせる oraobject グローバル変数は、ジョブには存在しません。また、ジョブが SQL*Plus スクリプトに対して使用する orainput グローバル変数はイベントにはありません。

NLS の問題とエラー・メッセージ

ユーザーがイベントを登録する、またはジョブをスケジューリングするとき、**Intelligent Agent** に対してユーザーの言語環境設定を使用できます。各コンソール・ユーザーの言語および現在のアドレスを報告するための、特殊なリモート・プロシージャ・コールが用意されています。oralogon 関数がコールされるたびに、**Intelligent Agent** は指定された言語への ALTER SESSION コマンドを前もって発行します。これは、それ以降に Oracle データベースから送られてくるメッセージまたは出力が、ユーザーの言語になることを意味します。加えて、キャラクタ・セットの変換は **Intelligent Agent** 上で明示的に行われず、コンソールがユーザー側でその変換を行うことができます。

イベント・スクリプトまたはジョブ・スクリプトの実行が失敗すると、エラー・メッセージがユーザーの言語でコンソールに返されます。通常これは、関数が不適切なパラメータを与えられた場合に、Oracle Tcl 拡張機能のいずれかによって返される Oracle メッセージです。たとえば、不正な接続文字列を指定すると、oralogon では「ERROR: ORA-01017: ユーザー名 / パスワードが無効です。ログインは拒否されました」というエラーが返されます。ただし、エラー・メッセージは「ERROR: Tcl-00456: division by zero error」という Tcl 固有のメッセージになることもあります。これはメッセージ・ファイルに格納されるため、ユーザーの作業環境の言語で返されることはありません。ユーザーの言語作業環境が指定されて

いないか、ユーザーの言語のエラー・メッセージ・テキストが存在しない場合、Intelligent Agent によって使用されるデフォルトの言語はアメリカ英語です。

OraTcl の関数とパラメータ

この項では、OraTcl の関数とパラメータの一覧を示します。OraTcl 構文中の関数またはその他の語は、`function` のようなフォントで示されます。角カッコ内のパラメータ [option] はオプションで、文字 'I' は「または」の意味です。すべてのパラメータは関数に渡され、IN モードです。

SQL および PL/SQL 関数

- `oraautocom`
- `oracancel`
- `oraclose`
- `oracols`
- `oracommmit`
- `orafetch`
- `oralogoff`
- `oralogon`
- `oraopen`
- `oraplexec`
- `orareadlong`
- `oraroll`
- `orasql`
- `orawritelong`

RDBMS 管理用関数

- `orastart`
- `orastop`

SNMP アクセス用関数

- `oradbsnmp`
- `orasnmp`

Intelligent Agent およびその他の Oracle ソフトウェアとの通信用関数

- [orafail](#)
- [orainfo](#)
- [orajobstat](#)
- [orareportevent](#)

キャラクタ・セット変換およびエラー処理用関数

- [convertin](#)
- [convertout](#)
- [msgtxt](#)
- [msgtxt1](#)

汎用ユーティリティ関数

- [orasleep](#)
- [oritime](#)

共通パラメータ

次のパラメータは複数の OraTcl 関数で 사용되는もので、この項に説明があります。

column

LONG または LONG RAW 列である列名です。

connect_string

有効な Oracle データベース接続文字列で、次のいずれかの形式になります。

name | name/password | name@n:dbname | name/password@n:dbname

destaddress

destaddress は、Intelligent Agent の宛先アドレスです。

filename

列に書き込まれる LONG または LONG RAW データを含むファイルの名前、または LONG または LONG RAW データが書き込まれるファイルの名前です。

logon-handle

以前に oraopen でオープンされた、有効なカーソル・ハンドルです。ハンドルとは、関数の追跡に使用されるメモリー内でのマッピングのことです。

rowid

既存の行の Oracle データベース rowid で、Oracle の rowid データ型の形式である必要があります。

table

行および列を含む Oracle データベースの表の名前です。

convertin

用途 この関数は、パラメータ文字列をクライアント（コンソール）のキャラクタ・セットから接続先のキャラクタ・セットに変換します。返される値は、変換後の文字列です。

構文 `convertin dest_characteraset string`

パラメータ

dest_characteraset

接続先のキャラクタ・セットです。ジョブまたはイベントに対しては、`$oramsmsg(agent_characteraset)` を使用します。3-5 ページの「[oramsmsg の要素](#)」を参照してください。

string

変換される文字列です。

コメント クライアントおよび Intelligent Agent のノードで、異なる言語またはキャラクタ・セットが使用される場合があります。キャラクタ・セットの変換を実行することは、Tcl スクリプト作成者の責任です。一般に、それらが ASCII であることが保証されていないかぎり、ジョブまたはイベントのすべての入力パラメータを変換してください。

convertout

用途 この関数は、パラメータ文字列を接続先のキャラクタ・セットからクライアント（コンソール）のキャラクタ・セットに変換します。返される値は、変換後の文字列です。

構文 `convertout dest_characteraset string`

パラメータ

dest_characteraset

接続先のキャラクタ・セットです。ジョブまたはイベントに対しては、`$oramsmsg(agent_characteraset)` を使用します。3-5 ページの「[oramsmsg の要素](#)」を参照してください。

string

変換される文字列です。

コメント クライアントおよび Intelligent Agent のノードで、異なる言語またはキャラクタ・セットが使用される場合があります。キャラクタ・セットの変換を実行することは、Tcl スクリプト作成者の責任です。一般に、それらが ASCII であることが保証されていないかぎり、ジョブまたはイベントのすべての出力を変換してください。

msgtxt

用途 この関数は、特定の製品名、機能およびメッセージ番号に対応するメッセージ・テキストを、クライアント（コンソール）の言語およびキャラクタ・セットで返します。出力は FACILITY-ERROR: MESSAGE TEXT のような形式になります。

構文 `msgtxt product facility error_no`

パラメータ**product**

製品名です。出力は rdbms のようになります。

facility

機能名です。出力は ora のようになります。

error_no

エラー番号です。出力は 1101 のようになります。

コメント この関数は、ジョブの出力ファイルにエラー・メッセージを出力するために使用されます。メッセージはクライアント（コンソール）の言語で表示されます。

msgtxt1

用途 この関数は、特定の製品名、機能およびメッセージ番号に対応するメッセージをクライアント（コンソール）の言語で返します。出力は MESSAGE TEXT のような形式になります。

構文 `msgtxt1 product facility error_no`

パラメータ

product

製品名です。出力は rdbms のようになります。

facility

機能名です。出力は ora のようになります。

error_no

エラー番号です。出力は 1101 のようになります。

コメント この関数は、ジョブの出力ファイルに確認メッセージを出力するために使用されます。メッセージはクライアント（コンソール）の言語で表示されます。

oraautocom

用途 この関数は、logon-handle が指示する接続を通してオープンされたカーソルを使用する、SQL データ操作文の自動コミットを有効または無効にします。

構文 oraautocom logon-handle {on | off}

パラメータ

logon-handle

3-13 ページの「[共通パラメータ](#)」を参照してください。

コメント 指定された logon-handle がオープンされていない場合、oraautocom は Tcl エラーを返します。

on または off のどちらかを指定する必要があります。自動コミット機能は、デフォルトではオフになっています。

oracancel

用途 この関数は、logon-handle が指示する接続を通してオープンされたカーソルを使用する、先行の orasql 関数によって返されたすべての保留中の結果を取り消します。

構文 oracancel logon-handle

パラメータ

logon-handle

3-13 ページの「[共通パラメータ](#)」を参照してください。

コメント 指定された logon-handle がオープンされていない場合、oracancel は Tcl エラーを返します。

oraclose

用途 この関数は、logon-handle に関連付けられたカーソルをクローズします。

構文 oraclose logon-handle

パラメータ

logon-handle

3-13 ページの「[共通パラメータ](#)」を参照してください。

コメント 指定された logon-handle がオープンされていない場合、oraclose は Tcl エラーを返します。

oracols

用途 この関数は、最後に実行された orasql、orafetch または oraplexec 関数から、列の名前を Tcl リストで返します。oracols は、oraplexec の後に使用できますが、この場合はバインドされた変数名が返されます。

構文 oracols logon-handle

パラメータ

logon-handle

3-13 ページの「[共通パラメータ](#)」を参照してください。

コメント 指定された logon-handle がオープンされていない場合、oracols は Tcl エラーを返します。

oramsq の配列索引 collengths には、列の長さに対応する Tcl リストが設定されます。索引 coltypes には、列の型に対応する Tcl リストが設定されます。索引 colprecs には、数値列の精度に対応する Tcl リストが設定され、対応するその他の非数値列は NULL 文字列になり

ます（バージョン7のみ）。索引 `colScales` には、数値列のスケールに対応する Tcl リストが設定され、対応するその他の非数値列は `NULL` 文字列になります（バージョン7のみ）。

oracommith

用途 この関数は、`logon-handle` が指示する接続を通してオープンされたカーソルを使用する、先行の `orasql` 関数によって返されたすべての保留中のトランザクションをコミットします。

構文 `oracommith logon-handle`

パラメータ

`logon-handle`

3-13 ページの「[共通パラメータ](#)」を参照してください。

コメント 指定されたログイン・ハンドルがオープンされていない場合、`oracommith` は Tcl エラーを返します。

oradbsnmph

用途 この関数は、SNMP MIB 値を取り出します。

構文 `oradbsnmph get | getnext object_Id`

パラメータ

`object_Id`

`object_Id` の取り得る値は、1.3.6.1.2.1.1.1.0 のような実際の MIB オブジェクト ID か、`sysDescr` または `sysDescr.0` のように、それに付加できる索引を伴ったオブジェクト名のどちらかです。

コメント `oradbsnmph` は、RDBMS パブリック MIB、または Oracle RDBMS プライベート MIB など、Intelligent Agent によってメンテナンスされる SNMP MIB 値を取り出すための関数です。この関数は、通常の SNMP 用 UDP ポートへの書込みを行わず、SNMP MIB 値を Intelligent Agent の内部データ構造から直接取得します。この関数は、ホスト上で SNMP Master Agent が動作していない場合に有用です。`get` および `getnext` の機能の詳細は、3-27 ページの「[oradbsnmph](#)」を参照してください。SQL コマンドを使用して V\$ 表から値をフェッチするのではなく、`oradbsnmph` を使用する必要があるのは、次のような理由によります。

- Intelligent Agent は、V\$ 表からフェッチされる MIB 値のキャッシュをメンテナンスして、RDBMS への過度の負荷を回避します。oradbsnmp は SQL よりも高速な場合が多いため、システムへのオーバーヘッドが軽減されます。
- SGA アクセスが実装されているとき、SGA から直接フェッチされるこれらの MIB 変数について、SGA アクセスはこの関数に対して透過です。
- getnext の場合、次の object_id は、プライベートおよびパブリックの RDBMS MIB 内部の次の object_id であって、別の MIB のそれではありません。この関数を使用してシステム固有の情報を取り出すことはできません。かわりに orasnmpp を使用してください。

orafail

用途 この関数は、Tcl スクリプトを強制的に失敗させます。

構文 orafail errmsg

パラメータ

errmsg

errmsg の取り得る値は、引用符で囲まれたテキスト文字列、または FAC-XXXXX のような形式の文字列のどちらかです。XXXXX の部分には、VOC-99999 のように、特定の機能に対する Oracle メッセージ番号が入ります。

コメント エラー・メッセージは、クライアント側での表示目的に使用されます。

orafetch

用途 この関数は、orasql で実行された最後の SQL 文からの次の行を Tcl リストで返します。

構文 orafetch logon-handle [commands]

パラメータ

logon-handle

3-13 ページの「[共通パラメータ](#)」を参照してください。

commands

オプションのコマンドを使用することにより、orafetch で行を繰り返しフェッチして、各行に対してコマンドを実行できます。

コメント 指定された logon-handle がオープンされていない場合、orafetch は Tcl エラーを返します。

返されるすべての列はキャラクタ文字列に変換されます。現行の結果セット内にそれ以上行がない場合、NULL 文字列が返されます。orafetch によって返される Tcl リストには、選択された列の値が、SELECT 文によって指定された順序で格納されます。

それぞれの行に対して Tcl リストを Tcl_Eval() に渡す前に、コマンド上で置換が行われます。orafetch では、コマンド中の文字列 @n を使用して結果の列を指定します。たとえば、@1、@2、@3 はそれぞれ結果中の第 1 列、第 2 列、第 3 列を指します。@0 は、結果行全体を Tcl リストで指します。置換列は任意の順番で、あるいは同じコマンド中に複数回指定できます。置換される列は、適切なリスト要素としてコマンド文字列に挿入されます。たとえば、置換の前後に 1 文字分の空白が追加され、埋め込まれた空白を伴う列値は必要に応じて中カッコ ({}) で囲まれます。

列の置換番号が結果中の列数を超える場合、Tcl エラーが返されます。コマンドがブレークを実行すると、orafetch の実行は中断され、Tcl_OK が返されます。残りの行は、それ以降の orafetch 関数でフェッチできます。コマンドがリターンまたはコンティニューを実行すると、残りのコマンドはスキップされ、orafetch の実行は次の行から続きます。コマンドがエラーを返すと、orafetch は Tcl エラーを返します。コマンドは二重引用符 ("") または中カッコ ({}) で囲んでください。

OraTcl は、すべてのデータ型に対する変換を実行します。生データは、先頭に 0x の付かない 16 進文字列で返されます。特定の変換を強制するには、SQL 関数を使用します。

oramsg の配列索引 rc は、フェッチのリターン・コードに設定されます。0 は行のフェッチに成功したことを示し、1403 はデータの終端に到達したことを示します。行の索引は、その時点までにフェッチされた累計の行数に設定されます。

oramsg の配列索引 maxlong は、返される各列に対して返される LONG または LONG RAW データの量を制限します。デフォルトは 32768 バイトです。oramsg の配列索引 nullvalue を設定して、列が NULL のときに返される値を指定できます。デフォルトは数値データに対しては "0"、その他のデータ型に対しては "" です。

destaddress は、orainfo 関数を使用して取得できます。提供されるアドレスは Intelligent Agent の子アドレス、つまり Intelligent Agent がファイル転送要求をリスニングする特別のアドレスであり、他のすべての RPC に対して使用される通常のアドレスではありません。

orainfo

用途 この関数はジョブによって使用され、構成情報を取得します。

構文 `orainfo destaddress`

パラメータ

destaddress

3-13 ページの「[共通パラメータ](#)」を参照してください。

コメント `orainfo` は、`destaddress` の Intelligent Agent から、Intelligent Agent の構成情報をフェッチします。`destaddress` が存在しない場合、ローカル・マシン上の Intelligent Agent から構成情報がフェッチされます。Intelligent Agent の構成は、次の要素からなる Tcl リストです。

- この Intelligent Agent によって監視されるデータベースのリスト。このリストには各データベースのデータベース名、ORACLE_HOME および SID が含まれます。
- Intelligent Agent の通常の RPC アドレス。`tnsnames` (TNS) 文字列で表されます。
- Intelligent Agent のファイル転送アドレス。TNS 文字列で表されます。

orajobstat

用途 この関数はジョブによって使用され、中間出力をコンソールに返します。

構文 `orajobstat destaddress string`

パラメータ

destaddress

3-13 ページの「[共通パラメータ](#)」を参照してください。

string

`string` の取り得る値は、引用符で囲まれたテキスト文字列、または FAC-XXXXX のような形式の文字列のどちらかです。XXXXX の部分には、VOC-99999 のような、特定の機能に対する Oracle のメッセージ番号が入ります。文字列はクライアント側での表示目的に使用されます。

コメント `destaddress` は、デーモンではなく Intelligent Agent のアドレスです。この関数は、Intelligent Agent プロセスの内部からではなくジョブ・プロセスから発行されます。Intelligent Agent のアドレスは `orainfo` を使用して取得できます。

oralogoff

用途 logon-handle に関連付けられた Oracle データベース接続からログオフします。

構文 oralogoff logon-handle

パラメータ

logon-handle

3-13 ページの「[共通パラメータ](#)」を参照してください。

コメント 指定されたログイン・ハンドルがオープンされていない場合、oralogoff は Tcl エラーを返します。oralogoff は NULL 文字列を返します。

oralogon

用途 connect_string を使用して Oracle データベースに接続します。

構文 oralogon connect_string

パラメータ

connect_string

3-13 ページの「[共通パラメータ](#)」を参照してください。

コメント この関数で確立された接続を使用し、logon-handle を必要とする他のすべての OraTcl 関数に対して、返された logon-handle を使用してください。同一または異なるサーバーへの複数の接続が可能です。

追加情報： oralogon がイベント・スクリプト内で使用されるとき、接続キャッシュが役に立ちます。oralogon は通常、同じデータベースに対して他のイベント・スクリプトがオープンした接続を再利用できます。詳細は、3-11 ページの「[NLSの問題とエラー・メッセージ](#)」を参照してください。無効なログインまたはネットワークの使用不可など、なんらかの理由で接続が確立されない場合、oralogon は Tcl エラーを返します。connect_string でデータベースが指定されていない場合、サーバーとして環境変数 ORACLE_SID の値が使用されます。

oralogon_unreached

用途 この関数により、接続文字列およびオプションのロールを使用して、Oracle サーバーに接続します。この接続は共有できません。

構文 oralogon_unreached connect_string [AS] [SYSDBA|SYSOPER|NORMAL]

connect_string

3-13 ページの「[共通パラメータ](#)」の共通パラメータを参照してください。

コメント この動詞は、返される接続が共有不可である点を除いて oralogon と同じです。また、オプションでロールを指定できます。

oraopen

用途 この関数は、サーバーへの SQL カーソルをオープンします。oraopen は、logon-handle を必要とするそれ以降の OraTcl 関数で使用するためのカーソルを返します。

構文 oraopen logon-handle

パラメータ

logon-handle

3-13 ページの「[共通パラメータ](#)」を参照してください。

コメント 指定された logon-handle がオープンされていない場合、oraopen は Tcl エラーを返します。合計で 25 のカーソルを上限として、同じまたは異なるログイン・ハンドルを通して複数のカーソルをオープンできます。

oraplexec

用途 この関数は、無名 PL/SQL ブロックを実行します。オプションで、PL/SQL 変数に値をバインドします。

構文 oraplexec logon-handle pl_block [:varname value ...]

