

# Oracle Intelligent Agent

ユーザーズ・ガイド

リリース 8.1.7

2000 年 11 月

部品番号 : J02262-01

ORACLE®

---

Oracle Intelligent Agent ユーザーズ・ガイド リリース 8.1.7

部品番号 : J02262-01

原本名 : Oracle Intelligent Agent User's Guide, Release 8.1.7

原本部品番号 : A85251-01

Copyright © 1996, 2000, 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います。

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 .....	v
------------	---

## 1 Intelligent Agent の概要

Oracle Intelligent Agent: 概要 .....	1-1
特徴 .....	1-2
ジョブおよびイベントのサポート .....	1-2
SNMP (Simple Network Management Protocol) のサポート .....	1-3
データ収集サービスの説明 .....	1-3
データ収集サービスのインストール方法 .....	1-3
データベースとオペレーティング・システムのデータを収集するための Oracle Data Gatherer の 構成位置の理解 .....	1-3
仲介ホスト上での Oracle Data Gatherer の実行 .....	1-4

## 2 インストールと構成

Windows NT 上での Intelligent Agent のインストール .....	2-1
ジョブ実行のための Windows NT ユーザー・アカウントの作成 .....	2-2
ドメイン・ユーザーを Agent のユーザーとして構成 .....	2-3
NT Agent の運用制御 .....	2-4
UNIX 上での Oracle Intelligent Agent のインストール .....	2-5
root.sh シェル・スクリプトの実行 .....	2-5
UNIX Agent の運用制御 .....	2-7
UNIX または Windows NT での SNMP の構成 .....	2-7
複数のネットワーク・カードと使用するためのリリース 8.1.7 Agent の構成 .....	2-7
複数のネットワーク・カード使用時の Agent の動作 .....	2-8
Oracle Intelligent Agent および Oracle Names .....	2-9

Agent に必要なロールとユーザー .....	2-10
自動検出 .....	2-11
自動検出のための前提条件 .....	2-12
サービス検出プロセス .....	2-13
Agent の検出プロセス (Windows NT) .....	2-13
Agent の検出プロセス (UNIX) .....	2-14
Intelligent Agent のアップグレード .....	2-15
Enterprise Manager バージョン 2 の Agent アップグレード・ガイドライン .....	2-15
Enterprise Manager ジョブ / イベントの 8.1.5 または 8.1.6 Agent から 8.1.7 Agent への移行 .....	2-16
Data Gatherer の構成 .....	2-16
別の Oracle ホームに新しい Data Gatherer をインストールする時の前のバージョンからの アップグレード .....	2-17
Collection Data の Agent Data Gatherer リリース 8.1.7 への移行 .....	2-18
NMUMIGRATE ユーティリティの機能 .....	2-19
NT および UNIX でのデータ収集サービスの運用制御 .....	2-20
Windows NT 上でのその他の運用制御方法 .....	2-21
データ収集サービスの動作確認 .....	2-22
Oracle Data Gatherer 上でのトレース情報の取得 .....	2-22

### 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 .....	3-1
Tcl 言語の説明 .....	3-2
OraTcl の説明 .....	3-3
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 .....	3-11
共通パラメータ .....	3-12
convertin .....	3-13
convertout .....	3-13
msgtxt .....	3-14
msgtxt1 .....	3-14
oraautocom .....	3-15

oracancel .....	3-15
oraclose .....	3-15
oracols .....	3-16
oracommit .....	3-16
oradbsnmp .....	3-17
orafail .....	3-17
orafetch .....	3-18
orainfo .....	3-19
orajobstat .....	3-19
oralogoff .....	3-20
oralogon .....	3-20
oraopen .....	3-21
oraplexec .....	3-21
orareadlong .....	3-22
orareportevent .....	3-22
oraroll .....	3-24
orasleep .....	3-24
orasnmp .....	3-24
orasql .....	3-25
orastart .....	3-26
orastop .....	3-27
orertime .....	3-27
orawritelong .....	3-27

## A 構成ファイル

構成ファイル .....	A-1
snmp_ro.ora .....	A-1
snmp_rw.ora .....	A-1
services.ora .....	A-1
ユーザーが構成可能なパラメータ .....	A-2

## 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-1
簡単なチェック .....	B-2
Windows NT Agent の簡単なチェック .....	B-2
UNIX Agent の簡単なチェック .....	B-3
質問と回答 .....	B-5

TCP/IP の構成と動作の確認 .....	B-6
DNS 名およびコンピュータ名の一致の確認 (Windows NT) .....	B-7
Net8 構成ファイルの確認 .....	B-8
Net8 の動作の確認 .....	B-9
Agent の正常起動の確認 .....	B-10
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-12
Agent が正しい許可で動作していることの確認 (UNIX) .....	B-12
OS 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-12
リリース 7.3.3 またはそれ以前の Agent を使用しているかどうかの確認 .....	B-13
エラーの有無の確認 .....	B-13
ジョブが実行されたにもかかわらず、Agent が状態通知を Enterprise Manager コンソールに 返信しない理由 .....	B-13
<b>Intelligent Agent の起動に関する問題および解決策 .....</b>	<b>B-14</b>
Agent 全般 .....	B-14
NT Agent .....	B-19
UNIX Agent .....	B-22
<b>Intelligent Agent のエラー・メッセージと解決策 .....</b>	<b>B-23</b>
Agent 全般 .....	B-23
NT Agent .....	B-25
UNIX Agent .....	B-26
Agent のトレース .....	B-28
TCL のトレース .....	B-29
Oracle Data Gatherer の構成に関するトラブルシューティング .....	B-30

## C キーボード・ショートカット

### 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 1 およびバージョン 2 の Oracle Enterprise Manager の利用者に加えて、Intelligent Agent を通して 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読者としています。

---

---

**重要：** Enterprise Manager バージョン 2 では、中間層 Management Server を使用します。Enterprise Manager バージョン 1 のユーザーは、このコンポーネントに関しては参照する必要がありません。

---

---

## 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、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 と検出サービスの機能の概要について説明します。
第 2 章	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	Tcl (Tool Command Language) を使用した、ジョブおよびイベントのスク립ト記述について説明します。
付録 A	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	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。
付録 C	一般的な Windows のキーボード・ショートカットのリストが記載されています。

## 関連ドキュメント

詳細は、次の各ドキュメント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』



---

# Intelligent Agent の概要

この章では、Intelligent Agent の概要について簡単に説明します。次の項目について説明します。

- [Oracle Intelligent Agent: 概要](#)
- [データ収集サービスの説明](#)
- [データ収集サービスのインストール方法](#)
- [データベースとオペレーティング・システムのデータを収集するための Oracle Data Gatherer の構成位置の理解](#)

## 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。Agent は、サポートするサービスと同じノード上に存在します。ただし、1 ノード上であっても、Agent は複数のサービスをサポートできます。たとえば、1 台のマシン上に 2 つのデータベースがインストールされている場合、1 つの Agent で両方のデータベースをサポートできます。Agent が備える機能には、次のようなものがあります。

- 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。
- Oracle Management Server またはその他のサードパーティ製アプリケーションから、ジョブまたはイベントを受け取ります。
- Oracle Enterprise Manager のジョブの実行、ジョブの結果および出力の収集、および必要に応じてその結果をキューイングします。
- 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。
- 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。

- Agent のプラットフォーム上で SNMP (Simple Network Management Protocol) がサポートされている場合に、SNMP 要求を処理します。

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 Server インストール・マニュアルを参照してください。

---

---

**注意：** Enterprise Manager バージョン 1 では、バージョン 2 の中間層 Management Server は使用されません。このマニュアルは、バージョン 1 およびバージョン 2 の両方のユーザーを対象としています。したがって、Enterprise Manager バージョン 1 のユーザーは、Management Server についての説明を参照する必要はありません。

---

---

## 特徴

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 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、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、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、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。

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。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。ジョブおよびイベントの状態に関する情報は、Agent の動作するノード上のファイルに格納されます。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。

## ジョブおよびイベントのサポート

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 Agent がジョブまたはテストを実行するとき、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 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。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。

## SNMP（Simple Network Management Protocol）のサポート

Agent は SNMP（Simple Network Management Protocol）をサポートします。これにより、サードパーティ製のシステム管理フレームワークは、SNMP を使用して Agent から SNMP トラップを直接受け取ることができます。Agent では、Oracle のデータベース MIB（Management Information Base）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。

## データ収集サービスの説明

データ収集サービス（Oracle Data Gatherer と呼ばれます）は、パフォーマンス・データを収集するために使用されます。

Oracle Data Gatherer の役割は、データの収集を必要とするクライアント・アプリケーション（Oracle Capacity Planner や Oracle Performance Manager など）からの要求を処理することです。各クライアント・アプリケーションに対して、収集するパフォーマンス・データの種類（ファイル I/O、CPU 使用率データなど）や、データ・サンプル間の時間間隔を指定します。Oracle Data Gatherer は次に、クライアントに対して要求されたデータを指定の間隔で収集します。

## データ収集サービスのインストール方法

Oracle Data Gatherer は Oracle Intelligent Agent の一部であり、管理対象のホスト上に Agent がインストールされるときに自動的にインストールされます。したがって、これ以降の項で、あるホスト上に Oracle Data Gatherer がインストールされているという表記は、そのホストに Agent をインストールした際に Oracle Data Gatherer サービスが同時にインストールされたことを意味します。

## データベースとオペレーティング・システムのデータを収集するための Oracle Data Gatherer の構成位置の理解

Oracle Data Gatherer は、次のものに関するパフォーマンス・データを収集します。

- データベース
- オペレーティング・システム
- その他のサポートされるソース

一般に、Oracle Data Gatherer は、監視対象のターゲット（データベースまたはホスト）が実際にあるホスト上に構成してください。言い換えると、Oracle Data Gatherer を使用してホスト BENEFITS 上の MEDICAL データベースからデータベース・データを収集する場合は、Oracle Data Gatherer をホスト BENEFITS 上にインストールおよび構成するということです。同様に、Oracle Data Gatherer を使用してホスト EMPLOYEES 上のオペレーティング・システム・データを収集する場合は、Oracle Data Gatherer をホスト EMPLOYEES 上にインストールおよび構成します。

ただし、一部のケースでは、特定のホスト上に Oracle Data Gatherer をインストールおよび構成できないことがあります。Oracle Data Gatherer は、次の必要条件の両方が満たされているときに限り、ホスト上にインストールできます。

1. ホストのオペレーティング・システムが、Oracle Data Gatherer のインストールおよび構成をサポートしている場合。
2. ホスト上に、Oracle 8.0.4 またはそれ以降のための Oracle ホームが存在する場合。ホスト上には Oracle 8.0.4 またはそれ以降のための Oracle ホーム・ディレクトリのみが必要であり、Oracle 8.0.4 またはそれ以降のデータベース自体は必須ではありません。

ホストがこれらの必要条件を満たさない場合、そのホスト上に Oracle Data Gatherer をインストールできません。つまり、Oracle Data Gatherer はそのホストのオペレーティング・システム統計を収集できません。ただし、データベース・データに関しては、Oracle Data Gatherer をインストールおよび構成できないホストからでも収集できます。

## 仲介ホスト上での Oracle Data Gatherer の実行

Oracle Data Gatherer をインストールおよび構成できないホストからデータベース・データを収集するには、前述の必要条件を両方とも満たす別のホスト上に Oracle Data Gatherer をインストールし、構成します。次に、そのホスト上の Oracle Data Gatherer を利用して、Oracle Data Gatherer がインストールされていないホスト上のデータベース・データをリモートで収集します。これは、仲介ホスト Oracle Data Gatherer の使用と呼ばれます。