パラメータ

logon-handle

3-13 ページの「[共通パラメータ](#)」を参照してください。

pl_block

pl_block の取り得る値は、完全な PL/SQL プロシージャ、または無名 PL/SQL ブロックとしてコーディングされたストアド・プロシージャへのコールのどちらかです。

:varname value

:varname value はオプションのペアです。

コメント 指定された logon-handle がオープンされていない場合、または PL/SQL ブロックがエラー状態にある場合、oraplexec は Tcl エラーを返します。PL/SQL ブロックの終了時に、oraplexec は各 :varname の内容を Tcl リストとして返します。oraplexec は、その戻り値として結果セットを Tcl リストで返します。

pl_block の後に、オプションの :varname value ペアが続くことがあります。変数名の先頭にはコロン記号を付け、プロシージャ内で使用される置換名に一致させる必要があります。値と一致しないすべての :varname は無視されます。:varname が出力に対して使用される場合、値を NULL 文字列 "" としてコーディングしてください。

oramsg の配列索引 rc には、ストアド・プロシージャからのリターン・コードが含まれます。

orareadlong

用途 この関数は、LONG または LONG RAW 列の内容を読み込み、結果をファイルに書き込みます。

構文 orareadlong logon-handle rowid table column filename

パラメータ**logon-handle rowid table column filename**

3-13 ページの「[共通パラメータ](#)」を参照してください。

コメント orareadlong が正常に終了すると、LONG 列から読み込まれたバイト数が 10 進値で返されます。

指定された logon-handle がオープンされていない、rowid、table または column が無効である、あるいは行が存在しないなどの場合には、orareadlong は Tcl エラーを返します。

orareadlong は、table、column および rowid の値を基に SQL の SELECT 文を構成し、実行します。SELECT rowid FROM table WHERE ... のような orasql を前もって実行することにより、適切な形式の ROWID を取得できます。

orareporevent

用途 この関数はジョブによって使用され、不要なイベントを Intelligent Agent、またはコンソール内のイベント管理システムに報告します。oemevent の実行可能ファイルを使用することもできます。

構文 orareporevent eventname object severity message [results]

パラメータ

eventname

eventname はイベントの名前です。これは、次のような形式の 4 つの部分からなるイベント名です。

/vendor/product/category/name

任意のキャラクタ文字列を入力できますが、4 つの部分すべてとスラッシュが必須です。

名前の最初の 2 つの部分は特に重要で、Oracle スクリプトの作成者が使用する必要がある多くの定義済文字列が用意されています。

- レベル 1 はスクリプトの定義者で、一般には oracle などのインテグレータ企業名、指定のない顧客に対しては user のような名前です。
- レベル 2 はスクリプトを関連付ける製品の名前で、例としては rdbms、office、agent、osgeneric、sqlnet または hpux などがあります。すべての Oracle サービスには、Oracle スクリプトの記述者が使用する必要がある定義済の名前が用意されています。

eventname は 7 ビット ASCII 形式であることが想定されます。プラットフォームまたは言語によって変化しないことがその理由です。定義済イベント名のリストは、%ORACLE_HOME%\net8¥ディレクトリにある eventdef.tcl (Windows NT プラットフォーム上の Oracle Enterprise Manager リリース 1.5.0 の場合) を参照してください。

注意： 実際のイベント・スクリプト名は、特定のプラットフォーム上で有効な一意のファイル名とするために、短縮されたり、大文字に変えられたり、その他の方法で操作されることがあります。操作形式はオペレーティング・システムごとに固有です。たとえば、/oracle/rdbms/security/SecurityError は、UNIX システム上では \$ORACLE_HOME/network/agent/events/oracle/rdbms/security/securityerror.tcl として格納されることがあります。

object

object はイベントが監視しているオブジェクトの名前で、snmp.ora ファイル、または \$oramsg(nodename) の snmp.visible services パラメータにリストされるデータベースまたはサービスの名前などが相当します。

severity

severity は、イベントの重大度のレベルを表します。orareportevent では、値は 1（警告）、2（限界）、または -1（安全）のいずれかです。oemevent では、リテラル・テキスト文字列 alert、warning または clear が使用されます。

message

message は、" ファイルが見つかりません " のような、コンソール上に表示される引用符付きテキスト文字列です。

[results]

results は、イベントから発生する可能性のあるすべての結果を表します。これは、エラーのある表領域やセキュリティ違反を犯したユーザーなど、イベントに対する特定の結果からなる Tcl リストです。

コメント この関数は、必要でないイベントを任意のジョブが Intelligent Agent に報告し、コンソールに返すための手段です。イベント管理システムの詳細は、『Oracle Enterprise Manager 管理者ガイド』を参照してください。

oraroll

用途 この関数は、logon-handle が指示する接続を通してオープンされたカーソルを使用する、先行の orasql 関数によって返されるすべての保留中トランザクションをロール・バックします。

構文 oraroll logon-handle

パラメータ

logon-handle

3-13 ページの「[共通パラメータ](#)」を参照してください。

コメント 指定されたログイン・ハンドルがオープンされていない場合、oraroll は Tcl エラーを返します。

orasleep

用途 この関数は、要求された秒数のみ Tcl スクリプトを一時停止します。

構文 orasleep seconds

パラメータ

seconds

コメント orasleep は、要求された秒数を引数として slcsleep() をコールします。デフォルト、下限または上限の値はありません。

orasnmp

用途 この関数は、object_id によって指定されたオブジェクト上で、SNMP の get または getnext のどちらかの操作を実行します。

構文 orasnmp get | getnext object_Id

パラメータ

object_Id

object_Id の取り得る値は、1.3.6.1.2.1.1.0 のような実際の MIB オブジェクト ID か、sysDescr または sysDescr.0 のように、それに付加される索引を伴ったオブジェクト名のどちらかです。

コメント オブジェクト名は MIB テキスト・ファイルに由来します。OpenView のような完全なネットワーク・マネージャは、MIB ファイルを受理し、ASN.1 を解析する MIB コンパイラを備えており、すべての MIB 内のすべてのオブジェクトのデータベースを作成します。Intelligent Agent はよりシンプルである必要があります。次のような形式の、1 つまたは複数の 2 列 ASCII ファイルを含む、標準の構成ディレクトリが存在します。

```
"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",
....
```

Tcl インタプリタはこれらのファイルを読み込み、実行時にこれらのファイル上でバイナリ検索を実行して、オブジェクト名を object_Id に解決します。

Oracle サービスに使用する索引値は、snmp.ora ファイルを介して構成されます。これらの索引は、グローバル変数 oraindex からも取得できます。3-5 ページの「[サーバー・メッセージとエラー情報](#)」を参照してください。

orasnmp の結果は、次のような形式の Tcl リストです。

```
{object_id value}
```

object_Id には、value に関連付けられたオブジェクト ID が入ります。object_Id は、orasnmp get の場合はオブジェクトと同じですが、getnext の場合は次の論理 object_Id になります。orasnmp 操作は、ローカル・ホストのみに適用されると想定されています。この関数は、ローカル・ホスト上の標準の SNMP ポートに SNMP 問合せを実際に送出します。したがって、SNMP をサポートするホストまたはその他のアプリケーションの MIB 変数など、Oracle 以外の MIB 変数を問い合わせることができます。この関数が機能するためには、ローカル・ホスト上で SNMP Master Agent が動作している必要があります。Oracle データベースの MIB オブジェクトを取り出すための最適化された方法については、3-18 ページの「[oradbsnmp](#)」を参照してください。Master Agent が動作していない場合、この関数は失敗します。

orasql

用途 この関数は、Oracle SQL 文をサーバーに送ります。

構文 orasql logon-handle sql_stmt

パラメータ

logon-handle

3-13 ページの「[共通パラメータ](#)」を参照してください。

sql_stmt

sql_stmt は、単一の有効な SQL 文です。

コメント logon-handle は、あらかじめ oraopen でオープンされた有効なハンドルである必要があります。指定された logon-handle がオープンされていないか、SQL 文の構文が不正な場合、orasql は Tcl エラーを返します。

SQL 文の実行が成功すると、orasql は数値リターン・コード 0 を返します。oramsq の配列索引 rc は、リターン・コードに設定されます。rows 索引は、挿入、更新または削除の場合に SQL 文で決定される行数に設定されます。sql_stmt では単一の SQL 文のみを指定できます。orafetch を使用すると、生成される戻り行を検索できます。最後に実行された orasql によって返された保留中の結果がある場合、orasql は暗黙のうちに oracancel を実行します。

orasql で行う表への挿入は、『Oracle9i SQL リファレンス』に記載の変換ルールに従ってください。

orastart

用途 この関数は、Oracle データベースのインスタンスを起動します。

構文 `orastart connect_string [init_file] [SYSDBA|SYSOPER] [RESTRICT] [PARALLEL] [SHARED]`

パラメータ

connect_string

3-13 ページの「[共通パラメータ](#)」を参照してください。

init_file

init_file は、使用する init.ora ファイルへのパスです。

コメント init_file のデフォルト値は、次のとおりです。

```
ORACLE_HOME/dbs/init${ORACLE_SID}.ora
```

[SYSDBA|SYSOPER] は、データベースを起動するユーザーのロール・フラグです。

[RESTRICT] [PARALLEL] [SHARED] はデータベース・オプションです。[RESTRICT] が指定されている場合、データベースは制限付きモードで起動されます。

orastop

用途 この関数は、Oracle データベースのインスタンスを停止します。

構文 `orastop connect_string [SYSDBA|SYSOPER] [IMMEDIATE|ABORT]`

パラメータ

connect_string

3-13 ページの「[共通パラメータ](#)」を参照してください。

コメント [SYSDBA|SYSOPER] は、データベースをシャットダウンするユーザーのロールを示すフラグです。[IMMEDIATE|ABORT] は、シャットダウン・モードを示すフラグです。

注意： 通常のシャットダウン操作が確実に失敗することが予測されている状況も存在します。これは、Intelligent Agent がデータベースへの独自の接続を維持している一方で、これを行うときにユーザーが Intelligent Agent に特殊な RPC を送り、それにより Intelligent Agent がデータベースから切断されることが原因です。

oritime

用途 この関数は、現在の日付と時刻を返します。

構文 oritime

パラメータ なし

コメント なし

orawritelong

用途 この関数は、ファイルの内容を LONG または LONG RAW 列に書き込みます。

構文 orawritelong logon-handle rowid table column filename

パラメータ

logon-handle rowid table column filename

3-13 ページの「[共通パラメータ](#)」を参照してください。

コメント orawritelong は、table、column および rowid を基に SQL の UPDATE 文を構成し、実行します。orawritelong が正常に終了すると、LONG 列に書き込まれたバイト数が 10 進の数値で返されます。SELECT rowid FROM table WHERE ... のような orasql 関数を前もって実行することにより、適切な形式の ROWID を取得できます。

指定された logon-handle がオープンされていない、rowid、table または column が無効である、あるいは行が存在しないなどの場合には、orawritelong は Tcl エラーを返します。

Intelligent Agent 構成ファイル

この付録では Intelligent Agent によって生成される構成ファイル、および異なるシステム設定に対して Intelligent Agent の運用を最適化するために設定できるパラメータについて説明します。次の項目について説明します。

- [構成ファイル](#)
- [ユーザーが構成可能なパラメータ](#)
- [Intelligent Agent のログ・ファイル](#)

構成ファイル

Intelligent Agent の操作は、次のファイルによって制御されます。

snmp_ro.ora

snmp_ro.ora ファイルは、\$ORACLE_HOME/network/admin ディレクトリ、または環境変数 TNS_ADMIN で指定されるディレクトリにあります。この読み専用ファイルは更新しないでください。

snmp_rw.ora

snmp_rw.ora ファイルは、Intelligent Agent の \$ORACLE_HOME/network/admin ディレクトリにあります。この読み書き両用ファイルは修正が可能です。修正は慎重に行ってください。

services.ora

services.ora ファイルは Intelligent Agent の起動時に作成され、Windows NT では %ORACLE_HOME%\network\agent ディレクトリに、UNIX では \$ORACLE_HOME/network/agent ディレクトリに置かれます。このファイルには、Oracle データベースやリスナーなど、Intelligent Agent が動作するノード上のサービスの一覧が含まれます。Intelligent Agent のリリース 9.x 以上では、Intelligent Agent が稼働している環境のオペレーティング・システムに関するバージョンおよびプラットフォーム情報も、このファイルに含まれます。このファイルは Oracle Enterprise Manager ナビゲータの「ノードの検出」メニュー・オプションを通して Intelligent Agent から検索されます。

注意： services.ora ファイルを手動で編集しないでください。このファイルは、起動時に Intelligent Agent によって再度書き込まれます。

ユーザーが構成可能なパラメータ

これらのパラメータは、Intelligent Agent リリースの構成ファイル snmp_rw.ora で使用されます。

agentctl.trace_level = off | user | admin | nn

指定されたレベルで、AGENTCTL 実行可能ファイルに対するトレースを有効にします。オラクル社では、トレース・レベルを 13 に設定することをお薦めしています。レベル 16 は最大の情報を提供しますが、これは不具合の調査時にのみ有用です。レベル 16 では、実際の TCP/IP パケットの内容を確認できます。レベル 15 では、パケットが渡されている様子を確認できます。このパラメータはオプションです。

agentctl.trace_directory = *directory*

トレース・ファイルが書き込まれるディレクトリです。この設定は、DBSNMP.TRACE_LEVEL との組合せでのみ意味を持ちます。このパラメータを省略すると、トレース・ファイルは Windows では %ORACLE_HOME%\network\trace に書き込まれます。UNIX では \$ORACLE_HOME/network/trace に書き込まれます。このパラメータはオプションです。

agentctl.trace_file = *filename*

トレース・ファイルのファイル名です。このパラメータはオプションです。

agentctl.trace_timestamp = *true/false*

true に設定すると、トレース・ファイルのトレース各行の後にタイムスタンプが挿入されます。

snmp.index.service_name.world = *index_number*

Intelligent Agent が監視しているサービスの、一意の索引番号です。索引番号には任意の番号を割り当てることができます。唯一の制限は、複数の索引行がある場合に索引番号が一意である必要があるという点です。次に例を示します。

```
snmp.index.<service_name1>=10
snmp.index.<service_name2>=20
```

snmp.connect.<service_name>.user = *user_name*

サブエージェントがデータベースへの接続に使用するユーザー名です。デフォルトは dbsnmp です。このパラメータはオプションです。このパラメータがデフォルト設定でない場合、catsnmp.sql スクリプトを編集して再実行してください。

サブエージェントは Intelligent Agent のことを指します。サーバー上で SNMP を構成するとき、SNMP Master Agent との対比で Intelligent Agent のことをサブエージェントと呼ぶ場合があります。ただし、Intelligent Agent を動作させる前にサーバー上で SNMP を構成する必要はありません (Netware プラットフォームを除く)。セキュリティ上の理由から、Intelligent Agent データベースのデフォルトのアカウント / パスワードである dbsnmp/dbsnmp を顧客が使用しない場合があります。ここに示した例は、Intelligent Agent のデータベース・ログイン・アカウントの変更を顧客が希望する場合にかぎり使用してください。

snmp.connect.<service_name>.password = *password*

サブエージェントがデータベースへの接続に使用するユーザー名に対するパスワードです。デフォルトは dbsnmp です。このパラメータはオプションです。このパラメータがデフォルト設定でない場合、catsnmp.sql スクリプトを編集して再実行してください。

サブエージェントは Intelligent Agent のことを指します。サーバー上で SNMP を構成するとき、SNMP Master Agent との対比で Intelligent Agent のことをサブエージェントと呼ぶ場合があります。ただし、Intelligent Agent を動作させる前にサーバー上で SNMP を構成する必要はありません (Netware プラットフォームを除く)。セキュリティ上の理由から、Intelligent Agent データベースのデフォルトのアカウント / パスワードである

dbsnmp/dbsnmp を顧客が使用しない場合があります。ここに示した例は、Intelligent Agent のデータベース・ログイン・アカウントの変更を顧客が希望する場合にかぎり使用してください。

dbsnmp パスワードが変更されると、自動的に暗号化されます。snmp_rw.ora ファイルのパスワード・パラメータのエントリが、解読不可能なフォーマットで表示されます。次に例を示します。

```
snmp.connect.<service>.encryptedpassword="f1d03f5c7912faa3"
```

新しいパスワードが受け入れられて暗号化されると、Agent の dbsnmp.log ファイルに "NMS-00056: データベース '<サービス>' に対するエージェントの db アカウント・パスワードは暗号化されました " が含まれます。

snmp.contact.<service_name> = "contact_info"

サービスに対する責任を持つ管理者の連絡先情報を含む文字列です。連絡先情報には名前、電話番号、電子メール・アドレスなどがあります。このパラメータはオプションです。

dbsnmp.addnl_db_conns = nn

Agent セッション・プールのサイズを増加します。Agent セッション・プールのデフォルトのサイズは、Agent が管理する ORACLE_DATABASE ターゲットの数に基づきます。接続の限界は、デフォルト値に加えて、DBSNMP.ADDNL_DB_CONNS に対して指定された値です。

dbsnmp.no_job_skipped_notifications = <false/true>

このパラメータが true の場合、スキップされたことを示す通知は送信されません。スキップの通知は、この値が false (デフォルト値) の場合にのみ送信されます。

dbsnmp.notificationtimeout = <nn>

ここで指定した時間内に、Intelligent Agent が Oracle Management Server への通知の配信に失敗した場合、Intelligent Agent は自動的に再起動されます。デフォルト値は 6 分です。<nn> はミリ秒単位です。

dbsnmp.polltime = nn

Intelligent Agent がデータベースをポーリングして、ダウンしていないかどうかを確認する時間間隔 (秒単位) です。データベースがダウンしている場合、あるいは接続されなかった場合には、この値は再試行間の間隔になります。デフォルトは 30 秒です。

注意： Intelligent Agent が 3 つ以上のインスタンスを監視する必要がある場合、監視されるインスタンスの数に比例して DBSNMP.POLLTIME の値を増やします。

次に例を示します。

Intelligent Agent が 10 個のインスタンスを監視する必要があります。その場合、DBSNMP.POLLTIME は 150 ($10/2 \times 30 = 150$) に設定します。

dbsnmp.noheuristic = {true/false}

このパラメータの値は、Intelligent Agent で、監視対象データベースの状態（データベースが起動しているか停止しているか）を確認するために監視および接続を行うかどうかを指定します。デフォルトでは、この値は FALSE（Intelligent Agent は監視および接続を行う）に設定されます。

注意： 監視対象ターゲットが Oracle Real Application Clusters データベース・インスタンスの場合、DBSNMP.NOHEURISTIC は TRUE に設定する必要があります。Oracle Real Application Clusters データベース・インスタンスには、この監視および接続が機能しないためです。

dbsnmp.trace_level = off | user | admin | nn

指定されたレベルで、Intelligent Agent プロセス (dbsnmp) に対するトレースを有効にします。オラクル社では、トレース・レベルを 13 に設定することをお薦めしています。レベル 16 は最大の情報を提供しますが、これは不具合の調査時にのみ有用です。レベル 16 では、実際の TCP/IP パケットの内容を確認できます。レベル 15 では、パケットが渡されている様子のみを確認できます。このパラメータはオプションです。

dbsnmp.trace_directory = directory

トレース・ファイルが書き込まれるディレクトリです。この設定は、DBSNMP.TRACE_LEVEL との組合せでのみ意味を持ちます。このパラメータを省略すると、トレース・ファイルは Windows では %ORACLE_HOME%\network\trace に書き込まれます。UNIX では \$ORACLE_HOME/network/trace に書き込まれます。このパラメータはオプションです。

dbsnmp.trace_file = filename

トレース・ファイルのファイル名です。このパラメータはオプションです。

dbsnmp.trace_filecnt = nn

Intelligent Agent により生成されるトレース・ファイルの最大数。完全トレースを実行することが好ましいですが、Intelligent Agent マシン上のディスク領域がかぎられている場合、

このオプション・パラメータを使用します。このパラメータは、リリース 8.1.7 の Intelligent Agent から使用可能になりました。

dbsnmp.trace_filesize = nn

Intelligent Agent により生成される個々のトレース・ファイルの最大サイズ (KB)。たとえば、値が 1024 の場合は 1MB のトレース・ファイルです。完全トレースを実行することが好ましいですが、Intelligent Agent マシン上のディスク領域がかぎられている場合、このパラメータを DBSNMP.TRACE_FILECNT とともに使用します。このパラメータは、リリース 8.1.7 の Intelligent Agent から使用可能になりました。

dbsnmp.trace_unique = true/false

true に設定すると、各ログ・エントリについて一意のログ・ファイルが生成されます。

dbsnmp.log_directory = directory

ログ・ファイルが書き込まれるディレクトリです。このパラメータはオプションです。

dbsnmp.log_file = filename

ログ・ファイルのファイル名です。このパラメータはオプションです。Windows NT では、デフォルトのファイル名は dbsnmp です。

dbsnmp.log_unique = true/false

true に設定すると、各ログ・エントリについて一意のログ・ファイルが生成されます。

dbsnmp.trace_timestamp = true/false

true に設定すると、トレース・ファイルのトレース各行の後にタイムスタンプが挿入されます。

dbsnmp.hostname = <hostname or ip address>

Agent を認識しているホスト名をオーバーライドします。IP アドレスが指定されている場合、Intelligent Agent はこのパラメータによってそのマシン上の特定のネットワーク・インタフェース・カードにバインドされます。また、ホスト名が指定されている場合、Intelligent Agent はこのパラメータによってそのマシン上のすべてのネットワーク・インタフェース・カードにバインドされます。