たとえば、ホスト EMPLOYEES 上の CONSULTANTS データベースのデータベース・データを収集するときに、Oracle Data Gatherer をホスト EMPLOYEES 上にインストールおよび構成できないとします。Oracle Data Gatherer がホスト BENEFITS 上にインストールおよび構成されている場合、BENEFITS 上の Oracle Data Gatherer を、仲介ホスト Oracle Data Gatherer として利用できます。BENEFITS 上の Oracle Data Gatherer は、ホスト EMPLOYEES 上の CONSULTANTS データベースからデータベース統計を収集できます。

データベース・データを収集するホスト上で、プロセスのフットプリント / オーバーヘッドをそれ以上許容できない場合にも、仲介ホスト Oracle Data Gatherer を利用できます。この状況での収集アクティビティも、同様にデータベースに対して行われます。収集アクティビティに伴うこの最小限のオーバーヘッドは、Oracle Data Gatherer がどのホスト上にあるかに関係なく発生します。

---

## インストールと構成

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。

- [Windows NT 上での Intelligent Agent のインストール](#)
- [NT Agent の運用制御](#)
- [UNIX 上での Oracle Intelligent Agent のインストール](#)
- [UNIX または Windows NT での SNMP の構成](#)
- [Oracle Intelligent Agent および Oracle Names](#)
- [Agent に必要なロールとユーザー](#)
- [自動検出](#)

### Windows NT 上での Intelligent Agent のインストール

Intelligent Agent はデータベースとともに出荷され、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されます。Agent はサービスとして動作します。

サービス名の構成には、次の規約が使用されます。

Oracle<ORACLE\_HOME\_NAME>Agent

ORACLE\_HOME\_NAME は、インストール中に指定した ORACLE\_HOME の名前です。

Intelligent Agent のインストールの詳細は、『Oracle Server のインストレーション・ガイド』を参照してください。

## 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 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 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 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。
- 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取り / 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、ノード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。

次の手順のどちらかを実行してください。

### 新規 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 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。

## ドメイン・ユーザーを 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 およびそれ以前の Agent ではサポートされていません。

---

---

リポジトリ・ユーザーを 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。

---

---

## NT Agent の運用制御

この項には、Windows NT および DOS プロンプトでの Agent の制御に関する情報が記されています。また、Agent のトラブルシューティングに関する項も含まれています。

---

---

**注意：** Oracle Enterprise Manager および Agent は、Net8 を使用して該当するデータベースと通信します。Agent の使用を試みる前に、該当する SID すべてに Net8 が接続できることを確認してください。

---

---

## Windows NT 上での Intelligent Agent の起動

Windows NT 上で 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は「手動」に設定されています。これは、ユーザーが手動で 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 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. 「起動」ボタンをクリックして、Agent を起動します。

---

---

**注意：** コマンド行で次のように入力しても、Agent を起動できます。

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

---

---

## Windows NT 上での Intelligent Agent の停止

Windows NT 上で 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. 「OracleAgent」サービスを選択します。
3. 「停止」 ボタンをクリックして、Agent を停止します。

---

**注意：** コマンド行から次のように入力しても、Agent を停止できます。

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

---

## Agent の動作検証

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「OracleAgent」が表示されます。

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

# UNIX 上での Oracle Intelligent Agent のインストール

『Oracle Server のインストレーション・ガイド』の説明に従って、Oracle CD-ROM から Oracle Intelligent Agent をインストールします。Intelligent Agent は独立したコンポーネントとして、インストール時の選択対象となります。

## root.sh シェル・スクリプトの実行

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 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、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。

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 Agent がインストールされます。これにより Agent は、ホストに対する優先接続情報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。

Agent が `setuid` プログラムでない場合、Enterprise Manager のすべてのジョブは、Agent を起動したユーザーの許可のもとで実行されます。

---

---

**注意：** Agent が `setuid root` に設定されることは、`root` ユーザーが 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 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 Agent に送るユーザーには、Agent の `ORACLE_HOME` ディレクトリに、読取り / 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、Agent に送られるすべてのジョブは、Agent の実行可能ファイル (`dbnsmp.exe`) の所有者であるユーザーの権限のもとで実行されます。`root.sh` は、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を設定するようユーザーに強制します。

---

---

**注意：** 1 台の UNIX マシン上に複数の Oracle ホームが存在しても構いませんが、1 台の UNIX マシン上で実行できる Intelligent Agent は 1 つのみです。Agent は、専用のポート (1748) 上で Management Server と通信します。

---

---

### 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、`dbnsmp` 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、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 dbnsmp` と入力します。

`dbnsmp` について、すべての関連詳細がリスト表示されます。

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

```
-rwsr-xr-x  1 root      g651      1497980 Jun 12 21:04 dbnsmp
```

`root` は所有者です。`dbnsmp` は 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` で、許可が `-rwsr-xr-x` のようになっていれば、`root.sh` は正しく実行されています。

## UNIX Agent の運用制御

UNIX 上では、Agent の起動と停止に `lsnrctl` コマンドが使用されます。関連する `lsnrctl` 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
UNIX プラットフォーム上で Agent を起動する	<code>lsnrctl dbsnmp_start</code>
UNIX プラットフォーム上で Agent を停止する	<code>lsnrctl dbsnmp_stop</code>
Agent のステータスを検証する	<code>lsnrctl dbsnmp_status</code>

ご使用のプラットフォームの、その他の情報または制限については、Intelligent Agent の `readme` ファイル (`ORACLE_HOME/network/agent/doc/readme.wri`) を参照してください。

## UNIX または Windows NT での SNMP の構成

サードパーティ製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。

SNMP の構成はプラットフォームによって異なります。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。

## 複数のネットワーク・カードと使用するためのリリース 8.1.7 Agent の構成

リリース 8.1.7 の Intelligent Agent より、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 Agent の構成に、3 つのオプション選択できます。デフォルトでは、Agent はそのマシン上の主 NIC（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。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 Agent にリスニングをさせるのが好ましくない場合は、使用しないでください。

---

---

**注意：** Agent は、EM ジョブおよびイベントの実行を求めて入ってくるすべての要求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。

---

---

また、同様のオプションが Agent Data Gatherer にも適用できることに注意してください。

Agent では、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。

Intelligent Agent が稼働しているノードに、それぞれ独自の IP アドレスを持つ複数のネットワーク・カードがある場合、Intelligent Agent では、その構成パラメータに応じて、主ネットワーク・インタフェース、特定の IP アドレス、または複数 IP アドレスの任意のアドレスのいずれかで、入ってくる要求をリスニングし、受信できます。

## 複数のネットワーク・カード使用時の Agent の動作

### 1. リスニング・アドレスが指定されていない場合

デフォルトでは、Agent は Agent マシンの主ネットワーク・インタフェース・カードでの接続をリスニングします。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、Agent は、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。

### 2. 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、Agent はそのアドレスでの接続のみをリスニングします。Agent のリスニング・アドレスを指定するには、既存のパラメータ *dbsnmp.address* および *dbsnmp.spawnaddress* が使用されます。

主ネットワーク・カード以外特定のネットワーク・インタフェース・カードに Agent をバインドするには、次のようにします。

1. *dbsnmp.address* および *dbsnmp.spawnaddress* パラメータを “HOST=< ネットワーク・カードの IP アドレス >” に設定します（詳細は A-5 ページの付録 A を参照してください）。
2. Agent を起動します。

### 3. 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、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。Intelligent Agent のリスニング・アドレスでホスト名を指定するには、既存のパラメータ *dbsnmp.address* および *dbsnmp.spawnaddress* が使用されます。

8.1.7 より前のリリースの Agent (8.1.5 以降) では、これが Agent のデフォルトの動作です (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動作であるためです)。

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には、次のようにします。

1. *dbsnmp.address* および *dbsnmp.spawnaddress* を “HOST=< ホスト名 >” に設定します。
2. Agent を起動します。

#### 4. Windows NT FailSafe 構成が使用されている場合

Windows NT FailSafe 構成では、Agent は FailSafe Agent の 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。

Agent 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。

#### 5. 複数ネットワーク・カードを使用するための Data Gatherer の構成

Data Gatherer の場合、パラメータは *vpp.node\_address* です。このパラメータは、/ORACLE\_HOME/network/admin ディレクトリ、または TNS\_ADMIN 環境変数で指定されたディレクトリにある SQLNET.ORA ファイルで指定してください。

デフォルトでは、Data Gatherer は、主ネットワーク・カードにバインドされます。主カード以外のネットワーク・カードを使用するには、*vpp.node\_address* をそのネットワーク・カードの IP アドレスに設定します。Data Gatherer をすべてのネットワーク・カードにバインドするには、*vpp.node\_address* をホスト名に設定します。

## Oracle Intelligent Agent および 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 8.0.4 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。Intelligent Agent 8.0.4 では、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・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。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。

---

## Agent に必要なロールとユーザー

Intelligent Agent に必要な dbsnmp ユーザー・アカウント（パスワードは dbsnmp）および SNMPAGENT ロールは、catsnmp.sql に含まれています。catsnmp.sql スクリプトは、データベースとともにインストールされます。catsnmp.sql スクリプトは、Oracle データベースのインストール時に catalog.sql によって自動的に実行されます。