返される文字列はマシン名であり、暗号の変換に使用されます。また、DBSNMP.HOSTNAME は、RPC のリスニングに使用するデフォルトのホストをオーバーライドします (dbsnmp.address および dbsnmp.spawnaddress)。

dbsnmp.threshold_job_status

Intelligent Agent によって Oracle Management Server にキューされるジョブ通知の数を設定します。

dbsnmp.threshold_evocc

OMS にキューされるイベント通知の数を設定します。

dbsnmp.avg_occ_per_event

各イベントごとに保存する通知の数を設定します。通知がこの数を超えるとキューはクリアされます。

dbsnmpj.trace_level = off | user | admin | nn

指定されたレベルで、*dbsnmpj* プロセスに対するトレースを有効にします。*dbsnmpj* は、Intelligent Agent にジョブが発行される際に必ず開始されます。オラクル社では、トレース・レベルを 13 に設定することをお薦めしています。レベル 16 は最大の情報を提供しますが、これは不具合の調査時のみ有効です。レベル 16 では、実際の TCP/IP パケットの内容を確認できます。レベル 15 では、パケットが渡されている様子のみを確認できます。このパラメータはオプションです。

dbsnmpj.trace_directory = directory

トレース・ファイルが書き込まれるディレクトリです。この設定は、DBSNMP.TRACE_LEVEL との組合せでのみ意味を持ちます。このパラメータを省略すると、トレース・ファイルは Windows では %ORACLE_HOME%\network\trace に書き込まれます。UNIX では \$ORACLE_HOME/network/trace に書き込まれます。このパラメータはオプションです。

dbsnmpj.trace_file = filename

トレース・ファイルのファイル名です。このパラメータはオプションです。

dbsnmpj.trace_filecnt = nn

Intelligent Agent により生成されるトレース・ファイルの最大数。完全トレースを実行することが好ましいですが、Intelligent Agent マシン上のディスク領域がかぎられている場合、このオプション・パラメータを使用します。このパラメータは、リリース 8.1.7 の Intelligent Agent から使用可能になりました。

dbsnmpj.trace_filecnt = nn

Intelligent Agent により生成される個々のトレース・ファイルの最大サイズ (KB)。たとえば、値が 1024 の場合は 1MB のトレース・ファイルです。完全トレースを実行することが好ましいですが、Intelligent Agent マシン上のディスク領域がかぎられている場合、このパラメータを DBSNMP.TRACE_FILECNT とともに使用します。このパラメータは、リリース 8.1.7 の Intelligent Agent から使用可能になりました。

dbsnmpj.trace_unique = true/false

true に設定すると、トレース各行について一意のトレース・ファイルが生成されます。

dbsnmpj.log_directory = directory

ログ・ファイルが書き込まれるディレクトリです。このパラメータはオプションです。

dbsnmpj.log_file = filename

ログ・ファイルのファイル名です。このパラメータはオプションです。Windows NT では、デフォルトのファイル名は *dbsnmpj* です。

dbsnmpj.log_unique = true/false

true に設定すると、各ログ・エントリについて一意のログ・ファイルが生成されます。

dbsnmpj.trace_timestamp = true/false

true に設定すると、トレース・ファイルのトレース各行の後にタイムスタンプが挿入されます。

注意： 次のアドレスは、Intelligent Agent によって自動的に設定されます。アドレスを変更すると、Intelligent Agent は Enterprise Manager コンソールによって検出できなくなり、手動での構成設定が必要になります。

**dbsnmp.address = (address=(protocol= <protocol>) (host=<host_name>)
(port=<port_no>))**

Intelligent Agent が、送られてくるリクエストのリスニングに使用する TNS アドレスです。アドレスには空白または改行文字を含めないようにしてください。このパラメータは、Intelligent Agent がネットワーク接続に対してリスニングを行うアドレスです。

Intelligent Agent がサービスを自動検出するためには TCP/IP が必須であるため、サーバー上に TCP/IP がインストールされている必要があります。

Intelligent Agent は、PORT=1748 をリクエストします。ポート・アドレス 1748 は、Internet Assigned Number Authority (IANA) によって Oracle に付与された登録済の TCP ポートです。ポート・アドレスは自動的に設定されます。このポートを変更すると、Intelligent Agent は Enterprise Manager コンソールによって検出できなくなり、手動での構成設定が必要になります。

**dbsnmp.spawnaddress = (address= (protocol=<protocol>)
(host=<host_name>)(port=<spnport_no>))**

Intelligent Agent が RPC を受け入れるために使用できる TNS アドレスです。このアドレスは、ファイル転送に使用されます。このパラメータで使用する spnport_no は、DBSNMP.ADDRESS パラメータで使用する port_no とは異なります。

Intelligent Agent は、PORT=1754 をリクエストします。ポート・アドレス 1754 は、Internet Assigned Number Authority (IANA) によって Oracle に付与された登録済の TCP ポートです。このポートを変更すると、Intelligent Agent は Enterprise Manager コンソールによって検出できなくなり、手動での構成設定が必要になります。

dbsnmp.cs_base_port = <port number>

Intelligent Agent の収集サービスによって使用されるデフォルトのポート番号をオーバーライドする際に使用するポート番号。たとえば、ポート番号 1700 を指定すると、Intelligent Agent はポート番号 1700 および 1701 を使用します。

Intelligent Agent のログ・ファイル

適切なログ・ファイル・パラメータが設定されている場合、次のログ・ファイルが生成されます。

表 A-1 Intelligent Agent のログ・ファイル

ログ・ファイル	説明
dbsnmp.log	Intelligent Agent の全プロセスのログ
dbsnmpj.log	Intelligent Agent により処理された全ジョブのログ
dbsnmp.nohup (UNIX)	Intelligent Agent の自動再起動 (dbsnmpwd) のログ
<agent service name>.log (Windows NT)	
agentsvc.log (Windows NT)	agentctl 機能のログ (Intelligent Agent の起動時に発生したエラーも示す)
nmiconf.log	検出情報およびエラーのログ

トラブルシューティング

この章では、Intelligent Agent が正しく機能しないときの一般的なトラブルシューティングについて説明します。次の項目について説明します。

- [Intelligent Agent のトラブルシューティング](#)
- [簡単なチェック](#)
- [追加のチェック](#)
- [Intelligent Agent のエラー・メッセージと解消](#)
- [リリース 9.x の Intelligent Agent のトレース](#)
- [Windows NT/2000 での Intelligent Agent の起動時エラーの理解](#)
- [Data Gatherer の理解およびトラブルシューティング](#)
- [Intelligent Agent の初期化](#)
- [UNIX での Intelligent Agent の検出エラーの診断](#)
- [Window NT での Intelligent Agent の検出エラーの診断](#)

Intelligent Agent のトラブルシューティング

ほとんどの状況下において、Intelligent Agent 自体が必要とする構成の手段はごくわずかです。ただし、正しく機能するためには、Intelligent Agent が管理側のホストおよび管理対象のサービスと通信できる必要があります。Oracle および使用しているオペレーティング・システムの知識が豊富なユーザーであれば、次の簡単なチェックリストを利用して、Intelligent Agent の運用を妨げる可能性のある問題を高い確率で解決できます。

重要： Intelligent Agent はリリースを重ねるごとに継続的に改善されているため、ご使用の特定のサーバー・リリースで使用可能な最新の Intelligent Agent にアップグレードすることを**強くお勧めします**。多くの場合、アップグレードにより旧バージョンの Intelligent Agent で発生した問題が解消します。一般的に、Intelligent Agent の受入れ可能な最も低いバージョンは、監視されるソフトウェアの最も高いバージョンとの互換性を持っている必要があります。

簡単なチェック

次のチェックリストでは、Intelligent Agent の運用に最も影響する領域をカバーしています。Intelligent Agent のトラブルシューティング・チェックリストは2種類あり、それぞれ、Intelligent Agent が動作する最も一般的なプラットフォームである Windows NT と UNIX を対象としています。チェックリストは簡略化されており、利用にあたっては、Oracle とオペレーティング・システムの両方、そして関連する通信プロトコルについての知識が前提となります。それぞれのトラブルシューティング手順の詳細は、この章の後の部分で詳細に説明します。

Windows NT での Intelligent Agent の簡単なチェック

Windows NT システム上で Intelligent Agent を実行している場合、次のチェックリストを使用します。

1. 「コントロール パネル」で OracleAgent サービスの状態を調べ、Intelligent Agent サービスが開始されていることを確認します。Intelligent Agent が開始されていない場合、次のヒントのいずれかを参考にして、問題を解決します。
2. Windows NT の「イベント ビューア」(「管理ツール」の下) に書き込まれたメッセージを確認します。「イベント ビューア」には、起動に関連するすべての問題が Windows NT の Intelligent Agent によって書き込まれます。
3. snmp_ro.ora、snmp_rw.ora および services.ora の各ファイルが、起動時に Intelligent Agent によって作成されていることを確認します。snmp_ro.ora および snmp_rw.ora は、%ORACLE_HOME%\network\admin ディレクトリにあります。services.ora は、%ORACLE_HOME%\network\agent ディレクトリにあります。

リストされたサービスを、マシン上で使用可能なサービスと比較します。有効なサンプル・ファイルの例は、付録 A「Intelligent Agent 構成ファイル」を参照してください。

欠落しているサービスがある場合、次のファイル間の不一致、またはファイルの破損がないかどうかを調べます。

- listener.ora
- tnsnames.ora

4. データベース・ソフトウェアがアップグレードされており、使用中のマシンのいずれかについて、生成された snmp_ro.ora、snmp_rw.ora または services.ora の各ファイルに問題がある場合、次の指示に従います。

- a. (dbsnmp アカунツではなく) SYS アカウンツで catsnmp.sql を実行します。通常、catsnmp.sql スクリプトはデータベースの作成時に catalog.sql の内部で実行されますが、データベースはアップグレードされているため、このスクリプトがまだ実行されていない可能性があります。必要なスクリプトを実行しないと、dbsnmp アカウンツは作成されません。

重要： リリース 9.x 以前のデータベースを実行しているマシンにリリース 9.x の Intelligent Agent をインストールする際、リリース 9.x の Enterprise Manager クライアント・インストールの \$ORACLE_HOME/sysman/admin ディレクトリ内にある、バージョン固有の catsnmp.sql のコピーを再実行する必要があります。たとえば、マシンで Oracle 8.1.7 を実行していて、リリース 9.x の Intelligent Agent をインストールする場合、新しい Intelligent Agent をインストールした後に catsnmp_8i.sql スクリプトを再実行する必要があります。この操作は、この Intelligent Agent によってサービスされるリリース 9.x 以前の各データベースに対して行われる必要があります。

リリース 9.x の catsnmp.sql スクリプトを、リリース 9.x 以前のデータベースに対して実行しないでください。

- b. oratab ファイルで複数の SID、または古い SID が参照されている場合、それぞれのデータベースに対して catsnmp.sql を実行します。
- c. snmp_ro.ora は読み込み専用ファイルであるため、このファイルへのすべての変更は、Intelligent Agent を起動するたびに上書きされます。必要であれば、snmp_rw.ora ファイルに変更を加えることができます。

バックアップを実行する場合は、dbsnmp/dbsnmp アカウンツで backup.ts.tcl を実行する必要があります。

警告： Intelligent Agent に付属の Tcl スクリプト（Tcl で記述されるジョブ・スクリプトおよびイベント・スクリプト）は変更しないでください。Intelligent Agent に定義済みのもの以外のジョブを送る場合は、Tcl ジョブを使用します。このジョブでは、任意のスクリプトを渡して Intelligent Agent に実行させることができます。

5. 外部ドライブに設定されているシステム・パスがないことを確認します。

Intelligent Agent は1つのサービスであり、デフォルトでは SYSTEM 権限で動作します。また、%ORACLE_HOME%\BIN ディレクトリにある DLL も必要とします。ドライブをパス内にマップする必要がある場合、それらを SYSTEM パスの内部に設定しないでください。

独自のパスを設定する手順は、次のとおりです。

- a. マップされるドライブのパスを、SYSTEM パス変数から独自のパスに移動します。
- b. 再起動して、システム・パスの設定を解除します。

6. Intelligent Agent が起動しない理由がこの時点でわからない場合、Intelligent Agent をトレースします。

- a. snmp_rw.ora で、次のように変数を設定します。

dbsnmp.trace_level=admin (16 に設定すると最も多くの情報を得られます)

dbsnmp.trace_directory=<Oracle ユーザーが書き込み権限を持つ任意のディレクトリ>

dbsnmp.trace_file=<トレース出力ファイルの名前>

dbsnmp.trace_unique=true/false (TRUE の場合、このパラメータは各回に一意的なトレース・ファイルを作成します。)

- b. Intelligent Agent を再起動します。

- c. %ORACLE_HOME%\network\log ディレクトリにあるログ・ファイルを確認します。

DBSNMP.LOG は、Intelligent Agent の一般的な問題を示します。

DBSNMP.NOHUP は、Intelligent Agent の監視プロセスである dbsnmpwd に関連するエラーを示します。

NMICONF.LOG は、自動検出に伴う問題を示します。

7. DNS ホストのエントリが、listener.ora および tnsnames.ora の各ファイルで指定されたノード名に設定されていることを確認します。

- a. 「スタート」>「設定」>「コントロールパネル」>「ネットワーク」>「プロトコル」>「TCP/IP プロトコル」を選択し、「プロパティ」をクリックします。

- b. DNS ホストのエントリを確認します。
8. TCP/IP がインストールされていることを確認します。TCP/IP は必要条件です。

UNIX での Intelligent Agent の簡単なチェック

UNIX システム上で Intelligent Agent を実行している場合、次のチェックリストを利用します。

1. 必要なシステム要件を満たしていることを確認してください。
 - TCP/IP が正しく設定されている
 - Oracle 環境の ORATAB ファイルが有効である
2. Intelligent Agent のステータスを確認します。次のコマンドを入力します。

```
agentctl status
```

または、次のコマンドを入力して、Intelligent Agent が動作していることを確認することもできます。

```
ps -eaf | grep dbsnmp
```

このチェックでは、dbsnmp プロセスが実行中であること、または dbsnmpwd 監視スクリプトが実行中であること（あるいはその両方）が表示されるはずです。

3. \$ORACLE_HOME/network/log/dbsnmp*.log ファイルを参照して、UNIX システムのエラーを確認します。（エラー検索の場合は、nmiconf.log ファイル）。
4. Oracle ユーザーが、次のディレクトリへの書き込み許可を持っていることを確認します。
 - \$ORACLE_HOME/network/log
 - \$ORACLE_HOME/network/trace
 - \$ORACLE_HOME/network/agent
 - \$ORACLE_HOME/network/agent/jobout

5. Intelligent Agent により作成されたエントリの snmp_ro.ora、snmp_rw.ora および services.ora を確認します。snmp_ro.ora および snmp_rw.ora は、\$ORACLE_HOME/network/admin ディレクトリにあります。services.ora は、\$ORACLE_HOME/network/agent ディレクトリにあります。または、環境変数 TNS_ADMIN で指定されているディレクトリをチェックすることもできます。

リストされたサービスを、マシン上で使用可能なサービスと比較します。有効なサンプル・ファイルの例は、[付録 A 「Intelligent Agent 構成ファイル」](#)を参照してください。

欠落しているサービスがある場合、次のファイル間の不一致、またはファイルの破損がないかどうかを調べます。

- listener.ora
 - tnsnames.ora
 - oratab
6. データベース・ソフトウェアがアップグレードされており、使用中のマシンのいずれかについて、生成された `snmp_ro.ora`、`snmp_rw.ora` または `services.ora` の各ファイルに問題がある場合、次の指示に従います。

- a. (`db snmp` アカウントではなく) `SYS` アカウントで `catsnmp.sql` を実行します。通常、`catsnmp.sql` スクリプトはデータベースの作成時に `catalog.sql` の内部で実行されますが、データベースはアップグレードされているため、このスクリプトがまだ実行されていない可能性があります。必要なスクリプトを実行しないと、`db snmp` アカウントは作成されません。

重要： リリース 9.x 以前のデータベースを実行しているマシンにリリース 9.x の `Intelligent Agent` をインストールする際、リリース 9.x の `Enterprise Manager` クライアント・インストールの `$ORACLE_HOME/sysman/admin` ディレクトリ内にある、バージョン固有の `catsnmp.sql` のコピーを再実行する必要があります。たとえば、マシンで `Oracle 8.1.7` を実行していて、リリース 9.x の `Intelligent Agent` をインストールする場合、新しい `Intelligent Agent` をインストールした後に `catsnmp_8i.sql` スクリプトを再実行する必要があります。この操作は、この `Intelligent Agent` によってサービスされるリリース 9.x 以前の各データベースに対して行われる必要があります。

リリース 9.x の `catsnmp.sql` スクリプトを、リリース 9.x 以前のデータベースに対して実行しないでください。

- b. `oratab` ファイルで複数の `SID`、または古い `SID` が参照されている場合、それぞれのデータベースに対して `catsnmp.sql` を実行します。
- c. `snmp_ro.ora` は読み込み専用ファイルであるため、このファイルへのすべての変更は、`Intelligent Agent` を起動するたびに上書きされます。必要であれば、`snmp_rw.ora` ファイルに変更を加えることができます。

バックアップを実行する場合は、`db snmp/db snmp` アカウントで `backupts.tcl` を実行する必要があります。

警告： `Intelligent Agent` に付属の `Tcl` スクリプト (`Tcl` で記述されるジョブ・スクリプトおよびイベント・スクリプト) は変更しないでください。`Intelligent Agent` に定義済みのもの以外のジョブを送る場合は、`Tcl` ジョブを使用します。このジョブでは、任意のスクリプトを渡して `Intelligent Agent` に実行させることができます。

7. Intelligent Agent 制御ユーティリティ (agentctl) の実行に問題がある場合は、agentctl のトレースを次のように設定します。
 - agentctl.trace_level=admin
 - agentctl.trace_directory
 - agentctl.trace_file
8. Intelligent Agent が起動しない理由がこの時点でわからない場合、snmp_rw.ora で次のように変数を設定して Intelligent Agent をトレースし、Intelligent Agent を再起動します。
 - dbsnmp.trace_level=admin (16 に設定するとより多くの情報を得られます)
 - dbsnmp.trace_directory=<Oracle ユーザーが書き込み許可を持つ任意のディレクトリ >
 - dbsnmp.trace_file=agent

追加のチェック

簡単なチェックで Intelligent Agent の問題を解決できなかった場合、次の項以降で各領域について記載された事項を参考にして、Intelligent Agent の運用に関する問題の原因を突き止めてください。また、チェックリストに記された手順の多くについて、Oracle または Intelligent Agent が動作しているオペレーティング・システム、あるいはその両方にそれほど詳しくないユーザーのために、より詳細に説明しています。この項では、次のような問題について説明します。

- [TCP/IP の構成と動作の確認 \(B-8 ページ\)](#)
- [DNS 名およびコンピュータ名の一致の確認 \(Windows NT\) \(B-10 ページ\)](#)
- [Oracle Net 構成ファイルの確認 \(B-10 ページ\)](#)
- [Oracle Net の動作確認 \(B-11 ページ\)](#)
- [Intelligent Agent の正常起動の確認 \(B-12 ページ\)](#)
- [Intelligent Agent がノード上の全インスタンスに接続していることの確認 \(B-14 ページ\)](#)
- [Intelligent Agent が正しい許可で動作していることの確認 \(UNIX\) \(B-15 ページ\)](#)
- [オペレーティング・システム・ユーザーが存在し、正しい許可を持っていることの確認 \(Windows NT\) \(B-15 ページ\)](#)