セキュリティ上の理由から、Intelligent Agent のデータベース・ログイン用のアカウントとパスワードを、ユーザーが変更しなければならない場合があります。デフォルトのアカウントとパスワードはともに dbsnmp です。ユーザー名とパスワードを dbsnmp/dbsnmp 以外のものに変更するには、catsnmp.sql をオープンして自分のアカウントとパスワード用に編集し、スクリプトを再実行する必要があります。その後、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>
```

データベースに SNMPAGENT ロールが存在するかどうかを判断するには、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、そのデータベース上で 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、Agent がコンタクトする必要のあるすべての付与と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 または INTERNAL 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 Oracle 8.x データベースでは、スクリプトは

ORACLE\_HOME¥rdbms8x¥admin ディレクトリにあります。catsnmp.sql スクリプトを複数回実行することによって問題が生じることはありません。

---

---

## 自動検出

リリース 7.3.3 以降の 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、Agent によって作成される読取り専用ファイルです。このファイルには、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 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、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、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。

---

---

**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「構成ファイル」](#)を参照してください。

---

---

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。

リリース 7.3.4 および 8.0.3 以降では、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 レジストリのどちらかからコンパイルします。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、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、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 Agent が見つけられない場合、<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：** 7.3.3 およびそれ以前のリリースでは、Agent は GLOBAL\_DBNAME を使用しません。

---

---

---

---

**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、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、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 ファイルで 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 を手動で編集しなければならない場合があります。

---

---

## 自動検出のための前提条件

- SQL\*Net バージョン 2 または Net80 TCP/IP 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 Net8）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。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。



- 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) 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、8.0.3/7.3.4 以降のリリースでのみ、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、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。  
 (NT) リリース 8.0.5 以降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取られます。この変数は、次の場所にあります。
  - (NT) 「コントロール パネル」→「システム」→「環境」タブ内の TNS\_ADMIN 変数
  - (NT) NT レジストリ内の TNS\_ADMIN キー

## サービス検出プロセス

Agent が起動時に最初に実行する操作は、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、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。

### Agent の検出プロセス (Windows NT)

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。

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。

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 (Agent により管理されるサービス) を判断します。

1. Agent は、Windows NT レジストリから読み取ら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、Agent は listener.ora ファイルを読み取り、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、SQL\*Net の構成ファイルの場所にに基づきます。たとえば、TNS\_ADMIN 環境変数と、\$ORACLE\_HOME¥network¥admin ディレクトリの場所は、Windows NT レジストリから読み取られた 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、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 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、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、Agent を再起動します。

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、Agent を再起動して、新しいサービスを 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 バージョンの Intelligent Agent についても適用します。

---

## Agent の検出プロセス（UNIX）

起動時に 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。

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（Agent により管理されるサービス）を判断します。

1. Agent は oratab ファイルから、すべての Oracle ホームおよび SID の値を読み取ります。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. oratab ファイルから読み取られた Oracle ホームの値を基に、Agent は listener.ora ファイルを検索して、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。
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、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 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。

## Intelligent Agent 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勧めします。Enterprise Manager 環境に不可欠な部分として、特定のステップに続いて、新しい Agent への移行によりご使用の Enterprise Manager ジョブおよびイベントが影響を受けないことを確認する必要があります。

## Enterprise Manager バージョン 2 の Agent アップグレード・ガイドライン

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。
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「ライブラリへのコピー」を選択します。
3. イベントの警告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、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 Agent をシャットダウンします。
6. 新しい 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 Agent に保存したジョブおよびイベントを再度送ります。

## Enterprise Manager ジョブ/イベントの 8.1.5 または 8.1.6 Agent から 8.1.7 Agent への移行

8.1.5/8.1.6 Intelligent Agent に対して発行された既存のジョブまたはイベントがあり、ジョブの再発行またはイベントの再登録を行わずに、8.1.7 Agent にアップグレードする場合、すべての既存のジョブまたはイベントを、次の手順で移行する必要があります。

1. 8.1.7 Agent を別の ORACLE\_HOME にインストールします。
2. コマンド行で `lsnrctl dbsnmp_stop` と入力し、古い 8.1.x Agent を停止します。
3. 古い Agent の ORACLE\_HOME/network/agent ディレクトリから 8.1.7 Agent の ORACLE\_HOME/network/agent ディレクトリに、すべてのファイル (\*.q ファイル、\*.inp ファイルおよび tcl\* ファイル) をコピーします。ディレクトリをコピーしないでください。この結果、これらのファイルは、8.1.7 Agent をインストールしたユーザーの所有物になります。新しい Agent のディレクトリで、これらのファイルの所有権を確認してください。
4. `lsnrctl dbsnmp_start` と入力して、8.1.7 Agent を起動します。
5. 新しい Agent で検出されたサービス（データベース、リスナーなど）が、古い Agent によって検出されたサービスと同一であることを確認してください。

古い Agent と新しい Agent の ORACLE\_HOME/network/agent/services.ora ファイルにあるサービス名および定義を、ただちに比較チェックすることをお勧めします。

6. Enterprise Manager コンソールから、Agent のノードをリフレッシュします。

移行後、以前のジョブまたはイベントは、すべて以前と同様に実行されます。

## Data Gatherer の構成

Data Gatherer は、Oracle Capacity Planner、および Java ベースの新しい Oracle Performance Manager によって使用されるパフォーマンス・データ収集ツールで、Intelligent Agent とともにインストールされます。ホスト上に Oracle Data Gatherer をインストールしたら、その構成を行う必要があります。

デフォルトでは、Oracle Data Gatherer はデータベースに対する優先接続情報として設定されたユーザー名 / パスワードのアカウントを使用して、Data Gatherer の場所の特定を試みます。優先接続情報が正しくない、または Data Gatherer がホスト上にない場合、Data Gatherer の場所を再度問い合わせるメッセージがクライアントによって表示されます。

クライアント・アプリケーション（Performance Manager および Capacity Planner）を起動する前に、データベースに対する優先接続情報を設定することがあります。

Oracle Data Gatherer がインストールされたホスト上の TNSNAMES.ORA ファイルに、次の要素に対するエントリが含まれている必要があります。

- ターゲット・データベース。
- Oracle Capacity Planner および Oracle Performance Manager のリポジトリ。リポジトリの場所は、Oracle Configuration Assistant を使用して定義します。

---

**注意：** Oracle Capacity Planner または Oracle Performance Manager のどちらでも使用する予定のない場合、Oracle Data Gatherer を構成または起動する必要はありません。

---

## 別の Oracle ホームに新しい Data Gatherer をインストールする時の前のバージョンからのアップグレード

前のバージョンとは異なる Oracle ホームに、Oracle Data Gatherer の新しいバージョンをインストールできます。これを行う場合の手順は、次のとおりです。

1. 前のバージョンの Oracle Data Gatherer を停止します。Oracle Data Gatherer を停止する方法については、2-20 ページの「[NT および UNIX でのデータ収集サービスの運用制御](#)」を参照してください。
2. 新しいバージョンの Oracle Data Gatherer をインストールします。ただし、この時点ではまだ起動はしないでください。
3. 前のバージョンの Oracle Data Gatherer に関連付けられた、Capacity Planner の構成ファイル（状態ファイル）およびデータ・ファイルを、新しいバージョンの Oracle Data Gatherer をインストールした Oracle ホームに移動します。

Oracle Data Gatherer の状態ファイルおよびデータ・ファイルは、`$ORACLE_HOME/odg/reco` ディレクトリにあります。Oracle Capacity Planner を使用して新しいバージョンの Oracle Data Gatherer に接続し、新しい収集を設定する前に、ファイル群を新しい `$ORACLE_HOME/odg/reco` ディレクトリにコピーする必要があります。

これらのファイルを移動しないと、次の問題が起こります。

- a. バイナリ・データがロードされない。  
Oracle Data Gatherer によって作成されたバイナリ・データ・ファイルのうち、Capacity Planner データベースにロードされていないものは、ロードされません。
- b. データ収集定義が保持されていない。

Capacity Planner のデータ収集を再定義する必要があります。

前のバージョンと同じ Oracle ホームに新しいバージョンの Oracle Data Gatherer をインストールした場合、あるいは現在 Oracle Capacity Planner を使用していない場合は、状態ファイルおよびデータ・ファイルを移動しないでください。

4. 新しいバージョンの Oracle Data Gatherer を起動します。Oracle Data Gatherer を起動する方法の詳細は、2-20 ページの「[NT および UNIX でのデータ収集サービスの運用制御](#)」を参照してください。

## Collection Data の Agent Data Gatherer リリース 8.1.7 への移行

ユーティリティ NMUMIGRATE を使用すると、既存の Agent Data Gatherer によって実行中の既存データベース収集を、8.1.7 Agent Data Gatherer で認識される新規フォーマットに移行できます。これは、8.1.7 より前の ODB カートリッジから新規 DBA カートリッジへの移行です。新規の DBA カートリッジにより、パフォーマンスならびに収集が強化されます。

収集データを移行する前に、次のことを確認してください。

- 8.1.7 Intelligent Agent をインストールします。
- ORACLE\_HOME 環境変数が、8.1.7 Intelligent Agent の ORACLE\_HOME に設定されていることを確認します。
- Data Gatherer が実行中でないことを確認します。

コマンド・プロンプトで、次のように入力します。

```
nmumigrate <old ORACLE_HOME>
```

このユーティリティは、8.1.7 ORACLE\_HOME/bin ディレクトリの下にインストールされ、旧 ORACLE\_HOME 内のデータベース収集を、新 ORACLE\_HOME 内の 8.1.7 Agent Data Gatherer で認識される新規のフォーマットに移行します。この新しい ORACLE\_HOME に、新規の収集状態ファイルが作成されます。ユーティリティでは、ORACLE\_HOME 環境変数を使用して、新しい ORACLE\_HOME を取得します。ユーザーが古い ORACLE\_HOME 内で 8.1.7 Agent Data Gatherer にアップグレードしている場合、ユーザーは、このユーティリティ実行時に ORACLE\_HOME を指定する必要はありません。データベース収集カートリッジ以外の収集カートリッジの場合、ユーティリティでは、8.1.7 ORACLE\_HOME にカートリッジがインストールされているかどうかを確認し、インストールされていれば、対応する収集ファイルを旧 ORACLE\_HOME から 8.1.7 ORACLE\_HOME へコピーします。

移行の実行後、8.1.7 ORACLE\_HOME/odg/log/migration.log ファイルで移行の詳細を確認してください。

---

---

**注意：** このユーティリティでは、ユーザー定義スクリプトの収集は移行されず、移行を試みると、次のメッセージが出力されます。

```
Warning: class not found in map
```

この場合、ユーザーは、ユーザー定義スクリプトの収集を、Oracle Capacity Planner に移動して、これらの収集を再起動することにより、手動で移行する必要があります。

---

---

## 移行後

\$ORACLE\_HOME/odg/log/migration.log ファイルで、移行の詳細を確認できます。

## NMUMIGRATE ユーティリティの機能

NMUMIGRATE は、移行時に次のタスクを実行します。

1. Data Gatherer が実行中でないことを確認します。Data Gatherer が実行中の場合、NMUMIGRATE は Data Gatherer が書き込みを行っている状態ファイルにはアクセスできません。
2. 現行の ORACLE\_HOME 設定から \$ORACLE\_HOME を取得します。
3. メッセージ・カタログ・ファイルを読み込みます。

```
$ORACLE_HOME/odg/msg/vpxodbus.msb
```

```
$ORACLE_HOME/odg/msg/vpxdbaus.msb
```

4. 移行ログを記録するためのログ・ファイルを作成または追加します。

```
$ORACLE_HOME/odg/log/migration.log
```

5. 次のディレクトリから移行する必要のある状態ファイルのリストを取得します。