注意： Intelligent Agent のデバッグを行う際、
\$ORACLE_HOME/network/agent ディレクトリ内のすべての .q ファイル
を削除する必要はありません。以前はこの方法をお薦めしていましたが、
最近のバージョンの Intelligent Agent のトラブルシューティングでは、こ
の操作は必要ありません。このルールには例外もありますが、詳細はこの
章の後の部分で説明します。

TCP/IP の構成と動作の確認

Intelligent Agent の起動に際して障害となる最も一般的な問題の 1 つは、TCP/IP の構成で
す。TCP/IP 設定が正しく構成されていることを確認するには、コマンドラインで次のよう
に入力します。

- ホスト・マシン (Intelligent Agent が動作するマシン) および指定されたネットワーク
IP アドレスが、同じマシンを指していることを確認します。コマンドラインで、次のよ
うに入力します。
 1. ping <ホスト名>
 2. ping <IP アドレス>
 3. 2 つのコマンドによって、同じ情報が返されることを確認します (Windows NT)。
UNIX システムでは、2 つのコマンドに対してそれぞれ「<hostname> is alive」、
「<IP address> is alive」と表示されることを確認します。
- 次のコマンドを実行して、ホスト・マシンに到達可能なことを確認します。
telnet <ホスト名>
- ホスト・マシンがローカル・ネットワーク上で有効なホストであること、また、
Management Server を実行しているマシンからホスト・マシンへのアクセスが可能であ
ることを確認します。
 1. Intelligent Agent が動作しているマシン上で、そのマシン自体の IP アドレスへの
ping を実行します。
 2. Management Server を実行しているマシン上で、Intelligent Agent が動作している
マシンの IP アドレスへの ping を実行します。
- Management Server がローカル・ネットワーク上で有効なホストであること、また、
Intelligent Agent を実行しているマシンが Management Server を実行しているマシンと
通信できることを確認します。
 1. ping <コンソール・マシンの IP アドレス>
Management Server が動作しているマシン上で、そのマシン自体への ping を実行
します。

2. ping <Management Server の IP アドレス>

Intelligent Agent が動作しているマシン上で、Management Server への ping を実行します。

3. これらの 2 つのコマンドにより、同じ情報が返されることを確認します。

注意： Windows NT システムのホスト名は、コマンド・プロンプトで hostname と入力することで確認できます。

TCP/IP 構成上の問題の修正

1. (Windows NT) WINNT¥system32¥drivers¥etc ディレクトリ内の、hosts ファイルおよび lmhosts ファイルを編集します。

これらのファイルが使用されていない場合、ディレクトリ内にはサンプル・ファイルのみが存在します。.sam ファイルの名前を変更するか、またはファイルをコピーして、拡張子なしの hosts ファイルおよび lmhosts ファイルを作成します。

(UNIX) root でログインして、/etc/hosts ファイルを編集します。

NIS/DNS セットアップでは、TCP/IP 構成の問題を修正するために NIS/DNS レベルでの変更が必要になることがあります。

2. 各システムの IP アドレスとホスト情報が正しいことを確認します。

例：(Windows NT)

HOSTS file:

```
122.111.111.111    myHost
```

LMHOSTS file:

```
122.111.111.111    myHost    #PRE
```

注意： この情報は、Windows NT の「コントロール パネル」>「ネットワーク」プロパティで修正することもできます。

3. \$ORACLE_HOME/network/agent ディレクトリ内の、*.q ファイルおよび services.ora ファイルを削除します。

注意： *.q ファイルは、Intelligent Agent の状態、現行のジョブおよびイベントに関する情報を含むバイナリ・ファイルです。これらのファイルを削除する前に、Intelligent Agent に対して登録されているすべてのジョブとイベントを削除するようにしてください。q ファイルの暗号化に、Agent が動作しているマシンのホスト名が使用されるため、これらのファイルのあるシステムから他のシステムにコピーすることはできません。

4. \$ORACLE_HOME/network/admin/snmp_ro.ora および \$ORACLE_HOME/network/admin/snmp_rw.ora ファイルを削除します。
5. Intelligent Agent を再起動します。

DNS 名およびコンピュータ名の一致の確認 (Windows NT)

リリース 8.0.4 より前の Intelligent Agent では、DNS ホスト名とコンピュータ名が一致している必要がありました。これらのパラメータは、Windows NT の「コントロールパネル」の、次のプロパティ・シートで確認または変更できます。

コンピュータ名を確認するには次のようにします。

- 「コントロール パネル」>「ネットワーク」>「識別」>「コンピュータ名」

DNS 名を確認するには次のようにします。

- 「コントロール パネル」>「ネットワーク」>「プロトコル」>「TCP/IP プロトコル」>「プロパティ」>「DNS」>「ホスト名」

Oracle Net 構成ファイルの確認

ネットワークが適切に構成され、ネットワーク内のノードが互いに通信できることに加えて、Oracle 環境のコンポーネントが互いに通信する必要があります。Oracle Net Services は、クライアント・マシンと Oracle サーバーの間、あるいは複数の Oracle サーバー間に、セッションおよびデータ通信の媒体を提供します。したがって、Intelligent Agent が通信を行うためには、Oracle Net の適切な構成が前提条件となります。この項では、Intelligent Agent の通信が失敗するときに起こりうる最も一般的な問題について説明します。

Oracle Net の構成ファイルは、\$TNS_ADMIN か、%ORACLE_HOME%\network\admin (Windows NT) または \$ORACLE_HOME/network/admin (UNIX) の各ディレクトリにあります。

主要な構成ファイルには、次のようなものがあります。

- listener.ora
- sqlnet.ora
- tnsnames.ora

これらのファイルの詳細および設定例は、付録 A 「Intelligent Agent 構成ファイル」を参照してください。

Intelligent Agent による検出の間の TNS_ADMIN 変数の使用方法

(UNIX) すべてのバージョンの UNIX 検出スクリプトで、TNS_ADMIN 変数を使用して入力ファイル (listener.ora および tnsnames.ora) の場所を特定できます。TNS_ADMIN が設定される場合、リリース 7.3.4 以上の Intelligent Agent でのみ、出力ファイル (snmp_ro.ora および snmp_rw.ora) を TNS_ADMIN に正しく書き込むことができます。

(Windows NT) リリース 8.0.5 以上では、検出スクリプトにより TNS_ADMIN の値が NT レジストリから併せて読み込まれます。

Intelligent Agent では、listener.ora ファイル中の TNS 別名情報も使用されます。Intelligent Agent のこの動作は、Oracle Names 環境の内部でも同じです。この動作は意図的なものです。これは、Oracle Names Server は一時的に使用不可になる可能性がある一方で、Intelligent Agent は常に名前を解決できる必要があるという理由によります。次の手順で、TNS 別名がローカルでどのように変換されるかを確認してください。

1. listener.ora ファイルに、ターゲットに対して構成された TCP エントリが 1 つ以上含まれていることを確認します。

Intelligent Agent はポート 1748 上でリスニングするため、このポート上でリスナーをアクティブ化しないでください。(Intelligent Agent に対して TNSPING を使用できるのはこのためです。TNSPING はリスナーと Intelligent Agent を区別できません。)

2. DNS ホストのエントリが、listener.ora ファイルおよび tnsnames.ora ファイル中のノード名に設定されていることを確認します。
 1. Windows NT のタスクバーから、「スタート」>「設定」>「コントロール パネル」をクリックします。
 2. 「ネットワーク」アイコンをダブルクリックします。
 3. 「プロトコル」タブをクリックします。
 4. 「TCP/IP プロトコル」を選択して「プロパティ」をクリックします。
 5. DNS ホストのエントリを確認します。

Oracle Net の動作確認

Oracle Net の構成が正しいにもかかわらず、依然として Intelligent Agent と通信できない場合、次の手順として、Oracle Net 内のどのサービスが到達可能であるかを調べます。コマンド・プロンプトで次のように入力することにより、アクセス先の各データベース上で TNSPING ユーティリティを使用できます。

```
tnsping <network service name>
```

TNSPING を使用して、クライアントからサーバーに（またはサーバー間で）正しく接続できる場合、前述のコマンドにより、Oracle Net Services への到達にかかる推定の往復時間（ミリ秒単位）が返されます。これは、Oracle Net が正常に動作していることを表しています。

Intelligent Agent の ping

Intelligent Agent が起動しているかどうかを確認するには、次を実行します。

```
$ tnsping (address=(protocol=tcp) (host=<<hostname>>) (port=1748))
```

Intelligent Agent への接続のテスト

TNSPING ユーティリティを使用して Intelligent Agent への接続をテストすることもできます。

```
$ tnsping " (address=(protocol=tcp) (host=<<hostname>>) (port=1748)) "
```

接続が正常である場合、次のようなメッセージが表示されます。

```
Attempting to contact (address=(protocol=tcp) (host=<<hostname>>) (port=1748))
OK (750 msec)
```

invited_nodes リストにないホストがノードに接続しようとすると、出力に次のようなエラーが表示されます。

```
Attempting to contact (address=(protocol=tcp) (host=<<hostname>>) (port=1748))
TNS-12547: TNS: lost contact
```

Intelligent Agent の正常起動の確認

次のコマンドを発行して、Intelligent Agent のプロセスが動作していることを確認します。

```
agentctl status
```

Intelligent Agent が開始されていない場合、次の表のヒントを参考にして、問題を解決してください。

表 B-1 Intelligent Agent が起動しない場合のトラブルシューティング

UNIX	Windows NT
ファイル \$ORACLE_HOME/network/log/dbsnmp*.log を参照して、エラーを確認します。	Windows NT の「イベント ビューア」（「管理 ツール」の下）に書き込まれたメッセージを確認します。「イベント ビューア」には、起動に関連するすべての問題が Windows NT の Intelligent Agent によって書き込まれます。

表 B-1 Intelligent Agent が起動しない場合のトラブルシューティング (続き)

UNIX	Windows NT
ファイル \$ORACLE_HOME/network/log/nmiconf.log を参照して、エラーを確認します。 Oracle ユーザーが、次のディレクトリへの書き込み許可を持っていることを確認します。 \$ORACLE_HOME/network/log	ファイル %ORACLE_HOME%\network\log\nmiconf.log を参照して、エラーを確認します。 Intelligent Agent サービスのプロパティで、Intelligent Agent によって使用されるオペレーティング・システム・アカウント（デフォルトでは System）を確認します。Intelligent Agent ユーザーが、次のディレクトリへの書き込み許可を持っていることを確認します。 %ORACLE_HOME%\network\log
snmp_ro.ora、snmp_rw.ora および services.ora の各ファイルを参照して、Intelligent Agent によって作成されたエントリを確認します。snmp_ro.ora および snmp_rw.ora は、\$ORACLE_HOME/network/admin ディレクトリにあります。services.ora は、\$ORACLE_HOME/network/agent ディレクトリにあります。	snmp_ro.ora、snmp_rw.ora および services.ora の各ファイルが、起動時に Intelligent Agent によって作成されていることを確認します。snmp_ro.ora および snmp_rw.ora は、%ORACLE_HOME%\network\admin ディレクトリにあります。services.ora は、%ORACLE_HOME%\network\agent ディレクトリにあります。
リストされたサービスを、マシン上で使用可能なサービスと比較します。有効なサンプル・ファイルの例は、付録 A を参照してください。欠落しているサービスがある場合、次のファイル間の不一致、またはファイルの破損がないかどうかを調べます。	リストされたサービスを、マシン上で使用可能なサービスと比較します。有効なサンプル・ファイルの例は、付録 A を参照してください。欠落しているサービスがある場合、次のファイル間の不一致、またはファイルの破損がないかどうかを調べます。
<ul style="list-style-type: none">■ listener.ora■ tnsnames.ora■ oratab	<ul style="list-style-type: none">■ listener.ora■ tnsnames.ora
TCP/IP がインストールされていることを確認します。TCP/IP は必要条件です。TCP/IP が正しく構成され、動作しているかどうかを確認します。	TCP/IP がインストールされていることを確認します。TCP/IP は必要条件です。TCP/IP が正しく構成され、動作しているかどうかを確認します。

表 B-1 Intelligent Agent が起動しない場合のトラブルシューティング（続き）

UNIX	Windows NT
Intelligent Agent が起動しない理由がこの時点でわからない場合、トレースを有効にします。（「リリース 9.x の Intelligent Agent のトレース」を参照してください。）	システム・パス変数に外部ドライブが含まれていないことを確認します。Intelligent Agent は 1 つのサービスであり、デフォルトでは SYSTEM 権限で動作します。また、 %ORACLE_HOME%\bin ディレクトリにある DLL も必要とします。パス内に外部ドライブをマップする必要がある場合、それらのドライブを SYSTEM パスの内部に設定しないでください。独自のパスを設定する手順は、次のとおりです。 <ol style="list-style-type: none">1. マップする外部ドライブのパスを、システム・パス変数から独自のパスに移動します。2. 再起動して、システム・パスの設定を解除します。 %ORACLE_HOME%\network\log\AGENTSRVC.log ファイルを参照してください。このファイルは、Intelligent Agent サービスの起動時に発生したエラーについて示します。 Intelligent Agent が起動しない理由がこの時点でわからない場合、トレースを有効にします。Intelligent Agent のトレース設定の詳細は、B-22 ページの「 リリース 9.x の Intelligent Agent のトレース 」を参照してください。

UNIX および Windows NT システムのいずれでも、次を確認します。

```
$ORACLE_HOME/network/log/dbsnmp.nohup
```

Intelligent Agent がノード上の全インスタンスに接続していることの確認

Intelligent Agent が、あるノード上で監視するデータベースに接続できることをテストするには、次の接続文字列を使用して、各データベースへの接続を試みます。

```
dbsnmp/dbsnmp@address_list
```

このテストは、Intelligent Agent が動作するノード上で実行する必要があります。

Intelligent Agent が正しい許可で動作していることの確認 (UNIX)

Intelligent Agent が正しいユーザー許可を持っていることを検証する方法については、2-2 ページの「[Intelligent Agent のインストール](#)」を参照してください。

オペレーティング・システム・ユーザーが存在し、正しい許可を持っていることの確認 (Windows NT)

ノードに対してオペレーティング・システム・ユーザーを指定し、次の許可を付与する必要があります。

- %ORACLE_HOME%\network ディレクトリと、そのすべてのサブディレクトリに対する読み込みおよび書き込み許可
- TEMP ディレクトリに対する読み込みおよび書き込み許可 (Windows NT の TEMP ディレクトリは、「コントロール パネル」>「システム」を選択して確認できます)。TEMP が定義されていない場合、Oracle ホーム・ディレクトリに対する読み込みおよび書き込み許可をオペレーティング・システム・ユーザーが持っている必要があります。オペレーティング・システム・ユーザーは、Oracle ホーム・ディレクトリの下に work という名前で一時ディレクトリを作成します。

エラーの有無の確認

(Windows NT) 「イベント ビューア」>「ログ」>「アプリケーション」で、DBSNMP プロセスの起動に関するエラーを確認します。

(Windows NT および UNIX) \$ORACLE_HOME/network/log/nmiconf.log ファイルを参照して、検出エラーの有無を確認します。

UNIX および Windows NT システムのいずれでも、次のファイルで追加のエラーを確認します。

```
$ORACLE_HOME/network/log/dbsnmp.nohup
```

Intelligent Agent のエラー・メッセージと解消

次に示すエラー・メッセージと解消は、オペレーティング・システム別に分類されています。すべてのシステムに共通の解消は、「Intelligent Agent 全般」の項にリストしてあります。

Intelligent Agent 全般

ジョブ実行時の「ユーザー認証に失敗しました」エラー 管理対象のノード上で Intelligent Agent がジョブを実行するために必要な条件は、次のとおりです。

- 高度なユーザー権利「バッチ ジョブとしてログオン」を持つ Windows NT ユーザー・アカウントが存在すること。(Windows NT)。権限は既存のローカル・ユーザーまたはドメイン・ユーザー (リリース 7.3.3 以上)、あるいは新規 Windows NT ユーザーに割り当てることができます。[第 2 章「インストール、構成および使用方法」](#)の、Windows NT に固有の説明箇所を参照してください。
- Oracle Enterprise Manager コンソールで、ノードに対する優先接続情報が前述のユーザーに対して設定されていること。
- ユーザーが \$ORACLE_HOME/network ディレクトリ、または %ORACLE_HOME%\network ディレクトリへの書き込み許可を持っていること。
- (UNIX) 高度なセキュリティ環境で使用されているシャドウ・パスワード・ファイルを同期していること。Intelligent Agent はローカルのシャドウ・パスワード・ファイルを確認して認証情報を探します。高度なセキュリティ・システム上でパスワードを同期する方法についての詳細は、オペレーティング・システム固有のドキュメントを参照してください。

トレース・ファイル中の「ログインは拒否されました」、「ユーザー名 / パスワードが無効です」の各メッセージ これらのメッセージは、マシン上にリリース 7.3.3 より前のデータベースが存在する場合に書き込まれます。リリース 7.3.3 以上では、CATSNMP.SQL という名前のスクリプトが、ディクショナリ・スクリプト CATALOG.SQL に組み込まれています。このスクリプトには、Intelligent Agent が接続のために必要とする DBSNMP ユーザーを作成する役割があります。リリース 7.3.3 より前のデータベースには、このスクリプトはありません。

ユーザー DBSNMP が存在することを確認してください。存在しない場合は、catsnmp.sql スクリプトを実行してください。

重要： リリース 9.x 以前のデータベースを実行しているマシンにリリース 9.x の Intelligent Agent をインストールする際、リリース 9.x の Enterprise Manager クライアント・インストールの \$ORACLE_HOME/sysman/admin ディレクトリ内にある、バージョン固有の catsnmp.sql のコピーを再実行する必要があります。たとえば、マシンで Oracle 8.1.7 を実行していて、リリース 9.x の Intelligent Agent をインストールする場合、新しい Intelligent Agent をインストールした後に catsnmp_8i.sql スクリプトを再実行する必要があります。この操作は、この Intelligent Agent によってサービスされるリリース 9.x 以前の各データベースに対して行われる必要があります。

リリース 9.x の catsnmp.sql スクリプトを、リリース 9.x 以前のデータベースに対して実行しないでください。

Intelligent Agent 起動時の「ORACLE_HOME は存在しません」 このメッセージは、検出スクリプト nmiconf.tcl によるものです。環境変数 \$ORACLE_HOME が Intelligent Agent の ORACLE_HOME に設定されていることを確認し、Intelligent Agent を再起動してください。

Intelligent Agent が 1 ノード上で 1 つのデータベースしか検出しない 1 つのノード上に複数のデータベースが存在する場合、次のいずれかを指定して各インスタンスが一意のデータベース名を持っていることを確認します。

- 一意の SID
- データベースの TNSNAMES.ORA における一意の別名
- LISTENER.ORA ファイルでの一意の GLOBAL_DBNAME

snmp_ro.ora および snmp_rw.ora が生成されていない このエラーは、Intelligent Agent が \$ORACLE_HOME/network/admin ディレクトリへの書き込み許可を持たない場合に発生することがあります。\$ORACLE_HOME/network/log/nmiconf.log を参照して、エラーを調べます。Intelligent Agent の起動の問題に関する詳細は、B-12 ページの「[Intelligent Agent の正常起動の確認](#)」を参照してください。

検出されないサービスがある services.ora ファイルを参照して、どのサービスが検出済であるかを調べます。

Intelligent Agent がマシン上で検出するすべてのデータベース・サービスは、関連する SQL*Net/Oracle Net 構成ファイルで定義されている必要があります。定義されていないサービスがあると検出は失敗し、最悪の場合、Intelligent Agent は停止するかエラーを返します。

Windows NT: リリース 8.0.4 以上の Intelligent Agent では、OracleService または OracleService<SID> で始まるサービス名が検索されます。OracleService で始まるすべてのエントリは、このマシン上で動作しているデータベースとみなされます。Intelligent Agent が取り扱うすべての SID は、関連する SQL*Net/Oracle Net ファイルで定義されている必要があります。