```
$OLD_ORACLE_HOME/odg/reco
```

これらのファイルは、‘S’ または ‘R’ で始まります。

6. これらの状態ファイルの MODULE\_ID を調べて、次のことを確認します。
  - a. ID が ODB かどうか。ODB の場合、DBA カートリッジに移行されます。
  - b. Windows NT オペレーティング・システムのカートリッジ、NT\_OS の場合、現行プラットフォームが Windows 2000 であれば移行されます。
  - c. 他のすべてのカートリッジ状態ファイルは移行されません。
7. NMUMIGRATE では、すでにアクティブな収集が存在する場合、新規の ORACLE\_HOME に状態ファイルがすでに存在するかどうかを確認します。移行は実行されません。
8. ODB カートリッジの場合、NMUMIGRATE はまず状態ファイルが有効であることを確認し、‘M’ ファイルにコピーします。次に、ODB カートリッジの収集設定を DBA カートリッジに変換します。ORACLE\_HOME に新しい状態ファイルを生成します。
9. 他のカートリッジ・ファイルの場合、異なる ORACLE\_HOME にインストールされている場合は、NMUMIGRATE はこのカートリッジがインストールされているかどうかを確認します。インストールされている場合、状態ファイルおよびデータ・ファイルを新規の ORACLE\_HOME にコピーします。

## NT および UNIX でのデータ収集サービスの運用制御

UNIX および Windows NT 上の Oracle Enterprise Manager では、vppcntl コマンドを使用して、データ収集サービスを管理します。vppcntl の実行可能ファイルは、ORACLE\_HOME/bin ディレクトリにあります。

次の表は、Oracle Data Gatherer の制御コマンド一覧です。

操作	入力するコマンド
Oracle Data Gatherer を起動する	vppcntl -start
Oracle Data Gatherer を停止する	vppcntl -stop
Oracle Data Gatherer の動作を確認する	vppcntl -ping
Oracle Data Gatherer のバージョンを調べる	vppcntl -version

**注意：** 1 台のホスト上では、同時に 1 つのバージョンの Data Gatherer のみを実行できます。したがって、1 台のホスト上に複数の Data Gatherer をインストールする必要はありません。前のバージョンの Data Gatherer がある場合は、それを削除することをお薦めします。1 台のホスト上で複数の Data Gatherer を起動しようとすると、エラーが返されます。

複数バージョンの製品が混在する環境で Oracle Enterprise Manager を実行している場合、最新のクライアントにアップグレードすることをお薦めします。詳細は、次の表の互換性情報を参照してください。

Oracle Enterprise Manager のバージョン	Data Gatherer のバージョン
Enterprise Manager 1.5.5 および 1.6.0 クライアント	Data Gatherer リリース 8.0.4 および 8.0.5 リリース 8.1.5 の Data Gatherer とは組み合わせないこと
Enterprise Manager 2.0.4 クライアント	Data Gatherer リリース 8.0.4、8.0.5 または 8.1.5



## Windows NT 上でのその他の運用制御方法

この項では、Windows NT を通しての Oracle Data Gatherer の制御に関する情報を説明しています。

デフォルトでは、1 台のホスト上で Oracle Data Gatherer を手動で起動します。Windows NT の「コントロール パネル」を通して Oracle Data Gatherer を自動的に起動するための手順は、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. 起動する Oracle Data Gatherer のサービス名を選択します。

ホスト上の Oracle ホームが 1 つの場合、Oracle Data Gatherer サービスの名前は OracleDataGatherer です。

ホスト上に複数の Oracle ホームがあり、Oracle Data Gatherer が複数の Oracle ホームにインストールされている場合、複数のデータ収集サービスが表示されます。Oracle Data Gatherer が複数の Oracle ホームにインストールされているとき、データ収集サービスの名前には、Oracle<Oracle\_Home\_name>DataGatherer のような命名規約が使用されます。たとえば、ホスト上に 804、805 という名前で 2 つの Oracle ホームが存在し、データ収集サービスがその両方にインストールされているとします。これらの Oracle ホームに対する Oracle のデータ収集サービスには、それぞれ Oracle804DataGatherer、Oracle805DataGatherer という名前が付けられます。

スタートアップの種類は「手動」に設定されています。これは、ユーザーが手動でデータ収集サービスを起動するための設定です。システムを起動するたびにデータ収集サービス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. 「起動」ボタンをクリックして、データ収集サービスを起動します。

Windows NT の「コントロール パネル」を通して Oracle Data Gatherer を停止するための手順は、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. 停止する Oracle Data Gatherer のサービス名を選択します。
3. 「停止」ボタンをクリックして、データ収集サービスを停止します。

## データ収集サービスの動作確認

Windows NT 上では、次のいくつかの事項を確認することによって、Oracle Data Gatherer が動作しているかどうかを判断できます。

- `vppcntl -ping` コマンドによって返されるステータス
- 「コントロール パネル」の「サービス」の「状態」
- 「Windows NT タスク マネージャ」での `vppdc` プロセスの状態

オラクル社では、`vppcntl -ping` コマンドの使用をお勧めします。これは、このコマンドが Oracle Data Gatherer の動作状況を報告するのみでなく、サービスの応答性および正常な動作を判断するためのテストを実行するためです。

データ収集サービスの警告ログは、`ORACLE_HOME%\odg\bin\alert_dg.log` ファイルに書き込まれます。

UNIX では、`vppcntl -ping` コマンドを使用して、Oracle Data Gatherer が動作していることを確認します。データ収集サービスの警告ログは、`$ORACLE_HOME%\odg\bin\alert_dg.log` ファイルに書き込まれます。

## Oracle Data Gatherer 上でのトレース情報の取得

Oracle Data Gatherer 上で、トレース情報を取得することもできます。トレース情報を取得するには、Oracle Data Gatherer を Oracle Enterprise Manager コンソールから実行する必要があります。

Oracle Data Gatherer のトレース情報を画面上に表示するには、DOS プロンプトまたは UNIX コマンド・ラインで、次のコマンドを入力します。

```
vppdc -console -debug
```

Oracle Data Gatherer のトレース情報をファイルに送るには、DOS プロンプトまたは UNIX コマンド・ラインで、次のコマンドを入力します。

```
vppdc -console -debug > trace.txt
```

前述のコマンドが使用される場合、トレース・ファイルには `trace.txt` という名前が付けられます。トレース・ファイルには別の名前を指定することもできます。

---

---

**注意：**すでに動作している Data Gatherer をデバッグ・モードで起動し直す場合、Data Gatherer をいったん停止し、前述のように `-debug` フラグを使用してコマンド・ラインから Data Gatherer を起動する必要があります。

---

---

---

# 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
- 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
- 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

- NLS 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取る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も生成できます。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.neosoft.com/tcl>
- <http://www.scriptics.com/resource/doc/papers/>

---

---

**注意：** World Wide 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

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、3-11 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。

```
#!/usr/local/bin/Tcl -f
```

```
#
# 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 $cur1]
while {$oramsg(rc) == 0}{
    oraplexec $cur2 $monthly_pay_pl :monthly "" :jobclass "$job"
    set total_for_job [lindex [orafetch $cur2] 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 `oramsg` を作成、維持して、Oracle Server メッセージのフィードバックを提供します。`oramsg` 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 を除くすべてのケースで、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。`oramsg` 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：** `oramsg` 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。

---

---

### `oramsg` の要素

`oramsg` の要素を次に示します。

#### **`oramsg (agent_characterstet)`**

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。3-13 ページの「[convertin](#)」および 3-13 ページの「[convertout](#)」を参照してください。

#### **`oramsg (db_characterstet)`**

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。3-13 ページの「[convertin](#)」および 3-13 ページの「[convertout](#)」を参照してください。

#### **`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 (colscale)`**

`oracols` によって返される数値列のスケールの Tcl リストです。`colscale` 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 Td 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、次のようなものがあります。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。

### **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 OraTd 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、OraTd 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。

### **oramsg (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。oramsg (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。

### **oramsg (ocifunc)**

OraTd によって最後にコールされた OCI 関数の数値 OCI コードです。詳細は、『Oracle コール・インタフェース・プログラマーズ・ガイド』を参照してください。

### **oramsg (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。

### **oramsg (orahome)**

ORACLE\_HOME ディレクトリです。



**oramsg (oraindex)**

構成ファイル `snmp.ora` から読み取られた SNMP 索引値の Tcl リストです。

**oramsg (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。

**oramsg (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、『Oracle エラー・メッセージ』を参照してください。一般的な値のリストを [表 3-1「エラー・メッセージ」](#) に示します。

**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	<code>orafetch</code> コマンドの実行時にデータの終端に到達。
1406	<code>orafetch</code> 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、 <code>maxlong</code> 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 (sqlfunc)**

最後に実行された SQL 関数の数値 OCI コードです。詳細は、『Oracle コール・インタフェース・プログラマーズ・ガイド』を参照してください。

**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す。

### `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（警戒）、2（警告）があります。

---

---

**警告：** `trigevent` は、修正ジョブでしか機能しません。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
```

この例の修正ジョブは、特定レベルの CPU の使用率を監視する CPUUTIL イベント・テストと対応づけられています。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。

- 警告のしきい値 = 20
- 警戒のしきい値 = 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： オペレーティング・システム・コマンド `top -dl -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、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 または OS/2 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 SQL\*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 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 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・スクリプトに限り、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、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、特殊なりモート・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、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 Server 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 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、oralogon は次のようなエラーを返します。“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。”これは、不正な接続文字列が与えられた場合のメッセージです。ただし、エラー・メッセージは次のような Tcl 固有のメッセージになることもあります。“ERROR: Tcl-00456: division by zero error”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、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 ‘|’ 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

<i>oraaudocom</i>	<i>oracancel</i>	<i>oraclose</i>	<i>oracols</i>	<i>oracommitt</i>
<i>orafetch</i>	<i>oralogoff</i>	<i>oralogon</i>	<i>oraopen</i>	<i>oraplexec</i>
<i>orareadlong</i>	<i>oraroll</i>	<i>orasql</i>	<i>orawritelong</i>	

### ■ RDBMS 管理用関数

<i>orastart</i>	<i>orastop</i>
-----------------	----------------

- 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*      *orertime*

## 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、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

**目的** 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。データベース固有のジョブまたはイベントに対しては、`$oramsg(db_characteraset)` を使用します。ノード固有のジョブまたはイベントに対しては、`$oramsg(agent_characteraset)` を使用します。3-5 ページの「[oramsg 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 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限り、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

**目的** 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。データベース固有のジョブまたはイベントに対しては、`$oramsg(db_characteraset)` を使用します。ノード固有のジョブまたはイベントに対しては、`$oramsg(agent_characteraset)` を使用します。3-5 ページの「[oramsg 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 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限り、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

**目的** 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 `rdbsms` 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

**目的** 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 `rdbsms` 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

**目的** 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ハンドルを使用する、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

3-12 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

**目的** 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

3-12 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

**目的** 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

3-12 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

**目的** この関数は、最後に実行された orasql、orafetch または oraplexec 関数が返した列の名前を Tcl リストで返します。oracols は、境界変数名が返されるようなケースで oraplexec の後に使用できます。

**構文** oracols logon-handle

## パラメータ

### logon-handle

3-12 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 colprec 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 colscale 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）。

## oracommit

**目的** 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** oracommit logon-handle

## パラメータ

### logon-handle

3-12 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。

## oradbsnmp

**目的** この関数は、SNMP MIB 値を取り出します。

**構文** oradbsnmp get | getnext object\_Id

**パラメータ** object\_Id

object\_Id のとりうる値は、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** oradbsnmp は、RDBMS パブリック MIB、または Oracle RDBMS プライベート MIB など、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 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、3-24 ページの「[orasnmp](#)」の項を参照してください。SQL コマンドで V\$ 表から値をフェッチするかわりに、oradbsnmp の使用を薦める理由には、次のようなものがあります。

- 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 のケースでは、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 orasnmp を使用してください。

## orafail

**目的** 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 のとりうる値は、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

**目的** 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

3-12 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よりも大きい場合、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継続します。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。ロー・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` 配列の索引 `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 配列の索引 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 配列の索引 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 Agent の子アドレス、つまり 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。

**追加情報：** Intelligent Agent のアドレスの詳細は、『Oracle Server のインストール・ガイド』の Agent の構成に関する章を参照してください。

## orainfo

**目的** 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

3-12 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 Agent から、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 Agent から構成情報がフェッチされます。Agent の構成は、次の要素からなる Tcl リストです。

- この 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。
- Agent の通常の RPC アドレス。tnsnames (TNS) 文字列で表されます。
- Agent のファイル転送アドレス。TNS 文字列で表されます。

## orajobstat

**目的** この関数はジョブによって使用され、即時性の出力をコンソールに返します。

**構文** orajobstat destaddress string

### パラメータ

#### destaddress

3-12 ページの「[共通パラメータ](#)」を参照してください。

### string

string のとりうる値は、引用符で囲まれたテキスト文字列、または FAC-XXXXXX のような形式の文字列のどちらかです。XXXXXX 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 Agent のアドレスです。この関数は、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。Agent のアドレスは orainfo を使用して取得できます。

## oralogoff

**目的** logon-handle に関連付けられた Oracle Server 接続からログオフします。

**構文** oralogoff logon-handle

### パラメータ

#### logon-handle

3-12 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

**目的** connect\_string を使用して Oracle Server に接続します。

**構文** oralogon connect\_string

### パラメータ

#### connect\_string

3-12 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。合計 6 本の接続を上限として、同じまたは異なるサーバーへの複数の接続が許容されています。

**追加情報：** 接続に関する制限は、オペレーティング・システム固有の注記において説明されています。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使用できます。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。

## oraopen

**目的** 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

3-12 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

**目的** この関数は、無名 PL 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

3-12 ページの「[共通パラメータ](#)」を参照してください。

#### pl\_block

pl\_block のとりうる値は、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。

oramsq 配列の索引 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

**目的** この関数は、LONG または LONG RAW 列の内容を読み取り、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**

3-12 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 列から読み取られたバイト数が 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

**目的** この関数はジョブによって使用され、不要なイベントを 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¥admin 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。orareporevent では、値は 1 (警戒)、2 (警告)、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 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

**目的** 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

3-12 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

**目的** 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

**目的** 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** orasnm get | getnext object\_Id

### パラメータ object\_Id

object\_Id のとりうる値は、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.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。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。

```

"rdbsDbPrivateMibOID",    "1.3.6.1.2.1.39.1.1.1.2",
"rdbsDbVendorName",      "1.3.6.1.2.1.39.1.1.1.3",
"rdbsDbName",            "1.3.6.1.2.1.39.1.1.1.4",
"rdbsDbContact",         "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取り、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、オブジェクトを object\_Id に解決します。