UNIX: oratab ファイルは、どの SID が存在するかを特定するために使用されます。リリース 7.3.3 以下の Intelligent Agent では、不正確な（Developer/2000 環境のものなど）SID が見つかると、検出は失敗します。この問題に対処するために、環境変数 \$ORATAB を使用して、Intelligent Agent に検出させるデータベースのみを含む別の oratab ファイルにアクセスできます。

残りのデータベースに対しては、oratab ファイル、および SQL*Net/Oracle Net のファイル群を調べることで、これらのファイルが存在し、すべての定義が記述されていることを確認します。listener.ora ファイルに、すべてのデータベースが記述されていることを確認します。詳細は、B-10 ページの「[Oracle Net 構成ファイルの確認](#)」、および B-11 ページの「[Oracle Net の動作確認](#)」を参照してください。

ジョブまたはイベント登録時の「サービス名が無効です」(エージェント固有のエラー)

または

ジョブまたはイベント登録時の「ファイル操作エラー」(オペレーティング・システム固有のエラー)

これらのエラーは通常、Enterprise Manager コンソール上のサービスと、Intelligent Agent によって検出されるサービスが同期していないときに発生します。たとえば、あるイベントを TESTDB に対して登録し、その後でデータベースの名前を PRODDB に変更したような場合、Intelligent Agent とコンソールは同期しなくなります。

この問題を修正するには、このサービスからジョブおよびイベントの登録をすべて削除し、次に、サービスが存在するノードをコンソールから削除します。自動検出ウィザードを使用し、コンソール上でノードを再検出します。

注意：リリース 7.3.2 では、別名の大文字と小文字が区別されます。

Windows NT で Intelligent Agent を使用している場合は、Windows NT での Intelligent Agent に関する説明のジョブまたはイベント登録時の 'サービス名が無効です' を参照してください。

「**Oralogin failed in orlon**」 oratcl 動詞 oralogon を使用した TCL スクリプトを実行している間に、このエラーが発生することがあります。メッセージ「**Oralogin failed in orlon**」は、接続文字列が間違っているか、使用されているアカウントがなんらかの理由でデータベースにログインできないことを表しています。このエラーをデバッグするには、Tcl トレースを有効にします。

ORA-01017 ユーザー名 / パスワードが無効です。ログオンは拒否されました。無効なユーザー名 / パスワード・エラーは、UNIX システムで X 端末から Intelligent Agent を起動する際に起こることがあります。この問題は、Data Gatherer (9i 以前) が、収集したデータをアップロードするために Capacity Planner リポジトリに接続できないために起こることがあります。このメッセージは 2、3 分おきに繰り返します。

Windows NT での Intelligent Agent

Intelligent Agent 起動時のすべての NT オペレーティング・システム・エラー

Windows NT-Agent のエラー・メッセージの相互参照については、B-28 ページの「[Oracle Intelligent Agent - Windows のイベント・ログ・メッセージ](#)」を参照してください。

オペレーティング・システム・エラーの発生後に Intelligent Agent の問題を解決する手順は、次のとおりです。

- 「イベント ビューア」を起動します。(このツールは「管理ツール」プログラム・グループにあります。) メイン・メニューの「ログ」をクリックし、次に「アプリケーション」を選択します。Intelligent Agent エラーのソースは、サービス名 `dbsnmp` の下にあります。リスト中の最も新しい `dbsnmp` エントリを選択します。イベントをダブルクリックすると、実際の原因が表示されます。
- `DBSNMP.LOG` および `NMICONF.LOG` の各ファイルには、発生した特定のエラーについてのより詳細な情報が記されています。
- `snmp_ro.ora` および `snmp_rw.ora` の各ファイルが
`%ORACLE_HOME%\network\admin` ディレクトリに存在し、サイズが 0 でないことを確認します。
- Intelligent Agent を起動したユーザーが、キュー・ファイルに対する読み込みおよび書き込み許可を持っていることを確認します。
- マシン名と TCP ホスト名が一致していることを確認します。
- 「DNS 名およびコンピュータ名の一致の確認」を参照してください。Oracle Net が実行中であることを確認します。
- TCP/IP が正しく構成され、動作していることを確認します。(「TCP/IP の構成と動作の確認」を参照してください。)
- `%ORACLE_HOME%\network\agent\mibs` ディレクトリから、必須でないファイルをすべて削除します。
- 最新の Intelligent Agent にアップグレードします。

「Agent への接続に失敗しました」エラー (送られた状態のままのジョブが存在する)

Windows NT 上では、実際には 2 種類のホスト名定義が存在します。1 つは NetBIOS の定義で、これは常にインストールされている、Windows NT 内部の Named Pipes プロトコル用に使用されます。もう 1 つは TCP/IP ホスト名で、Windows NT 上に TCP/IP をインストールしたときのみ構成可能です。

Windows NT の NetBIOS ホスト名を調べる手順は、次のとおりです。

- 「コントロール パネル」で「ネットワーク」をダブルクリックします。
- 表示されたダイアログ・ボックスの「コンピュータ名」が、NetBIOS のホスト名を表しています。

TCP/IP のホスト名を調べる手順は、次のとおりです。

- 「コントロール パネル」から「ネットワーク」>「プロトコル」>「TCP/IP プロトコル」>「プロパティ」>「DNS」の順に開きます。
- 表示された「ホスト名」が、TCP/IP のホスト名に相当します。

Windows NT Server 上では、前述の 2 種類の名前が異なるような構成がなされている場合でも、その両方に ping を実行できます。一方、クライアント側では、TCP/IP の実ホスト名のみに ping を実行できます。Intelligent Agent がローカル IPC 接続を使用している場合、Intelligent Agent は Named Pipes を使用し、さらに NetBIOS 名を使用します。すべての外部接続は TCP/IP 名を使用します。

この 2 種類の名前が一致しない場合、「Agent に接続できません」エラーが発生するか、ジョブがコンソール内に保留され続けます。したがって、NetBIOS のホスト名と TCP/IP のホスト名が一致していることを確認する必要があります。

ジョブ実行中の、失敗 -> 「output from job lost」エラーの発生 Intelligent Agent 用に作成した Windows NT ユーザー（『Oracle Enterprise Manager 構成ガイド』を参照）には、`%ORACLE_HOME%\network\agent` ディレクトリ（さらに、一部のアプリケーションでは TEMP ディレクトリ）に対する読み込み / 書き込み許可と、SYSTEM32 ディレクトリの読み込み許可を与える必要があります。

Windows NT ユーザーがこれらの許可を持っていることを確認してください。

UNIX での Intelligent Agent

サービスがまったく見つからずに検出が失敗する

まず、Oracle Net のファイル群がすべて存在し、正しく定義されていることを確認してください。次に、`oratab` ファイルを編集し、動作しているリスナーの有効な SID のみを記述することで、検出の問題を解決できます。この作業が終わったら、`oratab` ファイルに残りのエントリを追加して、問題の原因となるエントリを特定できます。

`$ORACLE_HOME/network/log/nmiconf.log` ファイルを参照して、エラーを確認してください。

NMS-0308 : 「アドレスでのリスニングに失敗しました : 別の Intelligent Agent が実行されている可能性があります」

このエラーの原因は 2 種類考えられます。

1. 1 台のマシン上の異なる `ORACLE_HOME` ディレクトリに 2 つの Intelligent Agent がインストールされている場合、Intelligent Agent の 2 つ目のインスタンスを起動しようとすると、このメッセージが表示されます。これは、両方の Intelligent Agent が同じ 1748 番のデフォルト・ポートをリスニングしようとするのが原因です。
- 1 台のマシン上には 1 つの Intelligent Agent のみを配置してください。
2. Intelligent Agent がリスニングするポート 1748 が他のプロセスによって使用されているか、以前にそのポートを使用して現在停止しているプロセスがポートを解放していません（SUN のマシン上で一般的な問題です）。

ポートが他のプロセスによって使用されていることを確認する手順は、次のとおりです。

1. 次の UNIX コマンドを入力します。

```
netstat -a | grep 1748
```

「LISTENING」で終わる文字列が画面に表示される場合、ポートは使用中です。

2. 次の条件に当てはまる場合、
 - netstat -a | grep 1748 （結果が「LISTENING」と表示されます。）
 - agentctl status agent （「データベース・サブエージェントは起動されていません。」と表示されます。）

次の操作を行います。

- ps -ef | grep dbsnmp
- kill -9 _____ （プロセス番号を入力）
- agentctl start agent で Intelligent Agent を再起動します

この時点でなお Intelligent Agent を起動できない場合、前述の手順を再度実行します。ただし、Intelligent Agent を再起動する前に次の操作を行います。

- cd \$ORACLE_HOME/network/agent
- *.q、services.ora、snmp_ro.ora および snmp_rw.ora を削除します。
- agentctl start agent で Intelligent Agent を再起動します

この操作により Intelligent Agent は再起動しますが、Intelligent Agent が過去に使用していたジョブおよびイベントのキューはすべて削除されます。

他の手段がすべて失敗する場合、マシンを再起動することでポートが解放されます。

Intelligent Agent 起動時の NMS-001 エラー（SNMP 環境のみ） このメッセージは、SNMP Master Agent（SNMP プロトコルを制御する UNIX 上のプロセス）と通信できないことを表します。デフォルトでは、Intelligent Agent は Oracle Net を介してリスニングし動作しますが、UNIX システム上の SNMP を経由して動作することもできます。

Master Agent を使用した通信を試みる場合以外は、このメッセージを無視しても差し支えありません。

NMS-00207 Agent xxxx user account is locked for database yyyy データベース・アカウントがロックされているため、データベースの 9i バージョンの監視用に Intelligent Agent で登録されているイベントは動作しません。

この場合は、Enterprise Manager データベースの起動・停止イベントにより、データベースが停止していることが常に表示されます。Intelligent Agent の dbsnmp.log ログファイルには

NMS-00207 エラー・メッセージが記録され、データベースの `dbsnmp` ユーザー・アカウントがロックされていることが示されます。

この問題を解消するには、データベースにログインして、次の操作を行う必要があります。

1. SQL 文を実行して、`dbsnmp` アカウントのロックを解除します。

```
ALTER USER dbsnmp ACCOUNT UNLOCK;
```

2. SQL 文を実行して、`dbsnmp` アカウントのパスワードをリセットします。

```
ALTER USER dbsnmp IDENTIFIED BY <password>;
```

3. リセットしたパスワードを、Intelligent Agent の `snmp_rw.ora` 構成ファイルに次のように追加します。

```
SNMP.CONNECT.<service_name>.PASSWORD=<password>
```

ここでの `service_name` は、Intelligent Agent によって `snmp_ro.ora/snmp_rw.ora` で検出されたシード・データベースの名前です。

4. `agentctl` を使用して Intelligent Agent を停止し、再起動します。

SYS アカウントで、データベースに対して `catsnmp.sql` スクリプトを実行します。

NMS-205 Failure to connect to database name with username/password string 'dbsnmp' ユーザーを見つけられませんでした。

SYS アカウントで、データベースに対して `catsnmp.sql` スクリプトを実行します。

NMS-351 Encryption key supplied is not the one used to encrypt the file この問題は、`$ORACLE_HOME/network/agent` ディレクトリにある `*.q` ファイルの ID 間に不一致がある場合に発生します。この状態は次の原因によって起こる場合があります。

- マシンのホスト名が変更された。
- 暗号化キーがパスワードの交換によって無効化され、Intelligent Agent が暗号化キーなしに、または他のキーで起動された。

`$ORACLE_HOME/network/agent` ディレクトリ中のすべての `*.q` ファイルを削除します。リポジトリを再構築します。Intelligent Agent を再起動します。

リリース 9.x の Intelligent Agent のトレース

Intelligent Agent のトレースおよびログでは、Intelligent Agent と Management Server 間のすべて通信、および Intelligent Agent の起動と検出情報を追跡できます。9i Intelligent Agent の追跡を有効にするには、Agent の `snmp_rw.ora` ファイルを変更する必要があります。ファイルは通常 `$ORACLE_HOME/network/admin` ディレクトリに存在します。`snmp_rw.ora` は Intelligent Agent のプロセスの最初の起動時に作成されます。ファイルが作

成されず、起動プロセスをトレースする必要がある場合、手動でテキスト・ファイルを作成し、必要なトレース・パラメータをファイルに追加します。

トレースが有効でない場合でも、ログ・ファイル

\$ORACLE_HOME/network/log/dbsnmp.log は、起動のたびに Intelligent Agent によって書き込まれます。このファイルには、Intelligent Agent の名前とバージョン、および Intelligent Agent 構成ファイルの名前とその場所が記されます。トレースが有効にされている場合、データベースおよびリスナーの接続に伴う問題も記述されます。

ログ・ファイル \$ORACLE_HOME/network/log/nmiconf.log は、Intelligent Agent の初回起動時に作成され、その後起動のたびに内容が追加されます。自動検出は、Tcl スクリプト nmiconf.tcl (ログ・ファイル名の由来) によって行われます。このファイルは起動の間にのみ書き込まれます。\$ORACLE_HOME/bin/oratclsh は、特別な目的に使用される Tcl シェルで、Intelligent Agent によってサポートされる OraTcl 動詞の (全部ではなく) 広範な部分集合に加えて、標準の Tcl 動詞 (TCL82 でサポートされる) のすべてをサポートします。ORATCLSH は汎用のユーティリティではなく、Intelligent Agent によってメンテナンスされるファイルおよびデータ構造に依存するため、Intelligent Agent との組合せでのみ使用できます。

ORATCLSH は Intelligent Agent のサポート対象機能セットの一部でなく、ORATCLSH のドキュメントも存在しません。ORATCLSH は、Enterprise Manager のジョブ・スクリプトおよびイベント・スクリプトを開発する Oracle ユーザーおよび開発者を支援する目的で限定的に提供されるデバッグ・ツールです。ORATCLSH の実行可能プログラムは、Tcl スクリプトのデバック用に提供されます。ORATCLSH を実行する前に、環境変数 TCL_LIBRARY が \$ORACLE_HOME/network/agent/tcl を指すように設定します。これは init.tcl ファイルのある場所です。

デフォルトでは、次のファイルが Intelligent Agent の \$ORACLE_HOME/network/log ディレクトリに作成されます。

- agntrvc.log (Windows NT)
- dbsnmp.log
- dbsnmp.nohup (監視スクリプト dbsnmpwd 用)
- dbsnmpj.log (ジョブのログ・ファイル)
- nmiconf.log (検出情報用)

snmp_rw.ora ファイルでさまざまなトレースおよびログ・パラメータを設定することにより、次の領域を監視できます。

- Intelligent Agent のプロセス (dbsnmp) および通信
- Intelligent Agent のジョブのサブシステム
- AGENTCTL ユーティリティ
- データ収集サービス
- Intelligent Agent のイベントのサブシステム

次の表では、各機能の領域に基づくパラメータのトレースおよびログを示します。

Intelligent Agent (dbsnmp) プロセスのトレースに使用されるパラメータ

表 B-2 Intelligent Agent (dbsnmp) プロセスのトレースに使用されるパラメータ

パラメータ	説明
dbsnmp.trace_level = <OFF USER ADMIN nn>	このパラメータは、トレースを有効にし、収集するデータの量（トレース・レベル）を指定するために使用される。トレース・レベル 16 では、最も詳細な情報を提供する。このパラメータは小文字である必要がある。トレース・レベルは SQL*Net および Oracle Net で使用されるレベルと同じ。
dbsnmp.trace_file= <filename>	デフォルトのトレース・ファイルは <code>dbsnmp.trc</code> 。異なるファイルにトレースを書き込む場合、 <code>dbsnmp.trace_file</code> パラメータを追加する。
dbsnmp.trace_directory=<directory>	デフォルトのトレース・ディレクトリは <code>\$ORACLE_HOME/network/trace</code> 。トレース・ファイルの場所を変更する場合、 <code>dbsnmp.trace_directory</code> パラメータを追加する。
dbsnmp.trace_filecnt=<integer>	このパラメータでは、生成されるトレース・ファイルの最大数を定義する。
dbsnmp.trace_filesize=<integer in kilobytes>	このパラメータでは、トレース・ファイルの最大サイズをキロバイトで定義する。
dbsnmp.trace_unique={true/false}	このパラメータが <code>true</code> に設定されている場合、常に固有のトレース・ファイルを生成する。
dbsnmp.trace_timestamp={true/false}	このパラメータでは、トレースの各行の前にタイムスタンプを配置するかどうかを定義する。
dbsnmp.log_directory=<directory>	デフォルトのログ・ディレクトリは <code>\$ORACLE_HOME/network/log</code> 。ログ・ファイルの場所を変更する場合、 <code>dbsnmp.log_directory</code> パラメータを追加する。ただし、Intelligent Agent は指定するディレクトリに対する書き込み権限を持っている必要がある。

表 B-2 Intelligent Agent (dbsnmp) プロセスのトレースに使用されるパラメータ (続き)

パラメータ	説明
dbsnmp.log_file=<filename>	デフォルトのログ・ファイルは <code>dbsnmp.log</code> 。名前を変更する場合、 <code>dbsnmp.log_file</code> パラメータを追加できる。トレースが有効でない場合でも、ログは起動のたびに Intelligent Agent によって書き込まれる。このファイルには、Intelligent Agent の名前とバージョン、および Intelligent Agent 構成ファイルの名前とその場所が記される。トレースが有効にされている場合、データベースおよびリスナーの接続に伴う問題も記述される。
dbsnmp.log_unique={true/false}	このパラメータを使用すると、Intelligent Agent が起動するたびに一意のログ・ファイルを作成する。

表 B-3 ジョブ・システムのトレースに使用されるパラメータ

パラメータ	説明
dbsnmpj.trace_level={OFF USER ADMIN nn}	このパラメータは、ジョブ・システムのトレースを有効にし、収集するデータの量 (トレース・レベル) を指定するために使用される。トレース・レベル 16 では、最も詳細な情報を提供する。このパラメータは小文字である必要がある。
dbsnmpj.trace_directory=<directory>	デフォルトのトレース・ディレクトリは <code>\$ORACLE_HOME/network/trace</code> 。トレース・ファイルの場所を変更する場合、 <code>dbsnmpj.trace_directory</code> パラメータを追加する。
dbsnmpj.trace_file=<filename>	デフォルトのトレース・ファイルは <code>dbsnmpj.trc</code> 。異なるファイルにトレースを書き込む場合、 <code>dbsnmpj.trace_file</code> パラメータを追加する。
dbsnmpj.trace_filecnt=<integer>	このパラメータでは、生成されるトレース・ファイルの最大数を定義する。
dbsnmpj.trace_filesize=<integer in kilobytes>	このパラメータでは、トレース・ファイルの最大サイズをキロバイトで定義する。

表 B-3 ジョブ・システムのトレースに使用されるパラメータ（続き）

パラメータ	説明
<code>dbsnmpj.log_directory=<directory></code>	デフォルトのログ・ディレクトリは <code>\$ORACLE_HOME/network/log</code> 。ログ・ファイルの場所を変更する場合、 <code>dbsnmpj.log_directory</code> パラメータを追加する。ただし、Intelligent Agent は指定するディレクトリに対する書込み権限を持っている必要がある。
<code>dbsnmpj.log_file=<filename></code>	デフォルトのログ・ファイルは <code>dbsnmpj.log</code> 。
<code>dbsnmpj.log_unique={true/false}</code>	このパラメータを使用すると、ジョブが起動するたびに Intelligent Agent が一意のログ・ファイルを作成する。

表 B-4 Intelligent Agent の起動のトレースに使用されるパラメータ

パラメータ	説明
<code>agentctl.trace_level={OFF USER ADMIN nn}</code>	このパラメータは、agentctl ユーティリティ（エージェントの起動）のトレースを有効にするために使用される。問題が発生する場合、Oracle Support では通常トレースのレベルを 16 に設定する必要がある。
<code>agentctl.trace_file=<filename></code>	デフォルトのトレース・ファイルは <code>agentctl.trc</code> 。異なるファイルにトレースを書き込む場合、 <code>agentctl.trace_file</code> パラメータを追加する。
<code>agentctl.trace_directory=<directory></code>	デフォルトのトレース・ディレクトリは <code>\$ORACLE_HOME/network/trace</code> 。トレース・ファイルの場所を変更する場合、 <code>agentctl.trace_directory</code> パラメータを追加する。
<code>agentctl.trace_filecnt=<integer></code>	このパラメータでは、生成されるトレース・ファイルの最大数を定義する。
<code>agentctl.trace_filesize=<integer in kilobytes></code>	このパラメータでは、トレース・ファイルの最大サイズをキロバイトで定義する。
<code>agentctl.trace_unique={true/false}</code>	このパラメータが <code>true</code> に設定されている場合、常に固有のトレース・ファイルを生成する。
<code>agentctl.trace_timestamp={true/false}</code>	このパラメータでは、トレースの各行の前にタイムスタンプを配置するかどうかを定義する。

データ収集サービスのトレース

データ収集サービス（以前の Data Gatherer）機能は 9i Intelligent Agent に統合されているため、データ収集ベースのトレースは、プラットフォームに応じて次のいずれかの手順で有効にできます。

UNIX の場合：

1. 環境変数 VP_DEBUG を 1 に設定します。

```
>setenv VP_DEBUG 1
```

2. Intelligent Agent を起動します。

```
> agentctl start agent
```

すべての収集アクティビティのログが \$ORACLE_HOME/network/log/dbsnmp.nohup に記録されます。

Windows NT（DOS ウィンドウから）の場合：

1. 環境変数 VP_DEBUG を 1 に設定します。

```
>set VP_DEBUG=1
```

2. Intelligent Agent を起動し、出力をテキスト・ファイルにリダイレクトします。

```
> dbsnmp -agent_name Oracleora920Agent > stdout.log2> odg.log
```

Windows 2000（DOS ウィンドウから）の場合：

1. 環境変数 VP_DEBUG を 1 に設定します。

```
>set VP_DEBUG=1
```

2. Intelligent Agent を起動し、出力をテキスト・ファイルにリダイレクトします。

```
> dbsnmp -agentname Oracleora920Agent > stdout.log2> odg.log
```

複数の ORACLE_HOME

同一のマシンに複数の ORACLE_HOME が存在する場合、次の操作を実行します。

1. Intelligent Agent の固有の ORACLE_HOME/bin ディレクトリに移動します。
2. 環境変数 VP_DEBUG を 1 に設定します。

```
>setenv VP_DEBUG 1
```

3. Intelligent Agent を起動します。

```
> agentctl start agent
```

イベント・システムのトレース (Tcl のトレース)

snmp_rw.ora ファイルの dbsnmp.trace_level パラメータをレベル 1 以上に設定して (dbsnmp.trace_level >= 1)、イベントのトレースを有効にすることもできます。これらのパラメータを有効にするには、Intelligent Agent をシャットダウンし、再起動する必要があります。Tcl のトレースにより、ORACLE_HOME/network/trace ディレクトリにファイル oratcl.trc が作成されます。イベントが発生するたびに、oratcl.trc ファイルにエントリが追加されます。

Windows NT/2000 での Intelligent Agent の起動時エラーの理解

Oracle Intelligent Agent サービスに失敗したり、または Oracle Intelligent Agent が、イベント・ログにエントリを書き込むために Windows ReportEvent 機能呼び出す規定のエラー・コードで起動に失敗した場合、Window NT はイベントログ・サービスにパラメータを渡します。これにより、かわりにイベント・ログにログの記録を書き込む情報を使用します。その他のエラーは nmi.log および nmiconf.log に記録されます。

Windows NT イベント・ビューア・アプリケーションの起動時に、OpenEventLog 機能を使用してイベント・ソースのイベント・ログをオープンします。イベント・ビューアは次に ReadEventLog 機能を使用してそのログからイベント・レコードを読み込みます。ReadEventLog は、EVENTLOGRECORD 構造、およびログ・イベントを記述した追加情報を含むバッファを返します。

Oracle Intelligent Agent - Windows のイベント・ログ・メッセージ

Intelligent Agent サービスが起動に失敗すると、Windows のサービス・マネージャは基になるエラー・コードを返しますが、Oracle のイベント・メッセージを解釈できないため、Windows NT Win32 のメッセージ・テキストを不正に返します。ただし、正しいメッセージは NT イベント・ログに記録されます。

次の表では、Intelligent Agent が Windows NT のイベント・ビューアで表示するイベントおよび関連する Win32 のエラー・テキストを定義しています。

表 B-5 MS Windows-Intelligent Agent のエラー・メッセージの変換

ID	Windows NT/2000 メッセージ	Intelligent Agent の説明
1	ファンクションが間違っています。	OracleAgent がサービス・コントロール・ハンドラの登録に失敗。
2	指定されたファイルが見つかりません。	OracleAgent がサービス・コントロール・マネージャへの状況の報告に失敗。
3	指定されたパスが見つかりません。	OracleAgent がスレッド同期オブジェクトの作成に失敗。
4	ファイルを開くことができません。	OracleAgent がスレッドの作成に失敗。

表 B-5 MS Windows-Intelligent Agent のエラー・メッセージの変換（続き）

ID	Windows NT/2000 メッセージ	Intelligent Agent の説明
5	アクセスが拒否されました。	OracleAgent がメモリの割り当てに失敗。
6	ハンドルが無効です。	OracleAgent が暗号化キーの取得に失敗。
7	記憶域制御ブロックが壊れています。	<p>Oracle Agent が自動検出スクリプト nmiconf.tcl の実行に失敗。詳細は nmiconf.log を参照。</p> <p>Intelligent Agent が自動検出プロセスの実行に失敗。これは、TCP/IP 層に問題がある無効な構成ファイルか、または TCL ライブラリのエラーによるものである。SQL*Net または Oracle Net 構成ファイルを検証し、SQL*Net または Oracle Net 経由でサーバーからデータベースに通常のループバック接続を行えるかどうかを確認する。</p> <p>問題が継続する場合、Intelligent Agent ソフトウェアを再インストールする。</p>
8	このコマンドを実行するのに十分な記憶域がありません。	OracleAgent が Oracle CORE ライブラリの初期化に失敗。
9	記憶域制御ブロックのアドレスが無効です。	<p>OracleAgent が Oracle NLS ライブラリの初期化に失敗。</p> <p>NLS 環境またはレジストリ変数（あるいはその両方）が不正である。NLS で始まる変数のシステムを確認すること。SQL*Plus または SQL*Worksheet などの標準ツールを使用してデータベースに接続し、NLS クライアントの設定を検証する。問題が継続する場合、必要なサポート・ファイル・ソフトウェアを再インストールして、このエラーを解決する。</p>
10	環境が間違っています。	<p>OracleAgent が Oracle SQL*Net または Oracle Net ライブラリ :%1 の初期化に失敗。</p> <p>SQL*Net か Oracle Net 環境またはレジストリ変数（あるいはその両方）が不正である。SQL*Plus または SQL*Plus Worksheet などの標準ツールを使用して、SQL*Net または Oracle Net の接続を試行する。問題が継続する場合、SQL*Net または Oracle Net クライアント（リスナーが動作している場合はサーバーも）ソフトウェアを再インストールしてこのエラーを解消する。</p>

表 B-5 MS Windows-Intelligent Agent のエラー・メッセージの変換（続き）

ID	Windows NT/2000 メッセージ	Intelligent Agent の説明
11	間違ったフォーマットのプログラムを読み込もうとしました。	OracleAgent が DES の暗号化の初期化に失敗。
12	アクセス コードが無効です。	OracleAgent が Oracle Remote Operations Library の初期化に失敗。
14	この操作を完了するのに十分な記憶域がありません。	OracleAgent が dbsnmp.ver ファイルの作成に失敗。
15	指定されたドライブが見つかりません。	OracleAgent がキュー・ファイルの作成 / 読込みに失敗。 マシンのホスト名はエージェントの .q ファイルで暗号化されている。エージェントによってファイル内で検出されたこのホスト名が、エージェントが動作しているマシンの名前に適合しない場合、このエラーが生成される。エージェントの .q ファイルを削除し、エージェントを再起動する。
16	ディレクトリを削除できません。	OracleAgent がジョブ・スケジュール・シンボル表の作成に失敗。
17	ファイルを別のディスク ドライブに移動できません。	OracleAgent が接続キャッシュの初期化に失敗。 マシンの TCP ホスト名と NetBios ホスト名が一致していない。「コントロール・パネル」の「ネットワーク」で、プロトコル・プロパティ・ダイアログのマシン名を同期します。
18	これ以上ファイルがありません。	OracleAgent が SNMP へのサイン・オンに失敗。
19	このメディアは書き込み禁止になっています。	OracleAgent がパラメータ・ファイルから SNMP 索引の読込みに失敗。
20	指定されたドライブが見つかりません。	OracleAgent がデータベースへの接続に失敗。 SQL*Net または Oracle Net 構成ファイルからの情報を使用したことにより、エージェントは目的のデータベースへの接続に失敗した。これは、TCP/IP の競合、または設定ファイルにおける不正な TCP ポートの参照などのプロトコル・エラーによるものである可能性が高い。特定の接続エラーのエージェント・ログ・ファイルを確認する。

表 B-5 MS Windows-Intelligent Agent のエラー・メッセージの変換（続き）

ID	Windows NT/2000 メッセージ	Intelligent Agent の説明
21	ドライブの準備ができていません。	OracleAgent が SNMP キャッシュの構築に失敗。
22	デバイスがコマンドを認識できません。	OracleAgent が MIB の構築に失敗。 MIB の仕様に準じない %ORACLE_HOME%\network\agent\mibs ディレクトリ内のファイルが存在する。 ディレクトリ内のすべてのファイルを確認 すること。非 MIB ファイルまたは破損した MIB ファイルをすべて削除すること。
23	データエラー（巡回冗長検査（CRC）エ ラー）です。	OracleAgent が MIB 行の登録に失敗。
24	プログラムはコマンドを発行しましたが、 コマンドの長さが間違っています	OracleAgent が通信スレッドの再起動に失 敗。
25	プログラムはコマンドを発生しましたが、 コマンドの長さが間違っています。	OracleAgent が指定のポートのリスニング に失敗。他の OracleAgent がすでに起動し ている可能性がある。 .q ファイルで使用されている暗号化キーの 使用における内部エラーが存在する。これ は、マシンで TCP/IP が競合しているか、 またはマシンのシステム・リソースの不足 によって Intelligent Agent ルーチンの適正 な初期化が妨げられている可能性がある。 システム・リソースを確認し、マシンの TCP/IP 設定が正しいことを検証する。
次のエラー・メッセージは、サービスを管理する Windows NT プログラムがサービスを制御し なくなったり、サービスへの制御を失った場合に生成されます。		
1067	プロセスを途中で強制終了しました。	DLL の問題または、システム設定の問題な どが考えられる。 正確な状況を確認するには、Dr. Watson の ログ、イベント・ビューアのエラー、およ びログ・ファイルのメッセージを確認す ること。
2140	内部エラーが発生しました。	サービス・プログラムがサービスから矛盾 する情報を受信。これは、検出されたシス テム情報の矛盾による Intelligent Agent の 異常終了が原因である可能性が高い。 Intelligent Agent の強制終了についての詳細 は、検出ログ・ファイルを参照。

表 B-5 MS Windows-Intelligent Agent のエラー・メッセージの変換（続き）

ID	Windows NT/2000 メッセージ	Intelligent Agent の説明
2186	サービスが制御機能に応答しません。	<p>Intelligent Agent は、サービスを監視する SERVICES プログラムにステータスを報告することに失敗。</p> <p>Agent サービスを停止できない場合、コマンド・ラインから KILL コマンドを使用してエージェント・プロセスを停止する。KILL コマンドはオプションの MS リソース・キットの一部であるため、標準の NT ツールには含まれない。</p>

UNIX での一般的な Intelligent Agent 設定の問題

Intelligent Agent ソフトウェアは RDBMS サーバー・ソフトウェアとともに提供されます。ただし、これは Intelligent Agent ソフトウェアをデータベースとともにインストールする必要があるということではありません。他の Oracle ソフトウェアとは異なる専用の \$ORACLE_HOME に、Intelligent Agent を単独でインストールすることもできます。

Intelligent Agent が監視するデータベースへ接続し、Enterprise Manager フレームワークの他の部分と通信できるようにするには、Intelligent Agent に SQL*Net または Oracle Net のいずれか 1 つが必要です。SQL*Net または Oracle Net 製品、および基本となる共通ライブラリは、Intelligent Agent が UNIX にインストールされる際に常に自動的にインストールされます。

Intelligent Agent が起動しているのと同じ \$ORACLE_HOME で SQL*Net リスナーを起動させる必要はありません。また、Intelligent Agent と同じベース・バージョンである必要もありません。

Intelligent Agent のリンク

重要： ソフトウェアにリンクする前に Intelligent Agent を停止する必要があります。

インストール・メディアからソフトウェアがインストールされると、Intelligent Agent はシステムにある現在のシステム・ライブラリを使用して直ちに再リンクされます。

UNIX カーネルまたはシステム・ライブラリに変更が加えられている場合、次のコマンドを使用して Intelligent Agent を再リンクすることをお勧めします。

```
$ cd $ORACLE_HOME/network/lib
$ make -f ins_oemagent.mk install
```

これにより、2つの新しい実行可能ファイルが作成されます。`db snmp` および `oratclsh` は、両方とも `$ORACLE_HOME/bin` ディレクトリに作成されます。これらの実行可能ファイルがすでに存在している場合、古いファイルは `db snmp0` および `oratclsh0` という名前に変更されます。新しいソフトウェアがテストされると、これらの安全のためのコピーは直ちに削除されます。

データベース・ソフトウェアを再リンクする際、またはインストールされた `SQL*Net` プロトコル・ドライバを変更した後、そのバージョンのデータベースをすべてシャットダウンする必要があります。

ただし、エージェントが再リンクされると、`root.sh` ファイルを直ちに再実行するか、または `db snmp` 実行ファイルを調整するための手動による介入を直ちに実行する必要があります。

`root.sh` ファイルが調整されたり、新しいファイルによって上書きされる場合、次のコマンドを `root` ユーザーとして手動で実行する必要があります。

```
$ cd $ORACLE_HOME/bin
$ chmod 6751 db snmp
$ chown root db snmp
```

これらの手順は、エージェントを適切に動作させるために必要です。エージェントが `setuid` の許可 (`chmod '6751'` コマンドによって与えられる) を持たない場合、または `root` (`chown root` コマンド) によって所有されていない場合、検出に失敗することがあり、これによりジョブは適切に実行されません。

また、マシン上でエージェントがすでに実行されていると、いくつかのファイルが `root` で所有されているため、エージェントは起動に失敗します。さらに、再リンクして所有権が `Oracle` の所有者に戻った後に、正しくないファイルを更新します。

Intelligent Agent の実行

プログラムを `root` によって UNIX 上で実行する場合、安全上の理由によって環境変数 `LD_LIBRARY_PATH` は読み込まれません。これは、`.EXE` ファイルに動的にリンクしているすべての共有ライブラリは、ディスク上の絶対パスを使用して参照される必要があることを意味します。

`LDD` コマンドを使用して、リンクされた共有ライブラリを確認できます。次に例を示します。

```
$ ldd db snmp
```

共有ライブラリの問題を回避するには、3つのオプションがあります。

- ソフトウェアを静的にリンクします。
ライブラリが静的にリンクされると、`ld` はそれらが共有であると認識しないため、問題が解決されます。ここでは、実行ファイルのサイズが非常に大きくなります。
- `/usr/lib` ディレクトリの共有ライブラリをコピーまたはリンクします。

簡単で直接的な方法としては、`/usr/lib` の必要な共有ライブラリにシンボリック・リンクを作成する方法があります。このディレクトリはライブラリ・ローダー内でハードコーディングされているため、このディレクトリのすべての共有オブジェクトは常に検出されます。

この場合、同じマシンに複数のバージョンのソフトウェアがインストールされていると、潜在的な問題が発生します。`/usr/lib` ディレクトリはマシン全体で同じであるため、このディレクトリのファイルをコピーしている場合、またはこのディレクトリのファイルにリンクしている場合、ソフトウェアの他のバージョンが正しくない共有オブジェクトを使用することになります。

- 絶対パスによって共有オブジェクトを参照します。

これは最良の方法ですが、**Make** ファイルの修正および使用ライブラリへのフル・ハードコーディング・パスの追加を伴うため、最も難しい方法でもあります。変更後、ソフトウェアを再リンクし、ライブラリが正しく検出されることを確認します。

特に **Intelligent Agent** では、共有ライブラリに問題があると次のエラーが発生する場合があります。

```
ld: Can not map libclntsh.so.1.0
ld: Error opening libclntsh.so.1.0
ld: Can not find libnet.so
```

エージェントの起動時にエラーが発生した場合、次の操作を実行します。

1. エージェントを再リンクします。

```
$ make -f ins_oemagent.mk install
```

2. 実行可能ファイルの所有者および許可を変更します。

```
$ chmod 6751 dbsonmp
$ chown root dbsonmp
```

Data Gatherer の理解およびトラブルシューティング

重要： この項は **9i 以前の Intelligent Agents** にのみ適用されます。Data Gatherer によって提供されるデータ収集サービスは、現在は **9.x Intelligent Agent** に統合されています。

Oracle Data Gatherer はデーモン・プロセスで、Oracle Performance Manager および Oracle Capacity Planner などの Enterprise Manager ツールで使用されるホスト・オペレーティング・システムや、Oracle データベースからのパフォーマンス統計の収集を管理します。前述のように、この機能は現在は **9.x Agent** の統合された一部になっています。この項は、古いバージョンの Agent または Data Gatherer にのみ適用されます。

Data Gatherer の構成

Data Gatherer は次の種類のデータを収集します。

- オペレーティング・システム
- データベース
- Oracle Applications のコンカレント・マネージャ
- Oracle HTTP Server
- Microsoft SQL Server

使用しているオペレーティング・システムで Data Gatherer が使用できない場合、オペレーティング・システムのデータを収集できないことがあります。オペレーティング・システムのデータを収集するには、オペレーティング・システムと同じホストに Data Gatherer がインストールされている必要があります。次の両方の要件が満たされていないと、特定のホストでの Oracle Data Gatherer のインストールおよび構成を実行できないことがあります。

- ホスト・オペレーティング・システムは、Oracle Data Gatherer のインストールおよび構成をサポートしている必要があります。ベース開発では、Data Gatherer は NT および Solaris 用です。
- ホスト上に Oracle 8.0.4 以降の Oracle ホームが存在する必要があります。

サーバー・ホストにインストールされている Data Gatherer の使用

これは、Oracle Data Gatherer を使用する際の基本構成です。Data Gatherer はデータベース・サーバー・インストールの一部として別個にインストールされます。8.0.5 以降では、Data Gatherer は Intelligent Agent とともにインストールされますが、Intelligent Agent の 9.x バージョンのように必須な要素としてではありません。Data Gatherer は、ホストのすべてのデータベースに対するデータベース統計の監視に使用したり、ホスト自体のオペレーティング・システム統計の監視に使用することができます。

クライアント・ホストにインストールされている Data Gatherer の使用

クライアントが動作するのと同じホストに Data Gatherer をインストールすることができます（ただし、クライアントが Windows NT 4.0 の場合のみ。Windows 95 と 98 ではサポートされていません）。この構成では、リモート・データベースのデータベース統計を監視できますが、リモート・ホストのオペレーティング・システム統計を監視することはできません。

代替ポートでの Data Gatherer の使用

Data Gatherer およびクライアントは、標準の（IANA 登録）ポート（1808）においてインストールおよび実行されることを前提としています。このポートはクライアントと Data Gatherer サーバー間の通信に使用されます。他のポートを使用することもできますが、その場合この構成はサポートされません。

複数の Oracle ホームの使用

複数の Oracle ホームを持つ環境に Oracle Data Gatherer をインストールできますが、これを行う場合は次の 2 つの点について注意する必要があります。

- Data Gatherer は標準のポートを使用するため、2 つのホームで Data Gatherer を同時に使用する場合は、1 つを（サポートされていない）代替ポートで実行する必要があります。
- Data Gatherer に送信されるデータベース接続文字列は、Oracle ホームの SQL*Net 環境から解釈可能である必要があります。

Data Gatherer のリカバリ能力

ホストの再起動時の Data Gatherer の再起動

Data Gatherer はすべての現在の履歴収集の状態を保存するために設定されています。これにより、Data Gatherer の再起動時に、状態ファイルからこれらの収集を再起動するためのリカバリ・スレッドを作成します。Data Gatherer がシステムの再起動時に自動的に起動するように設定されている場合、収集は継続します。

サイクル・データベースからの履歴データの収集

データが収集されるデータベースがサイクルしている場合（例、バックアップ用に毎晩停止される）、Data Gatherer は再起動時にデータベースからのデータの収集を継続するように設計されています。Data Gatherer は、指定された収集間隔でデータベースに再接続を試行し、接続を確立するまで続行します。

その他の Data Gatherer の問題

- Data Gatherer のアラート・ログ（\$ORACLE_HOME/odg/log/alert_dg.log）には、Data Gatherer が保存対象とする多くのメッセージが含まれます。現在これには、Data Gatherer によって報告されたエラーおよび情報メッセージが含まれます。アラート・ログに含まれるすべてのメッセージには、タイムスタンプが含まれます。
- Data Gatherer の起動
UNIX: vppcntl -start（UNIX コマンド）
NT: 「コントロール パネル」の「サービス」から、OracleDataGatherer サービスを起動します。
- Data Gatherer の停止
UNIX: vppcntl -stop（UNIX コマンド）
NT: 「コントロール パネル」の「サービス」から、OracleDataGatherer サービスを停止します。

- Data Gatherer のステータスの確認

```
vppcntl - status
```

このステータス確認では、次の 2 つのメッセージのうちのいずれかが表示されます。

The Oracle Data Gatherer is running!