Oracle サービスのため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、oraindex グローバル変数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。

orasnm 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。orasnm get のケースでは、object\_Id がオブジェクトと同じである一方で、getnext に関しては、object\_Id は次の論理 object\_Id になります。orasnm 操作は、ローカル・ホストのみに適用されることが想定されます。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、3-17 ページの「[oradbsnm](#)」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

**目的** 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

3-12 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 配列の rc 索引は、リターン・コードに設定されます。rows 索引は、挿入、更新または削除のケースで SQL 文の影響を受け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、『Oracle 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

**目的** 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

3-12 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

**目的** 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**

3-12 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、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 Agent に特殊な RPC を送り、それにより Agent がデータベースから切断されることが原因です。

---

---

## oratime

**目的** この関数は、現在の日付と時刻を返します。

**構文** `oratime`

**パラメータ** なし

**コメント** なし

## orawritelong

**目的** 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

3-12 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。

---

## 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 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](#)

### 構成ファイル

#### snmp\_ro.ora

snmp\_ro.ora ファイルは、\$ORACLE\_HOME/network/admin ディレクトリにあります。この読取り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、\$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が、修正は慎重に行ってください。

#### services.ora

services.ora ファイルは Agent の起動時に作成され、Windows NT プラットフォームでは \$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、Agent が動作するノード上のサービスの一覧が含まれます。このファイルは Oracle Enterprise Manager ナビゲータの「検出」メニュー・オプションを通して 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 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。

### **SNMP.INDEX.service\_name.world = index\_number**

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なければならない点です。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限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限り使用してください。



**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.POLLTIME = nn**

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。

Agent は 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.IPCTIME = nn**

Agent の Work プロセスがその Comm プロセスに ping を実行して、ダウンしていないことを確認する時間間隔（秒単位）です。Comm がこの時間内に Work からの ping に応答できない場合、Work は古い Comm プロセスを停止して新しいプロセスを起動します。デフォルトは 30 秒です。

誤ったノードの動作・停止イベント（短時間の間隔で引き起こされクリアされるイベント）が定期的にコンソールに報告される場合は、このパラメータの値を増加する必要があります。どの程度の頻度で Comm が再起動するかをチェックするには、次のコマンドを実行します。

```
ps -ef | grep dbsnmp
```

また、子プロセスの開始時刻に注意します。

このパラメータは、Windows NT 上の Agent とは無関係です。

**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（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 Parallel Server 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。Parallel Server 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。

---

---

**NMI.TRACE\_LEVEL = OFF | USER | ADMIN | nn**

指定されたレベルで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、これはバグ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。

**NMI.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、nmi.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 \$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。

**NMI.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。

**NMI.TRACE\_FILECNT = *nn***

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のが好ましいが、Agent マシン上のディスク領域が限られている場合、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。

**NMI.TRACE\_FILESIZE = *nn***

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のが好ましいが、Agent マシン上のディスク領域が限られている場合、このパラメータを NMI.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。

**NMI.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。

**NMI.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。

---

**注意：** 次のアドレスは、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、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>)))**

Agent が、送られてくる要求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、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。

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。

Agent は PORT=1748 を必要とします。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、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。

リリース 7.3.3 以前の Agent では、このアドレスが、Oracle Enterprise Manager コンソールが存在するマシン上の tnsnames.ora ファイル中の Agent に対するエントリと一致している必要があります。

**dbsnmp.spawnaddress =(ADDRESS= (PROTOCOL=<protocol>)  
(HOST=<host\_name>)(PORT=<spnport\_no>))**

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われる spnport\_no は、DBSNMP.ADDRESS パラメータで使われる port\_no とは異なります。

Agent は PORT=1754 を要求します。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、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。



---

# 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](#)
- [質問と回答](#)
- [Intelligent Agent の起動に関する問題および解決策](#)
- [Intelligent Agent のエラー・メッセージと解決策](#)
- [Agent のトレース](#)
- [TCL のトレース](#)

## 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、Agent が管理側のホストおよび管理対象のサービス  
と通信する必要があります。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、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：** 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 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 Agent で発生した問題が解決します。

---

## 簡単なチェック

次のチェックリストでは、Agent の運用に最も影響する領域をカバーしています。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、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しく説明します。

### Windows NT Agent の簡単なチェック

Windows NT システム上で 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、Agent サービスが開始されていることを確認します。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. 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 NT 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 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「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. 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。

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。
5. TCP/IP がインストールされていること確認します。TCP/IP は必要条件です。
  6. Agent が起動しない理由がこの時点でわからない場合、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。  
`nmi.trace_level=admin` (16 に設定すると最も多くの情報を得られます)  
`nmi.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>`  
`nmi.trace_file=<トレース出力ファイルの名前>`
  - b. 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。  
`NMI.LOG` は、Agent の一般的な問題を示します。  
`NMICONFIG.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. デーモンのトレースを有効にします。
- a. `$ORACLE_HOME¥network¥admin¥sqlnet.ora` ファイルを開き、  
`daemon.trace_level=13` と  
`daemon.trace_directory=e:¥orant¥net80¥trace` の 2 行を追加します。
  - b. コンソールを閉じて、デーモンを停止します。
  - c. コンソールを開いて、トレース・モードでデーモンを再起動します。
  - d. ジョブを送り、`daemon.trc` ファイルを参照して、デーモンおよびコンソールの問題を確認します。

## UNIX Agent の簡単なチェック

UNIX システム上で 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. Agent リスナーが機能していることを確認します。次のコマンドを入力します。

```
lsnrctl dbstatus
```

Agent が動作している場合、次のように表示されます。

```
LSNRCTL for Solaris: Version 8.1.5.0.0 - Production on 04-NOV-98 18:44:15
```

```
(c) Copyright 1997 Oracle Corporation. All rights reserved.
```

```
The db subagent is already running.
```

2. 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。
3. Oracle ユーザーが、ORACLE\_HOME/NETWORK/AGENT および ORACLE\_HOME/AGENT/LOG への書き込み許可を持っていることを確認します。
4. 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。

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「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
5. 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、Agent をトレースします。
    - nmi.trace\_level=admin (16 に設定するとより多くの情報を得られます)
    - nmi.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>
    - nmi.trace\_file=agent
  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. (dbsnmp アカウントではなく) INTERNAL または 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、dbsnmp アカウントは作成されません。
    - 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 は読取り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、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 アカウントで backupts.sql を実行する必要があります。



---

---

**警告：** 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。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 Agent に実行させることができます。

---

---

## 質問と回答

トラブルシューティング・チェックリストで 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、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 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、より詳しく説明しています。この項では、次のような問題について説明します。

- B-6 ページの「TCP/IP の構成と動作の確認」
- B-7 ページの「DNS 名およびコンピュータ名の一致の確認 (Windows NT)」
- B-8 ページの「Net8 構成ファイルの確認」
- B-9 ページの「Net8 の動作の確認」
- B-10 ページの「Agent の正常起動の確認」
- B-12 ページの「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-12 ページの「Agent が正しい許可で動作していることの確認 (UNIX)」
- B-12 ページの「OS 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-13 ページの「リリース 7.3.3 またはそれ以前の Agent を使用しているかどうかの確認」
- B-13 ページの「ジョブが実行されたにもかかわらず、Agent が状態通知を Enterprise Manager コンソールに返信しない理由」

---

---

**注意：** 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

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、コマンド行で次のように入力します。

- ホスト・マシン (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。コマンド行で、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、コンソール (バージョン 2 の場合は、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. 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. コンソールを実行しているマシン上で、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。
- コンソール・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、Agent を実行しているマシンがコンソール・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>  
コンソール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 <コンソール・マシンの IP アドレス>  
Agent が動作しているマシン上で、コンソール・マシンへの 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、ファイルをコピーするかして、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。

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)

(カッコ内の情報は、システムの実際のホスト情報に置き換えてください。)

HOSTS file:

```
<122.111.111.111>    <hostname>
```

LMHOSTS file:

```
<122.111.111.111>    <netbios name or hostname>  #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 ファイルには、現行のジョブとイベントに関する情報が含まれています。これらのファイルを削除する前に、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。

---

---

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. 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 以前の 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」

## Net8 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できる必要があります。Net8 は、クライアント・マシンと Oracle Server の間、あるいは複数の Oracle Server の間に、セッションおよびデータ通信の媒体を提供します。この理由から、Agent が通信を行うためには、Net8 が正しく構成されていることが前提条件となります。この項では、Agent の通信が失敗するとき起こりうる最も一般的な問題について説明します。

Net8 の構成ファイルは、`$ORACLE_HOME¥network¥admin`、または `$TNS_ADMIN` (Windows NT) か `$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「構成ファイル」](#)を参照してください。

### 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 以降の 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 以降では、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取られます。

Agent では、`listener.ora` ファイル中の TNS 別名情報も使用されます。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 サーバーは一時的に使用不可になる可能性がある一方で、Agent は常に名前を解決できなければならない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、それぞれのインスタンスに対する次の要素が含まれていることを確認します。

- 2つのIPC エントリ
- 1つのTCP エントリ

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。(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 Agent を区別できません。)

Agent による別名の変換には、IPC エントリと、TNS のサーバー上およびコンソール上での別名定義が必要になります。この正しい IPC エントリと TNS 別名定義は、Agent とコンソールの間 (バージョン 1)、または Agent と Management Server の間 (バージョン 2) で正確な通信を行うために必須です。

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。

---

---

**注意：** 2 枚の NIC カードが装着された Windows NT システム上でリリース 7.3.3 の Oracle Intelligent Agent を実行しているとき、一方の NIC カードの IP アドレスのみを記述した tnsnames.ora の中で、1 つのサービス記述子のみを作成するようにしてください。各 NIC カードに対して個別のサービス記述子を作成しないでください。また、1 つのサービス記述子に対する address\_list 部に両方の IP アドレスを記述しないでください。

---

---

## Net8 の動作の確認

Net8 の構成が正しいにもかかわらず依然として Agent と通信できない場合、次のステップとして、Net8 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、Net8 サービス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、Net8 が正常に動作していることを表しています。

その後、コンソールの `tnsnames.ora` ファイルに次の別名（Agent のデバッグ・エントリ）を追加します。

```
agent_<sid>.world=
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS =
        (COMMUNITY =TCP.world)
        (PROTOCOL = TCP)
        (Host = <your-agent-hostname>)
        (Port = 1748)
      )
    )
  )
```

その後、OEM コンソールから次のように入力して、Agent への ping を実行します。

```
tnsping agent_<sid>
```

または

```
tnsping80 agent_<sid>
```

TNSPING コマンドが機能しない場合、前述の別名を Agent マシンの `tnsnames.ora` に追加し、Agent が動作しているマシンから TNSPING の実行を試みます。すべての Agent は、この別名を使用して TNSPING を実行できる必要があります。

## Agent の正常起動の確認

動作中の Agent プロセスの種類を確認します。

### UNIX Agent

コマンド・プロンプトから、次のように入力します。

```
lsnrctl dbsnmp_status
```

ステータスが次のように返されます。

```
The db subagent is already running
```

### Windows NT Agent

1. 「スタート」メニューから、「設定」→「コントロール パネル」を選択します。
2. 「サービス」をダブルクリックします。
3. *OracleAgent* サービスが開始されていることを確認します。

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 Agent が起動しない場合のトラブルシューティング**

UNIX	Windows NT
<p>ファイル \$ORACLE_HOME/network/log/dbsnmp*.log を参照して、エラーを確認する。</p> <p>ファイル \$ORACLE_HOME/network/log/nmiconf.log を参照して、エラーを確認する。</p> <p>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する。 \$ORACLE_HOME/agent/log \$ORACLE_HOME/network/agent</p> <p>snmp_ro.ora、snmp_rw.ora および services.ora の各ファイルを参照して、Agent によって作成されたエントリを確認する。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。</p> <p>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する。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。</p> <ul style="list-style-type: none"> <li>■ listener.ora</li> <li>■ tnsnames.ora</li> <li>■ oratab</li> </ul> <p>TCP/IP がインストールされていることを確認する。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。</p>	<p>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する。「イベント ビューア」には、起動に関連するすべての問題が NT Agent によって書き込まれます。</p> <p>ファイル \$ORACLE_HOME¥network¥log¥nmiconf.log を参照して、エラーを確認する。</p> <p>Agent サービスのプロパティで、Agent によって使用される OS アカウント (デフォルトでは 'System') を確認する。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する。 \$ORACLE_HOME¥agent¥log \$ORACLE_HOME¥Net8¥agent</p> <p>snmp_ro.ora、snmp_rw.ora および services.ora の各ファイルが、起動時に Agent によって作成されていることを確認する。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。</p> <p>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する。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。</p> <ul style="list-style-type: none"> <li>■ listener.ora</li> <li>■ tnsnames.ora</li> </ul> <p>TCP/IP がインストールされていることを確認する。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。</p>

表 B-1 Agent が起動しない場合のトラブルシューティング

UNIX	Windows NT
Agent が起動しない理由がこの時点でわからない場合、トレースを有効にする。 <a href="#">「Agent のトレース」</a> 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する。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"><li>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。</li><li>2. 再起動して、システム・パスの設定を解除します。</li></ol>

Agent が起動しない理由がこの時点でわからない場合、トレースを有効にする。Agent のトレース設定の詳細は、B-28 ページの[「Agent のトレース」](#)を参照してください。

## 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

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、Agent が動作するノード上で実行する必要があります。

**注意：** リリース 7.3.3 およびそれ以前の Agent は、ローカル・データベースへの 2 つの永続的接続を維持します。リリース 7.3.3 よりも新しい Agent では、1 つの永続的接続のみが維持されます。

## Agent が正しい許可で動作していることの確認（UNIX）

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、2-5 ページの[「UNIX 上での Oracle Intelligent Agent のインストール」](#)を参照してください。

## OS 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 OS 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 または \$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取りおよび書込み許可



- \$TEMP ディレクトリに対する読取りおよび書込み許可 (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取りおよび書込み許可を OS ユーザーが持っている必要があります。OS 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。
- 前述のユーザーが、\$ORACLE\_HOME/network/agent ディレクトリへの書込み許可を持っている必要があります。

## リリース 7.3.3 またはそれ以前の Agent を使用しているかどうかの確認

Oracle Enterprise Manager のジョブ・システムとイベント・システムを正しく運用するためには、リリース 7.3.4 またはそれ以降の Intelligent Agent を実行する必要があります。リリース 7.3.3 またはそれ以前の Agent を実行している場合、ジョブ・システムおよびイベント・システムの機能のうち、使用可能なものが制限されます。

---

---

**重要：** リリース 7.3.3 またはそれ以前の Agent を、リリース 7.3.4 またはそれ以降にアップグレードすることをお勧めします。

---

---

## 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。

## ジョブが実行されたにもかかわらず、Agent が状態通知を Enterprise Manager コンソールに返信しない理由

ほとんどの場合、ジョブは実際に実行されていますが、Agent がコンソールに通信できないために通知を返すことができません。ホスト名が解決できることを確認します。コンソールが動作している Windows NT マシンの IP アドレスおよびホスト名が、UNIX マシンの /etc/hosts ファイルに記述されており、DNS/NIS を経由してホスト名が解決できることを確認します。ジョブを再試行します。

TCP/IP の解決をテストするには、コマンド・プロンプトから次のテストを実行します。

```
ping <hostname>
ping <IPaddress>
```

サーバー上で telnet または ftp のサービスが動作していることを確認します (UNIX)。

```
telnet <hostname>
```

```
ftp <hostname>
```

PING では TCP ではなく IP が使用されるため、これは問題がパケットのルーティングにあるかどうかを判断するための良い方法です。

問題が実際に TCP に関係しているかどうかを判断するには、telnet ユーティリティまたは ftp ユーティリティを使用します。

それとは別に、次の手順を実行することもできます。

1. OEM コンソール・マシンの名前および IP アドレスが、Sun サーバーの /etc/hosts ファイルに記述されていることを確認します。記述がない場合、Agent はコンソール・マシンの名前を IPADDRESS に解決できないため、コンソールにメッセージを返すことができません。
2. Oracle Enterprise Manager の Daemon Manager を起動し、「構成パラメータ」をクリックして、「リスニング・アドレス」パラメータにコンソール・マシンの完全な IP アドレスを指定します。以後、Agent からコンソール・マシンへの通信には、この指定された IP アドレスが使用されます。

デフォルトのリスニング・アドレス（TNS 形式）は、次のとおりです。