または

The Oracle Data Gatherer is not running

Intelligent Agent の初期化

Intelligent Agent の初期化には、すべての既存のジョブおよびイベント定義の削除を伴います。これは、Enterprise Manager 環境の再初期化が必要な場合、またはカスタム・サポートの担当者に指示された場合にのみ行う必要があります。この操作は、コンソールおよび Intelligent Agent ノードの両方で行う必要があります。

Intelligent Agent を初期化するには、次の操作を行います。

1. Enterprise Manager コンソールから

- エージェントが動作するマシンに登録されているすべてのジョブおよびイベントを削除します。
- すべてのイベントおよびジョブを削除して、コンソールを停止します。

2. ターゲットで現在動作している Intelligent Agent を停止します。

UNIX では、次のコマンドを実行します。

```
$ agentctl stop agent
```

stop コマンドを実行した後、ps コマンドを使用して db snmp が停止していることを確認します。Intelligent Agent を停止できない場合、ps コマンドを使用して db snmp プロセスのプロセス ID を取得し、kill -9 コマンドを使用してこれらのプロセスを停止します。

Windows NT では、「コントロール パネル」>「サービス」を使用して Agent サービスをオフにするか、またはコマンド・ライン・オプションを発行します。

```
C:\> agentctl stop agent
```

エージェントを停止したら、TaskManager を使用して db snmp プロセスが停止していることを確認します。

3. \$ORACLE_HOME/network/agent ディレクトリに移動して、次のファイルを削除します。

- SERVICES.ORA: エージェントが検出したすべてのサービスを含むファイル。このファイルは、次のエージェントの起動時に再作成されます。
- .q 拡張子を持つすべてのファイル: これらのファイルは、登録され、動作しているすべてのジョブ / イベントをバイナリで表したものです。エージェントは、q ファイルが存在しない場合、起動時に新しい q ファイルを作成します。

エージェントが動作するホストについての情報は、これらのファイルにも保存されます。エージェント・マシンの名前や IP アドレスが変更された場合、ファイルを再作成する必要があります。

- .inp 拡張子を持つすべてのファイル: これらは登録されたジョブのパラメータ・ファイルです。
- .jou 拡張子を持つすべてのファイル: これらは、エージェントがジョブの実行に使用するための情報を持つジャーナル・ファイルです。
- out 接頭辞を持つすべてのファイル: これらは、ジョブの実行時に生成される一時出力ファイルです。これらのファイルは、ジョブの終了後にディレクトリ内に残ることはありません。
- tcl 接頭辞を持つすべてのファイル: これらは、エージェントに送られるジョブで指定されている特定のコマンドを含む、コマンド Tcl テンプレート・スクリプトです。
- DBSNMP.VER: これはエージェントのバージョン情報を含むテキスト・ファイルです。このファイルは、エージェントが再度起動した際には再作成されます。

4. \$ORACLE_HOME/network/admin ディレクトリに移動します。このディレクトリでは、次のファイルを削除します。

- SNMP_RO.ORA: エージェントによって検出された、マシン上のサービスの読み込み専用情報です。

ユーザーはこのファイルを絶対に編集してはなりません。起動時にこのファイルが存在すると、このファイルは読み込まれ、さらに新しい検出情報で再作成されます。

- SNMP_RW.ORA: エージェントによって検出された、マシン上のサービスの読み書き両用情報です。

管理者はこのファイル内の一部の情報を編集して、検出したサービスについての詳細情報を提供できます。

Intelligent Agent の起動時にこのファイルが存在すると、このファイルは読み込まれ、さらに検出後に再書き込まれます。情報はこのファイルにのみ追加されます。

5. エージェントが作成した既存のログ・ファイルを名称変更または削除します。これにより、Intelligent Agent の初期化以降のすべてのログ・メッセージを追跡できます。

次のファイルは、\$ORACLE_HOME/network/log ディレクトリ内に存在します。エージェントの起動時に使用されます。

- NMICONF.LOG: 検出ログ・ファイル
- NMI.LOG: SQL*Net 通信エラー (Windows NT のみ)
- DBSNMP.LOG: 通信および動作スレッドのログ (ログが有効な場合)

\$ORACLE_HOME/network/agent/library ディレクトリからすべてのファイルを削除する必要があります。

6. 空の SNMP_RO.ORA ファイルを作成し、次の行を追加します。

```
snmp.visible_services = ()
```

7. 特定の状況のデバッグのためにトレースが必要な場合、デバッグ・パラメータを持つ新しい SNMP_RW.ORA を作成できます。

```
dbsnmp.trace_level=16
dbsnmp.trace_unique=true
```

8. Intelligent Agent を再起動します。

エージェントの起動後、まず初めに SERVICES.ORA ファイルを検証します。このファイルがマシンのすべてのサービスを含んでいる場合、SNMP_RO.ORA および SNMP_RW.ORA ファイルをチェックします。NMICONF.LOG ファイルで検出された問題が発見されます。

重要： q ファイルを削除すると、Agent からすべての状態情報も削除されます。すなわち、Intelligent Agent には登録されたジョブおよびイベントの情報がなくなります。

エージェントへのジョブおよびイベントの送信時に q ファイルを削除すると、フレームワークおよびエージェント間の同期エラーが発生します。そのため、Intelligent Agent の再初期化が必要であるとカスタマ・サポートの担当者から特別に指示された場合にのみ、この方法を使用します。

イベント / ジョブは順序番号によって内部的に識別されます。q ファイルが作成されると、順序はリセットされます。イベント送信時において Management Server は、戻される内部 ID がノードに対して以前に登録された他の内部 ID よりも小さいと認識し、システム間に歪みがあることを示すメッセージを発行します。唯一のオプションは、すべてのイベントとジョブを削除して、それらを再登録することです。Management Server が ID の不一致を検出すると、Intelligent Agent は corrupted とマークされ、エージェントに送信するすべてのジョブまたはイベントはエラーになります。

```
VNI-4040: The agent on node is not in sync with the Management Server
```

9. \$ORACLE_HOME/network/agent/reco ディレクトリの内容を削除します。

UNIX での Intelligent Agent の検出エラーの診断

UNIX での検出プロセスは次の操作を伴います。

- 主ホスト名およびその別名の取得。
- ORATAB ファイルの読み込み。データベースのリストおよび対応する ORACLE_HOME はこのファイルから取得されます。
- ORATAB ファイルから読み込まれる各 ORACLE_HOME の SQL*Net LISTENER.ORA および SQLNET.ORA ファイルの読み込み。データベース・サービスの名前は LISTENER.ORA ファイルにより決定されます。これらの名前は、Enterprise Manager コンソールで表示可能な名前であり、データベースに接続するための TNS の別名として使用されます。
- データベースの GLOBAL_DBNAME パラメータが LISTENER.ORA ファイルで指定されていない場合、利用可能な TNS の別名を検索するために TNSNAMES.ORA ファイル内を検索します。
- TNS の別名が見つからない場合、または重複した TNS の別名の文字列が存在する場合、Intelligent Agent はデータベースの TNS の別名を生成し、この生成された名前をサービス名として使用します。

UNIX で検出時に使用されるファイル

listener.ora : 受信 SQL*Net 接続の定義

\$ORACLE_HOME ごとに 1 ファイル

次の場所のうちの 1 つに存在します（検索はこの順序で行われます）。

- \$TNS_ADMIN
- /etc または /var/opt/oracle
- \$ORACLE_HOME/network/admin

nmiconf.log : Intelligent Agent の検出警告 / エラー

Intelligent Agent ごとに 1 つ

\$ORACLE_HOME/network/log 内

nmiconf.lst : サード・パーティ製の追加の検出スクリプトのリスト

Intelligent Agent ごとに 1 つ

\$ORACLE_HOME/network/agent/config 内

nmiconf.tcl : Intelligent Agent の検出スクリプト

Intelligent Agent ごとに 1 つ

\$ORACLE_HOME/network/agent/config 内

oratab : マシン上のすべてのデータベースのファイル

マシンごとに 1 つ

/etc または /var/opt/oracle 内

services.ora : エージェントが検出したすべてのサービス定義のファイル

Intelligent Agent ごとに 1 つ

\$ORACLE_HOME/network/agent 内

snmp_ro.ora : すべての読み専用サービス情報のファイル

Intelligent Agent ごとに 1 つ

\$TNS_ADMIN または \$ORACLE_HOME/network/admin 内

snmp_rw.ora : すべての更新可能なサービス情報のファイル

Intelligent Agent ごとに 1 つ

\$TNS_ADMIN または \$ORACLE_HOME/network/admin 内

sqlnet.ora :SQL*Net 固有のパラメータのファイル

\$ORACLE_HOME ごとに 1 ファイル

次のいずれかに存在します（検索はこの順序で行われます）。

- \$TNS_ADMIN
- /etc または /var/opt/oracle
- \$ORACLE_HOME/network/admin

tnsnames.ora : データベースに接続するための TNS の別名のファイル

\$ORACLE_HOME ごとに 1 ファイル

次のいずれかに存在します（検索はこの順序で行われます）。

- \$TNS_ADMIN
- /etc または /var/opt/oracle
- \$ORACLE_HOME/network/admin

UNIX での検出問題の診断

フェーズ 1: ORATAB ファイルの確認

ORATAB ファイルは /etc または /var/opt/oracle ディレクトリのいずれかに存在します。

いずれのディレクトリに構成ファイルが存在するかについては、使用しているオペレーティング・システム固有のドキュメントを参照してください。

確認事項：

- ダミー・データベースを '*' として SID で指定できます。
- SID の指定では大文字と小文字の違いが識別されます。これは、サービスを有効にするためには、すべての構成ファイルで SID の大文字と小文字の違いを正確に指定する必要があります。
- 特定のデータベースを Enterprise Manager で表示できるようにするには、ORATAB ファイルを使用してデータベースが検出されないようにする必要があります。
 1. ORATAB ファイルを他の場所にコピーします。Intelligent Agent が起動時にファイルを読み込むことができれば、ファイルの物理的な場所は重要ではありません。
 2. コピーしたファイルを編集し、検出したくないデータベース・エントリを削除します。
 3. 環境変数を設定します。

```
ORATAB=$ORACLE_HOME/network/agent/oratab; export ORATAB
```

4. Intelligent Agent を再起動します。

ORATAB 解析の 2 つのリストが表示されます。

- 可能なデータベース・サービスのリスト
- マシンに存在する ORACLE_HOME のリスト

フェーズ 2: SQLNET.ORA ファイルの確認

Intelligent Agent は、システムの各 ORACLE_HOME に対して SQL*Net または Oracle Net ファイルを検索します。データベース・サービス定義を取得するには、まず初めに SQLNET.ORA および LISTENER.ORA が必要になります。情報が不足している場合には、TNSNAMES.ORA も必要になります。

Intelligent Agent は次の順序で SQL*Net ファイルを探します。

1. 初めに、環境内で TNS_ADMIN 変数を検索します。見つかった場合、このディレクトリが SQLNET 情報の取得に使用されます。

注意： Intelligent Agent は独自の環境を使用します。ログイン (profile/login スクリプトの実行)、および特定の ORACLE_HOME の情報を取得するための oraenv スクリプトは実行しません。ログイン・スクリプト、または Intelligent Agent 環境とは異なる oraenv スクリプトで特定の TNS_ADMIN 値が必要とされる場合、これらは Intelligent Agent では使用されません。

システムに複数の ORACLE_HOME が存在し、TNS_ADMIN 変数が指定されている場合、2 つめの ORACLE_HOME から開始する NMICONF.LOG ファイルに重複定義の警告が記録されます。これは、Intelligent Agent がそのホームに対して同じ SQL*Net を発見したために起こります。その SQL*Net には同じ情報が含まれます。これらの警告は無視できます。

2. 環境変数 TNS_ADMIN が指定されていない場合、デフォルトのオペレーティング・システム固有の構成の場所内で SQL*Net を検索します。これは、/etc または /var/opt/oracle ディレクトリのいずれかで、使用している UNIX によって異なります。

注意： Intelligent Agent がオペレーティング・システム固有の場所内でファイルを検出し、システムに複数の ORACLE_HOME が存在する場合、NMICONF.LOG ファイルに複数定義の警告が記録されます。

3. 最後に、何も検出されない場合、デフォルトの \$ORACLE_HOME/network/admin 内で必要なファイルを検索します。

注意： 特定の \$ORACLE_HOME で SQL*Net ファイルが検出されない場合、Intelligent Agent は次のホームに移動して、そこにある情報を検索します。すべてのホームのスキャンが終了し、ORATAB ファイルで検出された特定の SID の情報が検出されなかった場合、NMICONF.LOG ファイルに SID がスキップされるという警告が記録されます。これは、Intelligent Agent が異なる ORACLE_HOME にインストールされている場合の一般的な警告です。Intelligent Agent のインストール時、インストール・ルーチンは ORATAB ファイルを調整しますが、インストール時に与えられた SID 名を持つ実際のデータベースは存在しません。

SQL*Net 構成ディレクトリが確立すると、情報の実際の読み込みを開始できます。

SQLNET.ORA ファイルからは 1 つのパラメータ (names.default_domain パラメータ) のみが読み込まれます。

フェーズ 3: LISTENER.ORA ファイルの確認

同じ SQL*Net 構成ディレクトリを使用して、listener.ora ファイルから情報を読み込みます。

2 つの部分が含まれます。

- このリスナーを使用して、サービスにコンタクトする方法のリスト。リスナーの定義自体で指定されているプロトコル定義です。TCP プロトコル定義は、Intelligent Agent がこのデータベースに接続する方法を定義するために Intelligent Agent によって使用されます。また、Intelligent Agent が TNSNAMES.ORA ファイルをスキャンする必要があります。また、Intelligent Agent が TNSNAMES.ORA ファイルをスキャンする必要があります。これにより、どちらの TNS エントリがどちらのデータベースのリスナー定義に適合するかを検証します。

注意 1： プロトコル定義が、多くの IPC エントリおよびその他の非 TCP 定義を含んでいる場合、プロトコル定義の長さは非常に長くなることがあります。

データベースの SID 説明を追加したプロトコル定義の長さが、SQL*Net の限度である 256 文字を超えると、Intelligent Agent が構成した TNS 文字列を使用して接続する際に、エラー ORA-12163 が発生します。

ORA-12163: TNS: 接続記述子が長すぎます

これを回避するには、リスナー定義から非 TCP エントリのいくつかを削除し、Intelligent Agent を再起動します。

注意 2: TCP 定義の HOST 変数で定義されているホスト名が、Intelligent Agent で検出されたホスト名と一致しない場合、NMICONF.LOG にメッセージが記録されます。

Could not find corresponding listener ...

このリスナーによってリスニングされている SID はスキップされ、またその他のリスナーがリスニングしていない場合、次のメッセージが記録されます。

No listener found for SID ...

- このリスナーが動作しているすべてのデータベースのリスト。LISTENER.ORA ファイルの SID の説明部分で検出される各 SID が検証されます。
 - extproc SID の説明は自動的にスキップされます。
 - SID が ORATAB ファイルに存在しない場合、スキップされます。
 - システムの他の LISTENER.ORA ファイルで SID がすでに検出されている場合は、スキップされます。

メッセージが NMICONF.LOG に記録されます。

例：

Warning : Multiple Listeners found for SID ORCL.