```
LISTENING ADDRESS = (ADDRESS= (PROTOCOL= TCP) (Host=machine_name) (Port=7770))
```

ジョブがスケジューリングされた状態のままになっている場合、[Del] キーを使用してそのジョブの削除を繰り返します。ジョブを再起動します。一般に、ジョブが起動するまでの間には最大 1 分程度の遅延があるため、複数回の送信を経てはじめてジョブが起動されることがあります。特に、古い Agent（リリース 7.3.2）において、Agent が最初に OEM コンソールとの同期を試みるときに、そのような傾向があります。

## Intelligent Agent の起動に関する問題および解決策

これ以降の項では、Intelligent Agent の起動時に発生する個別の問題、状況およびエラーについて説明します。

### Agent 全般

この項では、UNIX および Windows NT の各バージョンの Intelligent Agent に共通する問題について説明します。問題には、発生の確率によって優先順位が付けられています。

**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 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 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 以降）、あるいは新規 NT ユーザーに割り当てることができます。第 2 章「インストールと構成」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。『Oracle Enterprise Manager 構成ガイド』の「作業環境の設定」を参照してください。
- 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。

**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 以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 およびそれ以降では、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、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 以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。

**Agent 起動時の‘SID のリスナーが見つかりません’** Intelligent Agent を起動しようとする際に、エラー「SID のリスナーが見つかりません」が発生します。表示される SID は常に、Oratab ファイル中の最後の SID です。listener.ora および tnsnames.ora の各ファイルには、その SID の有効な TNS 記述子が記されています。oratab ファイルには無効な SID は含まれず、すべての SID は dbsnmp アカウントを持ちます。この問題は、リリース 7.3.4 およびそれ以降の Agent で修正されています。

リリース 7.3.3 の nmiconf.tcl スクリプトは、listener.ora ファイルを解析して、大文字の文字列 ADDRESS、SID\_LIST\_、SID\_DESCRIPTION および SID\_NAME を探します。検出を有効にするには、これらのパラメータを大文字に修正します。

**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 によって表示されます。\$ORACLE\_HOME 環境変数が Agent の ORACLE\_HOME に設定されていることを確認し、Agent を再起動してください。

**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ある場合、listener.ora 中に、各インスタンスに対する一意の GLOBAL\_DBNAME の記述があることを確認する必要があります。listener.ora 中に、このパラメータを手動で定義しなければならない場合もあります。

**snmp\_ro.ora および snmp\_rw.ora が生成されていない** このエラーは、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/og/nmicnfl.log を参照して、エラーを調べます。Agent の起動に関する問題の詳細は、B-10 ページの「[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。

Agent がマシン上で発見するすべてのサービスは、関連する SQL\*Net/Net8 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、Agent は停止するかエラーを返します。

**Windows NT:** リリース 8.0.4 以降の 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。Agent が取り扱うすべての SID は、関連する SQL\*Net/Net8 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、またはそれ以前の 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、\$ORATAB 環境変数を使用して、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/Net8 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-8 ページの「[Net8 構成ファイルの確認](#)」および B-9 ページの「[Net8 の動作の確認](#)」を参照してください。

**Agent が正しく起動しない** 以前には起動していた Agent が正常に起動しなくなったような場合、環境になんらかの変更があった可能性があります。通常は、Agent にすべてのサービスを再検出させることが良い方法です。

snmp\_ro.ora、snmp\_rw.ora および services.ora の各ファイルを削除して、Agent を再起動します。これで問題が解決しない場合、それらのファイルを削除し、さらに \$ORACLE\_HOME/network/agent ディレクトリ内の \*.q ファイルをすべて削除します。

---

---

**警告：** この操作は、すべてのジョブおよびイベントを削除します。そのことを確認の上、まずコンソールからジョブおよびイベントを削除するようにしてください。

---

---

**複数のリスナー** Intelligent Agent は、1 台のマシン上の単一データベースに対する複数のリスナーをサポートしていません。services.ora には、検出情報を Agent からデーモンに送るときに使用される情報が記述されています。services.ora では、単一のデータベースに対する複数のリスナーはサポートされていません。サービスの検出における制限のため、マシン上で監視する 1 データベースごとに 1 つのリスナーのみを配置します。同一のデータベースを 2 つのリスナーがリスニングしている場合、Agent はエラーを返すか、検出を拒否します。

Agent リリース 8.0.5 以降の場合、複数のリスナーが 1 つのノードに存在すると、Agent が最初に見つけたリスナーしか使用されません。

**DBSNMP ERROR: 複数のリスナーの検出** 動作しているリスナーが 1 つのみであっても、それ以外のリスナーが存在しないとは限りません。Agent は動作しているリスナーのみでなく、(listener.ora ファイルを通して) 構成されるすべてのリスナーを検出します。また、Agent による検索の範囲は、\$TNS\_ADMIN が指すディレクトリに限定されません。listener.ora ファイルが存在する可能性のあるすべての場所が調べられます。

---

**注意：** UNIX マシンで、複数のリスナーが同じ SID (oratab ファイルに記述される) を持つように構成されている場合、Agent は最初に検出したリスナーのみを選びますが、同時に、同じ SID のリスナーが複数存在することをユーザーに知らせるためのメッセージも表示します。

---

ユーザーの意図と異なるリスナーが Agent によって選ばれた場合、問題を修正して Agent を再起動してください。

**ジョブまたはイベント登録時の‘サービス名が無効です’または‘ファイル操作エラー’の表示** これらのエラーは通常、コンソール上のサービスと、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 PRODDDB に変更したような場合、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。

---

NT Agent を使用している場合は、NT 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。

**Agent による過度のメモリー消費** リリース 7.3.4、8.0.3.1.1 (NT) および 8.0.3 (Solaris) 以前の Agent では、メモリー・リークにより、メモリー使用量が次第に増大するという現象が起きます。メモリー・リークは、Agent が通信できない（データベースがダウンしている、リスナーが起動していないなど）別名やサービスが指定されている場合に発生します。Agent がそのようなサービスへの通信を試みるたびに、その要求に関連付けられたメモリーは解放されずに残ります。また、dbprobe および listnerupdownm の各イベントに伴うハンドルも失われます。

Agent をアップグレードしてください。アップグレードできない場合は、動作中のサービスのみを監視するようにします。また、Agent のメモリー使用量を検証します。メモリー使用量が多くなりすぎた場合、Agent を停止し、再起動します。

---

---

**警告：** Agent に付属の TCL スクリプトを修正しないでください。

Intelligent Agent にあらかじめ用意された Tcl (Tool Command Language) スクリプトは、Oracle Enterprise Manager のジョブ・システムおよびイベント・システムで使用されます。Agent で定義済み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 Agent に実行させることができます。

---

---

**Windows NT Enterprise Edition 4.0 上で DBSNMP.EXE の CPU 使用率が 100% になる** この問題は、NT サービスの OracleServiceORCL が動作していないか (ORCL 以外の SID で独自のデータベースが作成されたことが原因で) 存在せず、Oracle Performance Utility (Oracle Enterprise Manager Performance Pack ではなく Oracle8 のパフォーマンス・ユーティリティ) がインストールされているときに起こる可能性があります。

Oracle Performance Utility はインストールの間に、SID “ORCL” を参照する BEQ 接続記述子をハードコードします。したがって、SID が ORCL であるデータベースがマシン上にない場合、前述の問題が発生します。接続記述子の場所は (HKEY\_LOCAL\_MACHINE¥System¥CurrentControlSet¥Services¥Oracle80¥Performance) です。Username、Password および Hostname の値を確認します。

解決策は、Oracle8 Performance Utility を削除することです。

**‘Transport read error’ または ‘Transport write error’ メッセージ** これらのメッセージは、TCP/IP レイヤーでの問題を示すものです。考えられる最大の原因は、IP アドレスとホスト名が同じ物理マシンを参照していないことです。

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。)

‘Oralogin failed in orlon’ Software Developer’s Kit を通して、oratl 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。

**リリース 8.x Agent の動作時に Oracle8 データベースが 1 分おきに REDO ログを切り替える** 非常に短い間隔 (30 秒) にスケジューリングされているジョブまたはイベントがある場合、このことがシステムのアクティビティの原因となります。たとえば、イベント USER BLOCKS が定義されている場合、Agent は一時表を構築することによって同時 / 待機中のロックをチェックし、表を削除し、チェックごとに表を再構築します。

**解決策：** 間隔をより短くするか、基礎となっている表のロギングを無効にします。

## NT Agent

この項では、Windows NT バージョンの Intelligent Agent で発生する可能性がある Intelligent Agent エラーについて説明します。前述の項と同様、エラーには、発生の確率によって優先順位がつけられています。

### ‘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にのみ構成可能です。

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、次の通りです。

- 「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 サーバー上では、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。Agent がローカル IPC 接続を使用している場合、Agent は Named Pipes を使用します。これは NetBIOS の名前ですが、その一方で、すべての外部接続には TCP/IP の名前が使用されます。

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** Agent 用に作成した Windows NT ユーザー（第 2 章「インストールと構成」および『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）に対する読取りおよび書き込み許可と、SYSTEM32 ディレクトリの読取り許可を与える必要があります。

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。

**検出の後で Agent がサービスを見つけられない** この問題は、リリース 7.3.4 およびそれ以降の Agent で修正されています。リリース 7.3.3 およびそれ以前の Agent では、次の代替手段を取ることができます。

listener.ora ファイルの SID\_LIST セクションで、\$ORACLE\_HOME パラメータが指定されていないことを確認します。SID\_LIST セクションで \$ORACLE\_HOME が指定されていると、Agent はサービス検出のために必要なファイルを見つけられなくなります。

**ジョブまたはイベント登録時の‘サービス名が無効です’** この問題は、リリース 8.0.4 の Agent を使用している場合に起きる可能性があります。“world” 以外のデフォルト・ドメインがある場合、Agent は検出時に “world” をデータベース名に追加しようとします。たとえば、デフォルト・ドメインが nl.oracle.com で、GLOBAL\_DBNAME = database.nl.oracle.com のように定義されている場合、Agent はデータベース名を database.nl.oracle.com.world のように定義します。この問題は、Agent とコンソールが同じマシン上で動作する（同じ構成ファイルを共有する）場合に限り発生します。

代替の手段は、指定されたドメインをその時点で持っていないすべてのサービスに .world を付加することです。

**OracleAgent サービスが起動時に停止する** この問題は、システム・パス変数上にマップされた外部ドライブが存在する場合に起こることがあります。詳細は、B-10 ページの「[Agent の正常起動の確認](#)」を参照してください。

Intelligent Agent 7.3.3 と、それ以外のリリースの Oracle データベース（7.3.2 など）の組合せに固有の問題が発生することもあります。そのような問題が発生しているかどうかは、Windows NT の「コントロールパネル」→「サービス」ダイアログで確認できます。問題が発生している場合、OracleAgent のステータスが「起動」と表示されている一方で、Agent のシャットダウンを選択しようとしても、「起動」、「一時停止」、「続行」の各ボタンとともに「停止」ボタンが使用不可のままになっています。

Windows NT 上では、Intelligent Agent 7.3.3 と、リリース 7.3.2 のデータベースを同時にインストールできません。これは、両製品が 2 種類の異なるリリースの RSF（7.3.3 および 7.3.2）を使用することが原因です。これらの RSF を同時にインストールすることはできません。



SQL\*Net 2.3.3 とともに、データベースをリリース 7.3.3 にアップグレードしてください。この処置が不可能な場合、Oracle Enterprise Manager 1.3.5 および Intelligent Agent 7.3.3 を削除した上で、SQL\*Net 2.3.2 クライアント / サーバー / アダプタがインストールされていることを確認する必要があります。次に、Intelligent Agent 8.0.3 をインストールし、Oracle Enterprise Manager 1.4 を使用します。Intelligent Agent 8.0.3 は RSF 8.0.3 および Net8 を使用するため、Oracle 7.3.x との競合はありません。

**ジョブを高頻度で実行後に Agent が停止したように見える** スケジューリングされたジョブを毎分実行していて、NT コンソール上に “Event ID 2009, Number of sessions exceeded 2048” というエラーが表示される場合、「スタート」→「プログラム」→「管理ツール」→「Windows NT 診断プログラム」の順にクリックし、「ネットワーク」タブで、ログインしているユーザーの数が 2048 を超えていないかどうかを確認します。スケジューリングされたジョブの実行を終了するときに、Intelligent Agent はセッションを解放しません。

代替の手段として、2 時間おきに Agent を停止し、再起動してください。

または

リリース 8.0.4 の Agent にアップグレードしてください。

### Agent がワトソン博士エラーを表示して異常終了する

この問題は、NT サービス OracleServiceORCL が動作していないか（ORCL 以外の SID でデータベースを作成したことが原因で）存在せず、かつ、Oracle8 Performance Utility がインストールされているときに発生する可能性があります。

Oracle8 Performance Utility はインストールの間に、SID“ORCL”を参照する BEQ 接続記述子をハードコードします。したがって、SID が ORCL であるデータベースがマシン上にない場合に、前述の問題が発生します。接続記述子の場所は (HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Services\Oracle80\Performance) です。Username、Password および Hostname の値を確認します。

解決策は、Oracle8 Performance Utility を削除することです。

この処置で問題が解決しない場合、sqlnet.ora ファイルで変数を automatic\_ipc = off のように設定し、listener.ora ファイル中の IPC アドレスをコメント・アウトして、Agent を再起動してみてください。

**ワトソン博士： アプリケーション PLUS80.EXE のアクセス違反** SQL\*Plus ジョブの実行中にこのエラーが発生する場合、SQL\*Plus のバグが原因です。tnsnames.ora ファイルに記述された別名以外の接続文字列を使用してコマンド・ラインから SQL\*Plus を起動すると、同じエラーが発生して SQL\*Plus が異常終了します。これは、接続文字列のサイズが、接続文字列に割り当てられたバッファ・サイズよりも大きいことが原因です。詳細は、B-23 ページの「ORA-12163: TNS: 接続記述子が長すぎます」を参照してください。

tnsnames.ora ファイルで、255 文字よりも短いデータベース接続文字列を定義してください。その後、Agent を再起動し、該当のノードをコンソール上で再検出します。

**Agent が重大な NT エラーを発してサービスが制御不能になる** この問題は、NT マシン上の同じ ORACLE\_HOME ディレクトリに、Oracle7 と Oracle8 の両方がインストールされている場合に発生します。これはサポートされている構成ではありません。Agent では起動を試み、なんらかの種類のタイムアウト・エラーを発し、起動に失敗したことを知らせるメッセージを表示した後、Agent サービスの停止または開始の操作ができない状態になります。

まず Agent を起動し、次に Oracle8 サービスを起動してください。

**OEM 1.5 へのアップグレード後の OS および ORA エラー** この問題は、Windows NT 上で、リリース 8.0.3 のデータベースとともに OEM 1.5 をインストールするときに発生します。インストールの間、必要なサポート・ファイルが 8.0.3 から 8.0.4 にアップグレードされます。

次回以降 Agent を起動しようとする、エラー・スタックが返されます。NT マシンを再起動すると、データベースがない、Oracle Enterprise Manager コンソールが 1 つしかない、Agent がないなど、動作に異常がみられます。

このことは既知の問題であり、次のように定義されます。Windows NT 上では、最初の 2 つの番号のみがサポートされます。小数点第 3 位のみが異なる 2 種類のバージョンをインストールしようとしても、正しくインストールできません。OEM 1.5 の番号のみでは、リリース 8.0.4 の RSF を同時にインストールしていることに気が付きません。

## UNIX 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、SQL\*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。

### ‘ノード <nodename> のサービス情報の取得に失敗しました’ と表示されて自動検出が失敗する

これは Solaris 版のリリース 7.3.3 における既知のバグで、リリース 7.3.4 および 8.0.3 では修正済です。このエラーは、クライアント側のインストールなど、oratab ファイル中にデータベース以外のエントリがある場合に発生します。

そのようなエントリに対しては、Agent の起動時に次のようなメッセージが表示されます。

```
No listener found for SID <sid>
```

これには 2 種類の対処方法があります。

1. Agent の起動時に、`oratab` ファイル中のそれらのエントリをコメント・アウトします。
2. `oratab` ファイルのコピーを作成し、それらのエントリを削除します。次に、Agent を起動するユーザーの `$ORATAB` 環境変数をそのファイルに設定します。

## lsnrctl dbsnmp\_start の実行中に ‘NO LISTENER FOUND FOR <XXX>’ のように表示される

これは Intelligent Agent 7.3.3 での問題です。`services.ora` ファイルが空です。または、`snmp_ro.ora` ファイル中の `SNMP.VISIBLESERVICES` パラメータが空です。

“cannot find listener for XXXX” というメッセージの原因は通常、`listener.ora` ファイルに記述のない SID が `oratab` ファイルに記述されていることです。これは、リスナーがこの特定の SID をリスニングしていないことを意味します。Agent を接続しようとしている SID とこの SID が一致しない場合、これは警告であり、その他の SID に対する検出プロセスが停止されるわけではありません。

`oratab` および `listener.ora` の中で、すべての SID に対して大文字 / 小文字が正しく区別されており、\* やその他のプレフィクスが付いた SID が `oratab` ファイル中に存在しないことを確認してください。この問題の例として、\* で始まる SID が Agent 以外の Oracle 製品で使用されていて、それにより検出が停止した、などの事例があります。

# Intelligent Agent のエラー・メッセージと解決策

## Agent 全般

### ORA-12163: ‘TNS: 接続記述子が長すぎます’

`$ORACLE_HOME/network/admin/snmp_ro.ora` ファイルから、`snmp.address.<host_name>` パラメータをコピーします。このアドレスおよびパラメータを、`$ORACLE_HOME/network/admin/snmp_rw.ora` ファイルにペーストします。`snmp_rw.ora` では、IPC に対するアドレス・エントリを削除することによって、この接続文字列の長さを切りつめます。(NMP および SPX も削除できます。)

Agent をシャットダウンし、再起動します。次の例を参照してください。

---

**注意：** リリース 7.3.4/8.0.3 以降の Agent では、`snmp_ro.ora` にパラメータ `snmp.address` は存在しません。したがって、この例を参考にして、`snmp_rw.ora` に新しい変数を追加する必要があります。

---

**例：**

snmp\_ro.ora から、次のエントリをコピーします。

```
snmp.address.ORCL_MACHINE-PC = (DESCRIPTION=(ADDRESS_LIST
=(ADDRESS=(PROTOCOL=IPC) (KEY=oracle.world)) (ADDRESS=(PROTOCOL=IPC) (KEY=ORCL)) (ADDRESS=
S=(COMMUNITY= TCP.world) (Host=machine-pc)
(PROTOCOL=TCP) (Port=1521)) (ADDRESS=(COMMUNITY=TCP.world) (Host=machine-pc)
(PROTOCOL=TCP) (Port= 1526))) (CONNECT_DATA=(SID=ORCL) (SERVER=DEDICATED)))
```

snmp\_rw.ora で修正されたエントリは次のとおりです。