- 検出された残りのすべての SID に対してサービス名が生成されます。このサービス名はマシン上で一意です。GLOBAL_DBNAME パラメータを持つ LISTENER.ORA ファイルでサービス名が強要されない場合、TNSNAMES.ORA ファイルで有効なサービス名が検索されます。

重複したサービス名が検出された場合、Intelligent Agent はこのデータベースに対して新しい一意のサービス名を構成します。NMICONF.LOG に、新しく構成された名前に対する警告メッセージが示されます。

注意： 異なるマシンで同じサービス名を使用している場合、Enterprise Manager ナビゲータからの検出に失敗し、その重複したサービス名が Enterprise Manager で管理されているサービスのリストに追加されます。

結果としてリスナーのリストが作成されます。さらに、各リスナーに対して、リスナーが動作する SID のリストが作成されます。これらのリストの各 SID には、それぞれそのデータベース・サービスに必要とされる詳細のリストが存在します。

フェーズ 4: 情報の検証

すべてのファイルが解析および処理され、すべてのサービスが検出されると、Intelligent Agent はすべての情報が存在し、それらが有効であるかどうかを直ちに検証します。

- SID を持たないリスナーが存在する場合、リスナーがスキップされるという警告が NMICONF.LOG に示されます。

例：

```
Warning :Listener LISTENER defined in  
/oracle/815/network/admin/listener.ora will be ignored.
```

- データベースに対して動作しているリスナーを持たないデータベースが ORATAB で検出されると、SID は検出されません。

NMICONF.LOG には、SID がスキップされるというメッセージが示されます。

例：

```
Warning :No Listener found for SID ORCL.ORCL will be skipped
```

残りのすべての情報は検出されたとみなされ、検出ファイル SNMP_RO.ORA、SNMP_RW.ORA および SERVICES.ORA に配置されます。

重要： SERVICES.ORA および SNMP_RO.ORA ファイルは常に完全に再書き込みされます。ただし、SNMP_RW.ORA ファイルは、新しいサービスの情報でのみ更新されます。そのため、索引にギャップが生じたり、マシンにすでに存在しないサービスについての情報がこのファイルで検出される可能性が多くなります。

Window NT での Intelligent Agent の検出エラーの診断

Windows NT での検出プロセスは次の操作を伴います。

- 主ホスト名およびその別名の取得。
- レジストリから Oracle サービスのリストを取得します。対応する ORACLE_HOME のデータベースはこのリストから取得されます。
- 検出された各 ORACLE_HOME には、SQL*Net ファイル LISTENER.ORA および SQLNET.ORA が読み込まれます。データベース・サービスの名前は LISTENER.ORA ファイルにより決定されます。これらの名前は、Enterprise Manager コンソールで表示可能な名前です。またこれらの名前は、データベースに接続するための TNS の別名として使用されます。
- データベースの LISTENER.ORA ファイルで指定されている GLOBAL_DBNAME パラメータが存在しない場合、TNSNAMES.ORA ファイル内をスキャンして利用可能な TNS の別名を検索します。

- TNS の別名が見つからない場合、または重複した TNS の別名の文字列が存在する場合、Intelligent Agent はデータベースの TNS の別名を生成し、この生成された名前をサービス名として使用します。

検出時に使用されるファイル

listener.ora : 受信 SQL*Net 接続の定義のファイル

\$ORACLE_HOME ごとに 1 ファイル

次のいずれかに存在します（検索はこの順序で行われます）。

- 環境内の %TNS_ADMIN%
- レジストリ内の %TNS_ADMIN%
- %ORACLE_HOME%\network\admin

nmiconf.log : Intelligent Agent の検出警告 / エラーのファイル

Intelligent Agent ごとに 1 つ

次に存在します。

- %ORACLE_HOME%\network\log

nmiconf.lst : サード・パーティ製の追加の検出スクリプトのリスト

Intelligent Agent ごとに 1 つ

次に存在します。

- %ORACLE_HOME%\net80\agent\config （バージョン 8.0.X のみ）
- %ORACLE_HOME%\network\agent\config

nmiconf.tcl : Intelligent Agent の検出スクリプト

Intelligent Agent ごとに 1 つ

次に存在します。

- %ORACLE_HOME%\net80\agent\config （バージョン 8.0.X のみ）
- %ORACLE_HOME%\network\agent\config

services.ora : エージェントが検出したすべてのサービス定義のファイル

Intelligent Agent ごとに 1 つ

次に存在します。

- %ORACLE_HOME%\net80\agent （バージョン 8.0.X のみ）
- %ORACLE_HOME%\network\agent

snmp_ro.ora : すべての読み込み専用サービス情報のファイル

Intelligent Agent ごとに 1 つ

次のいずれかに存在します（優先度順）。

- 環境内の %TNS_ADMIN%
- レジストリ内の %TNS_ADMIN%
- %ORACLE_HOME%\net80¥admin（バージョン 8.0.X のみ）
- %ORACLE_HOME%\network¥admin

snmp_rw.ora : すべての更新可能なサービス情報のファイル

Intelligent Agent ごとに 1 つ

次のいずれかに存在します（優先度順）。

- 環境内の %TNS_ADMIN%
- レジストリ内の %TNS_ADMIN%
- %ORACLE_HOME%\net80¥admin（バージョン 8.0.X のみ）
- %ORACLE_HOME%\network¥admin

sqlnet.ora : SQL*Net 固有のパラメータのファイル

\$ORACLE_HOME ごとに 1 ファイル

次のいずれかに存在します。

- 環境内の %TNS_ADMIN%
- レジストリ内の %TNS_ADMIN%
- %ORACLE_HOME%\net80¥admin（バージョン 8.0.X のみ）
- %ORACLE_HOME%\network¥admin

tnsnames.ora : データベースに接続するための TNS の別名のファイル

\$ORACLE_HOME ごとに 1 ファイル

次のいずれかに存在します（優先度順）。

- 環境内の %TNS_ADMIN%
- レジストリ内の %TNS_ADMIN%
- %ORACLE_HOME%\net80¥admin（バージョン 8.0.X のみ）
- %ORACLE_HOME%\network¥admin

NT での検出問題の診断

フェーズ 1: レジストリのスキャン

レジストリはデータベース・サービスに対してスキャンされます。検出された各 OracleService NT サービスに対して、有効なデータベース・サービス・エントリが作成され、対応する ORACLE_HOME が決定されます。

確認事項：

- SID の指定では大文字と小文字の違いは識別されません。
- UNIX での ORATAB の場合のように、ここではデータベースの検出を回避できません。レジストリのスキャンによって 2 つのリストが生成されます。
- 可能なデータベース・サービスのリスト
- マシンに存在する ORACLE_HOME のリスト

フェーズ 2: SQLNET.ORA ファイルのスキャン

Intelligent Agent は、システムの各 ORACLE_HOME に対して SQL*Net/Oracle Net ファイルを検索します。データベース・サービス定義を取得するには、まず初めに SQLNET.ORA および LISTENER.ORA が必要になります。情報が不足している場合には、TNSNAMES.ORA が必要になります。

Intelligent Agent は次の順序で SQL*Net ファイルを探します。

1. 環境内で TNS_ADMIN 変数を検索します。見つかった場合、このディレクトリが SQLNET/Oracle Net 情報の取得に使用されます。
2. 次に、Intelligent Agent が検証する ORACLE_HOME の HOME_x でレジストリを確認し、TNS_ADMIN 変数を検索します。見つかった場合、それを使用します。
3. 最後に、何も検出されない場合、デフォルトの %ORACLE_HOME%\net80\admin (バージョン 8.0.X のみ) または %ORACLE_HOME%\network\admin 内で必要なファイルを検索します。

注意： 特定の %ORACLE_HOME% で SQL*Net/Oracle Net ファイルが検出されない場合、Intelligent Agent は次のホームに移動して、そこにある情報を検索します。すべての ORACLE_HOME のスキャンが終了し、レジストリで検出された特定の SID の情報が検出されなかった場合、SID がスキップされるという警告が NMICONF.LOG ファイルに記録されます。

SQL*Net/Oracle Net 構成ディレクトリが確立すると、情報の実際の読み込みを開始できます。

SQLNET.ORA ファイルからは 1 つのパラメータ (names.default_domain パラメータ) のみが読み込まれます。

フェーズ 3: LISTENER.ORA ファイルの確認

同じ SQL*Net/Oracle Net 構成ディレクトリを使用して、listener.ora ファイルから情報を読み込みます。

2 つの部分が含まれます。

- このリスナーを使用して、サービスにコンタクトする方法のリスト。このリストは、リスナーの定義自体で指定されているプロトコル定義です。TCP プロトコル定義は、Intelligent Agent がこのデータベースに接続する方法を定義するために Intelligent Agent によって使用されます。また、Intelligent Agent が TNSNAMES.ORA ファイルをスキャンする必要がある場合にも使用されます。これにより、どちらの TNS エントリがどちらのデータベースのリスナー定義に適合するかを検証します。

注意： プロトコル定義が、多くの IPC エントリおよびその他の非 TCP 定義を含んでいる場合、プロトコル定義の長さは非常に長くなることがあります。データベースの SID 説明を追加したプロトコル定義の長さが、SQL*Net/Oracle Net の限度である 256 文字を超えると、Intelligent Agent が構成した TNS 文字列を使用して接続する際に、エラー ORA-12163 が発生します。

ORA-12163: TNS: 接続記述子が長すぎます

これを回避するには、リスナー定義から非 TCP エントリのいくつかを削除し、Intelligent Agent を再起動します。

注意： TCP 定義の HOST 変数で定義されているホスト名が、Intelligent Agent で検出されたホスト名と一致しない場合、NMICONF.LOG にメッセージが記録されます。

Could not find corresponding listener ...

このリスナーによってリスニングされている SID はスキップされ、またその他のリスナーがリスニングしていない場合、次のメッセージが記録されます。

No listener found for SID ...

- このリスナーが動作しているすべてのデータベースのリスト。LISTENER.ORA ファイルの SID の説明部分で検出される各 SID が検証されます。
 - extproc SID の説明は自動的にスキップされます。

- SID がレジストリ・ファイルに存在しない場合、スキップされます。
- システムの他の LISTENER.ORA ファイルで SID がすでに検出されている場合は、スキップされます。メッセージが NMICONF.LOG に記録されます。

例：

Warning : Multiple Listeners found for SID ORCL.

- 検出された残りのすべての SID に対してサービス名が生成されます。このサービス名はマシン上で一意です。GLOBAL_DBNAME パラメータを持つ LISTENER.ORA ファイルでサービス名が強要されない場合、TNSNAMES.ORA ファイルで有効なサービス名が検索されます。

重複したサービス名が検出された場合、Intelligent Agent はこのデータベースに対して新しい一意のサービス名を構成します。NMICONF.LOG に、新しく構成された名前に対する警告メッセージが表示されます。

最終的な結果として、リスナーのリストが提供されます。そのリストには、各リスナーに対してこのリスナーが動作する SID のリストが示されます。これらのリストの各 SID には、それぞれそのデータベース・サービスに必要とされる詳細のリストが存在します。

フェーズ 4: 情報の検証

すべてのファイルが解析および処理され、すべてのサービスが検出されると、Intelligent Agent はすべての情報が存在し、それらが有効であるかどうかを直ちに検証します。

- SID を持たないリスナーが存在する場合、NMICONF.LOG にリスナーがスキップされるという警告が表示されます。

例：

Warning : Listener LISTENER defined in C:\ORA920\network\admin\listener.ora will be ignored

- データベースが、データベースにサービスするリスナーを持たない場合、SID は検出されません。NMICONF.LOG には、SID がスキップされるというメッセージが示されます。

例：

Warning : No Listener found for SID ORCL. ORCL will be skipped

- 起動時に Intelligent Agent は検出された各データベースへの接続を試行します。そのため、Intelligent Agent が起動すると、オープン状態のデータベースのそれぞれは 2 つの dbsnmp セッションが必要になります。

Select username from v\$session where username = 'DBSNMP'

データが何も取得されず、Intelligent Agent が起動している場合、ユーザー DBSNMP が存在しているか、およびその接続が有効であるかを確認します。

- 残りのすべての情報は検出されたとみなされ、検出ファイル SNMP_RO.ORA、SNMP_RW.ORA および SERVICES.ORA に配置されます。

重要： SERVICES.ORA および SNMP_RO.ORA ファイルは常に完全に再書き込みされます。

ただし、SNMP_RW.ORA ファイルは、新しいサービスの情報でのみ更新されます。そのため、索引にギャップが生じたり、マシンにすでに存在しないサービスについての情報がこのファイルで検出される可能性が多くあります。

用語集

Oracle プライベート MIB (Oracle Private MIB)

Oracle 製品のものに特有の MIB。

RDBMS パブリック MIB (RDBMS Public MIB)

Internet Engineering Task Force (IETF) に承認されているリレーショナル・データベースの標準 MIB。RDBMS パブリック MIB では、たとえば `rdbsDbName`, `1.3.6.1.3.55.1.2.1.4` のようなデータベース名など、一般のリレーショナル・データベースに関連する様々な OID がサポートされる。

SNMP: Simple Network Management Protocol

OID を操作するネットワーク・プロトコル。Oracle の場合は、2 種類のプリミティブな SNMP 操作のみがサポートされている。`oid` の値をフェッチする `get oid` と、`oid` の後に来る次の OID の値を取得する `getnext oid` である。

SNMP オブジェクト ID (SNMP Object ID: OID)

`a.b.c...x.y.z` という構成の、ピリオドで区切られた連続した数字。MIB の一部である情報の 1 項目に対する一意の識別子である。一般に OID には、それぞれに関連のある名前が付けられる。OID は本来は階層構造を有する。そのため、`1.2.3` は `1.3` の前に来るが、`1.2` の後になる。たとえば、Oracle7 データベースによって実行された物理的読込みの数を含む OID は次のようになる。

`oraDbSysPhysReads, 1.3.6.1.4.1.111.4.1.1.1.8`

イベント (Event)

イベントとは、Intelligent Agent に監視されるデータベースあるいはノードのいずれかで発生する可能性のある状態である。たとえば、突然停止したデータベースは、DBDOWN イベントとなる。イベントは、次の 2 通りの方法のいずれかで検出できる。(1) 定期的に Tcl スクリプトを実行し、特定の状態を監視する。(2) サード・パーティを利用してイベントの発生を Agent に直接報告させる。

管理情報ベース (Management Information Base: MIB)

SNMP オブジェクト ID (OID) のコレクション。通常は相互に関連付けられている。

修正ジョブ (Fixit Job)

イベントの発生によって実行される特殊なジョブ。たとえば、`tablespace full` イベントによって表領域が 90% 以上埋まっていることが検出されると、修正ジョブが自動的に実行され、表領域にデータ・ファイルが追加される。

ジョブ (Job)

ジョブとは、1 度あるいは繰り返し発生するスケジュールで実行できる Tcl スクリプトである。特定の状態を監視するイベントとは異なり、ジョブは特定のタスクを完了することが期待される。ジョブの例には、バックアップやデータベースの起動などがある。

索引

A

agentctl.trace_directory, A-3
agentctl.trace_file, A-3
agentctl.trace_level, A-2
agentctl.trace_timestamp, A-3

C

convertin, 3-14
convertout, 3-14

D

Data Gatherer、トラブルシューティング, B-34
dbsnmp.addnl_db_conns, A-4
dbsnmp.address, A-8
dbsnmp.avg_occ_per_event, A-7
dbsnmp.cs_base_port, A-8
dbsnmp.hostname, A-6
dbsnmp.log_directory, A-7
dbsnmp.log_file, A-7
dbsnmp.log_unique, A-8
dbsnmp.trace_directory, A-7
dbsnmp.trace_file, A-7
dbsnmp.trace_filecnt, A-7
dbsnmp.trace_filesize, A-7
dbsnmp.trace_level, A-7
dbsnmp.trace_timestamp, A-8
dbsnmp.trace_unique, A-7
dbsnmp.log_directory, A-6
dbsnmp.log_file, A-6
dbsnmp.log_unique, A-6
dbsnmp.no_job_skipped_notifications, A-4
dbsnmp.noheuristic, A-5

dbsnmp.notificationtimeout, A-4
dbsnmp.polltime, A-4
dbsnmp.spawnaddress, A-8
dbsnmp.threshold_evocc, A-6
dbsnmp.threshold_job_status, A-6
dbsnmp.trace_directory, A-5
dbsnmp.trace_file, A-5
dbsnmp.trace_filecnt, A-5
dbsnmp.trace_filesize, A-6
dbsnmp.trace_level, A-5
dbsnmp.trace_timestamp, A-6
dbsnmp.trace_unique, A-6
dbsnmp.pwd, 2-8
DNS 名, B-10

I

Intelligent Agent の ping, B-12
Intelligent Agent のインストール, 2-2
Intelligent Agent のトレース, B-23
Intelligent Agent ログ・ファイル, A-9

M

msgtxt, 3-15
msgtxt1, 3-15

N

NLS の問題とエラー・メッセージ, 3-11
NT/2000 の起動時のエラー, B-28
NT での起動, 2-2
NT での停止, 2-3
NT ユーザー・アカウン
 権限, 2-4

作成, 2-4
ジョブ, 2-3

O

oraautocom, 3-16
oracancel, 3-16
Oracle Names, 2-15
Oracle Net 構成ファイル, B-10
oraclose, 3-17
oracols, 3-17
oracommith, 3-18
oradbsnmp, 3-18
orafail, 3-19
orafetch, 3-19
orainfo, 3-21
orajobstat, 3-21
oralogoff, 3-22
oralogon, 3-22, 3-23
oramsg (agent_characterstet), 3-5
oramsg (collengths), 3-6
oramsg (colprec), 3-6
oramsg (colscale), 3-6
oramsg (coltypes), 3-6
oramsg (db_characterstet), 3-6
oramsg (errortxt), 3-6
oramsg (handle), 3-6
oramsg (jobid), 3-6
oramsg (language), 3-6
oramsg (maxlong), 3-6
oramsg (nullvalue), 3-7
oramsg (orahome), 3-7
oramsg (oraindex), 3-7
oramsg (orainput), 3-7
oramsg (oraobject), 3-7
oramsg (rc), 3-7
oramsg (rows), 3-8
oramsg (starttime), 3-8
oramsg の要素, 3-5
oraopen, 3-23
oraplexec, 3-23
orareadlong, 3-24
orareportevent, 3-25
oraroll, 3-26
orasleep, 3-26
orasnmp, 3-27
orasql, 3-28

orastart, 3-29
orastop, 3-29
OraTcl, 3-4
OraTcl、関数とパラメータ, 3-12
OraTcl の例, 3-4
orertime, 3-30
orawritelong, 3-30

R

RDBMS 管理用関数, 3-12
Real Application Clusters, 2-21
root.sh, 2-7

S

services.ora, A-2
snmp_ro.ora, A-2
snmp_rw.ora, A-2
snmp.connect.password, A-3
snmp.connect.user, A-3
snmp.contact., A-4
snmp.index.service_name.world, A-3
SNMP、UNIX の構成, 2-12
SNMP アクセス用関数, 3-12
SNMP の構成, 2-6
SQL および PL/SQL 関数, 3-12

T

TCP/IP、正しい設定, B-8
trigevent (arguments), 3-9
trigevent (name), 3-8
trigevent (object), 3-8
trigevent (results), 3-9
trigevent (severity), 3-9
trigevent の要素, 3-8
Tcl、Agent での使用, 3-10

U

UNIX Agent、制御, 2-7
UNIX での Intelligent Agent
起動と停止, 2-8
UNIX での検出プロセス, 2-20

W

Windows 2000、ユーザー・アカウント、2-4
Windows NT での Intelligent Agent の操作、2-2
Windows NT での起動、2-2
Windows NT での検出プロセス、2-19

あ

アップグレード、2-21
アップグレード・プロセス、Agent-Enterprise
Manager、2-26
暗号化のパスワード、2-9

い

一般的な設定、2-1
イベント・システム、トレース、B-28
イベントのトレース、B-28
イベント・ログ・メッセージ (NT)、B-28

え

エラー・メッセージおよび解決方法、B-15

か

概要、1-2
簡単なチェック、NT、B-2
簡単なチェック、UNIX、B-5

き

起動、Intelligent Agent の初期化、B-37
起動時のトレース・パラメータ、B-26
起動時の問題、B-12
キャラクタ・セット変換およびエラー処理用関数、
3-13
共通パラメータ、3-13
column、3-13
connect_string、3-13
destaddress、3-13
filename、3-13
logon-handle、3-13
rowid、3-14
table、3-14
許可、NT、B-15

許可、UNIX、B-15

け

検出エラー、NT、B-46
検出エラー、UNIX、B-20
検出エラーの診断、UNIX、B-40
検出、使用されるファイル (NT)、B-47
検出、使用されるファイル (UNIX)、B-41

こ

構成、2-1
構成の問題、UNIX、B-32
構成ファイル、A-1

さ

サーバー・メッセージ、OraTcl、3-5
サービス検出プロセス、2-19
最適な方法、2-22
Agent のインストール、2-22
Intelligent Agent の構成、2-25
Intelligent Agent の互換性、2-26

し

自動検出、2-17
前提条件、2-18
ジョブ・スクリプトとイベント・スクリプト、3-2
ジョブ・トレース・パラメータ、B-25
ジョブを実行するための NT ユーザー・アカウント、
2-3

す

スクリプト、ジョブとイベント、3-2

せ

接続のテスト、B-12

つ

通信用関数、3-13

て

データ収集サービス、トレース, B-27

データ収集のトレース, B-27

デバッグ、OS エラー, B-20

と

ドメイン・ユーザー、設定, 2-5

トラブルシューティング, B-2

トレース, B-23

トレース・パラメータ, B-24

は

汎用ユーティリティ関数, 3-13

ふ

複数のネットワーク・カード、Agent の動作, 2-14

複数のネットワーク・カード、構成, 2-13

ブラックアウト, 2-10

 commands, 2-10

 コマンドの例, 2-11

 定義, 2-10

ゆ

ユーザーが構成可能なパラメータ, A-2

ろ

ロールとユーザー, 2-16

ログ・ファイル、Intelligent Agent, A-9