```
snmp.address.ORCL_machine-PC = (DESCRIPTION=(ADDRESS_LIST
=(ADDRESS=(COMMUNITY=TCP.world) (Host = machine-pc) (PROTOCOL= TCP) (Port=
1521)) (ADDRESS=(COMMUNITY= TCP.world) (Host = machine-pc) (PROTOCOL=
TCP) (Port=1526))) (CONNECT_DATA=(SID=ORCL) (SERVER=DEDICATED)))
```

**TNS-12542: 'TNS: アドレスがすでに使用中です'**

これは実際には、Net8 リスナーのエラーです。

次に示すのは、リリース 8.0.3.0.0 Intel NT 版 Net8 リスナーのリリース・ノートから抜粋した説明です。クライアントが専用のサーバー・モードで Oracle8 Server に接続するとき、WINSOCK2 の共有ソケット機能を使用して、クライアントの接続がリスナーからデータベース・サーバーにルーティングされます。クライアントはリスナーとのソケット接続をクローズする必要なしに、データベース・サーバーと新しい接続を確立するため、この機能により接続時間が短縮されます。

共有ソケットを使用する場合、スレッドもリスナーと同じポートを使用します。リスナーをシャットダウンし、同じポートでそのリスナーを再び起動しようとするとき、データベースとの間のなんらかのオープンな接続によりポートが使用中であれば、リスナーは起動しません。リスナーを起動する前に、データベースに接続しているクライアントがないことを確認してください。別のポート番号でリスナーを使用する場合には、リスナーを起動できます。

---

---

**警告：** データベースに接続しているクライアントが1つでもあるときには、リスナーを停止しないでください。新しいデータベースをリスニングする必要がある場合、構成ファイル listener.ora を修正して、リスナー制御ユーティリティ LSNRCTL80 から reload コマンドを発行します。

---

---

listener.ora ファイルと LSNRCTL80 ユーティリティの詳細は、Oracle ネットワーク製品の Windows プラットフォーム用のスタート・ガイドを参照してください。オラクル社では、WINSOCK2 のオプションを使用してポートの再使用を可能にすることにより制限の克服を試みましたが、オプション機能の信頼性は十分ではありません。オラクル社では現在、マイクロソフト社と協力して、この問題の解決に取り組んでいます。

reload コマンドの詳細は、『Oracle8i Net8 管理者ガイド』を参照してください。

**VOC-04816 ‘宛先が無効です’** ジョブの送信中、「Agent\_node のアドレスの検索に失敗しました」というメッセージを表示して認証が失敗します。次に、VOC-04816 ‘宛先が無効です’ エラーが発生します。これは、コンソール・マシンの tnsnames.ora 中に無効なアドレスが記述されていることが原因である可能性もあります。

リリース 7.3.3 またはそれ以降の Agent にアップグレードしてください。

また、SQL\*Net の構成ファイル群が正しいことを確認してください。

## NT Agent

### Agent 起動時の NT オペレーティング・システム・エラー

Agent の起動時に OS エラーが発生する場合、それが実際に snmimsg.mc に記述されている Agent エラーであるかどうかを確認してください。一部の Windows API が定義どおりに機能しないことが原因で、Agent がエラーの実際の原因を出力できないことがあります。

Windows NT の「管理ツール」グループにある「イベント ビューア」を使用してください。記録された問題の本当の原因を見つける必要があります。Agent エラーのソースは、サービス名 db snmp の下にあります。リスト中の最も新しい db snmp 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。

OS エラーの発生後に 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。Agent エラーのソースは、サービス名 db snmp の下にあります。リスト中の最も新しい db snmp 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。
- NMI.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。
- Agent を起動したユーザーが、キュー・ファイルに対する読取りおよび書込み許可を持っていることを確認します。
- マシン名と DNS 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 (Windows NT)」を参照してください。SQL\*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¥MIB 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。

- NT マシン上で、8.0.3 データベースと同じ ORACLE\_HOME ディレクトリに Oracle Enterprise Manager 1.5 がインストールされていないことを確認します。
- Oracle 8.0.3 以前のデータベースを使用している場合、実行している Agent のインスタンスが 1 つのみであることを確認します。Oracle リリース 8.0.4 以降では、新しい Agent サービス (OracleAgent80) を作成することにより、マシン上で 2 つの Agent サービスを動作させることができます。
- リリース 8.0.4 の Agent にアップグレードします。

## UNIX Agent

**Agent 起動時の NMS-004 エラー** UNIX マシン上で snmpd が動作している場合、サーバーの /etc/snmpd.conf ファイルで次のように設定します。

```
smux 0.0 "" <ipaddress of server>
```

nms-4 エラーが解消されない場合、次のパラメータを指定して、snmpd.conf ファイルのロギングを有効にします。

```
logging file=/usr/tmp/snmpd.log enabled
```

ログ・ファイルには、発生する可能性のある状況についてのより詳しい情報が記されています。たとえば、smux 行の二重引用符の間に空白があると、アプリケーションによってその空白がパスワードとして誤って解釈される可能性があります。二重引用符自体はパスワードを意味しません。

**NMS-0308: ‘アドレスでのリスニングに失敗しました: 別の 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 Agent がインストールされている場合、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 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 Agent のみを配置してください。

2. 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
```

^----- これはポート番号です

“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 (“LISTENING” と表示されます。)
- LSNRCTL> dbsnmp\_status (“The db subagent is not started” 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)
- 次のコマンドを入力して、Agent を再起動

```
LSNRCTL> dbsnmp_start
```

この時点でなお 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、Agent を再起動する前に次の操作を行います。

- cd \$ORACLE\_HOME/network/agent
- rm \*.q, services.ora, snmp\_ro.ora, and snmp\_rw.ora
- 次のコマンドを入力して、Agent を再起動

```
LSNRCTL> dbsnmp_start
```

この操作により Agent は再起動しますが、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確実にポートが解放されます。

この問題を解消するために、Agent を再リンクする必要がある場合もあります。詳細は、『Oracle Enterprise Manager 構成ガイド』および『Oracle Enterprise Manager 日本語リリース・ノート』を参照してください。

**Agent 起動時の NMS-001 エラー** 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、Agent は SQL\*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。

**Agent 起動時の NMS-205 エラー** ‘dbsnmp’ ユーザーを見つけられませんでした。

SYS または INTERNAL どちらかのアカウントで、データベースに対して catsnmp.sql スクリプトを実行します。

**Agent 起動時の NNL-018 エラー** Agent が Names Server と通信できません。この問題は、インストールされているのみで実際には Names Server が使用されていない場合に発生する可能性があります。

snmp\_rw.ora ファイルに次の行を追加します。

```
NMI.REGISTER_WITH_NAMES = FALSE
```

**Agent 起動時の NMS-351 エラー** 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。  
\$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。Agent を再起動します。

## Agent のトレース

リリース 7.3.3 以降では、Agent は \$ORACLE\_HOME/network/admin ディレクトリにある snmp\_ro.ora および snmp\_rw.ora の各ファイルから情報を読み取ります。

---

---

**注意：** これらのファイルは、Agent を最初に起動した後でのみ存在します。Agent の起動以後、最初に Agent をトレースする場合、snmp\_rw.ora という名前の新規ファイルを手動で作成して、このファイルにトレース・パラメータを追加できます。そうでない場合は、Agent を起動し、snmp\_rw.ora ファイルにトレース情報を追加して、Agent を再起動します。

---

---

snmp\_rw.ora ファイルに加える変更の例を次に示します。

```
NMI.TRACE_LEVEL = (OFF | USER | ADMIN | 16 )
```

NMI.TRACE\_LEVEL の設定は、SQL\*Net 用に使用されるものを反映しています。

オプションとして、次のパラメータを設定します。

```
NMI.TRACE_FILE = agent          Default=dbsnmp.trc
NMI.TRACE_DIRECTORY = C:\TEMP   Default=$ORACLE_HOME/network/trace
```

(Agent が書き込み許可を持つ任意の既存ディレクトリ)

トレースが有効でない場合でも、ログ・ファイル \$ORACLE\_HOME/network/log/nmi.log は、起動のたびに Agent によって書き込まれます。このファイルには、Agent の名前とバージョン、および Agent 構成ファイルの名前とその場所が記されています。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 `$ORACLE_HOME/network/log/nmiconf.log` は、Agent の初回起動時に作成され、その後起動のたびに内容が追加されます。自動検出は、Tcl スクリプト `nmiconf.tcl` (ログ・ファイル名の由来) によって行われます。このファイルは起動の間にのみ書き込まれます。`$ORACLE_HOME/agentbin/ORATCLSH` は、特別な目的に使用される TCL シェルで、OEM 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 (TCL75.dll 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、OEM 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、OEM Agent との組合せでのみ使用できます。

ORATCLSH のドキュメントは存在せず、OEM Agent のサポート対象機能セットの一部でもありません。ORATCLSH は、OEM 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、TCL\_LIBRARY 環境変数が `$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。

## TCL のトレース

ORATCL\_DEBUG 環境変数を設定し、`snmp_rw.ora` ファイル中でトレースを有効にすることにより、Tcl のトレースを有効にできます。ORATCL\_DEBUG は、`$ORACLE_HOME/network/trace` ディレクトリに設定する必要があります。これらのパラメータを有効にするには、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、前述のディレクトリにファイル `oratcl.trc` が作成されます。イベントが実行されるたびに、`oratcl.trc` ファイルにエントリが追加されます。

# Oracle Data Gatherer の構成に関するトラブルシューティング

この項では、構成を行う際に発生する問題とその解決方法について説明します。

表 B-2 Oracle Data Gatherer の構成に関する問題のトラブルシューティング

問題	問題を修正する手順
クライアント・ホスト (Oracle Capacity Planner または Oracle Performance Manager がインストールされている) が、Oracle Data Gatherer がインストールされたホストに接続できない。	<div><div>■ クライアント・ホストから、ping ユーティリティを使用して Oracle Data Gatherer ホストに ping を実行。  ping Oracle-data-gatherer-host-name</div><div>■ データベースに対する優先接続情報が設定されており、Data Gatherer がそのホスト上に存在する場合、Data Gatherer の場所を問い合わせるメッセージは表示されない。そのホスト上の Data Gatherer が、デフォルトとして使用されます。データベースに対する優先接続情報が設定されていないか、そのホスト上に Data Gatherer が存在しない場合、Oracle データベースへのログイン用ダイアログ・ボックスで、Oracle Data Gatherer ホストの名前を入力するよう求められます。ping コマンドで入力したものと同一ホスト名を、正確に入力する必要があります。  たとえば、クライアント・ホストからホスト DAVID.COMP.COM への ping が成功した場合、Oracle データベースへのログイン用ダイアログ・ボックスで、Oracle Data Gatherer ホストの名前を DAVID と指定することはできません。Oracle データベースへのログイン用ダイアログ・ボックスで、Oracle Data Gatherer ホストの名前を DAVID.COMP.COM のように指定する必要があります。</div></div>
Oracle Data Gatherer が配置されたホストが、ターゲット・データベースに接続できない。	<div><div>■ Oracle Data Gatherer があるホストから、SQL*Plus を使用し、ターゲット・データベースのユーザー名、パスワードおよびサービスを指定して、ターゲット・データベースに接続。  SQLPLUS&gt; connect username/password@service-name</div><div>■ データベースに対する優先接続情報が設定されており、Data Gatherer がそのホスト上に存在する場合、Data Gatherer の場所を問い合わせるメッセージは表示されない。そのホスト上の Data Gatherer が、デフォルトとして使用されます。データベースに対する優先接続情報が設定されていないか、そのホスト上に Data Gatherer が存在しない場合、Oracle データベースへのログイン用ダイアログ・ボックスで、ターゲット・データベースのユーザー名、パスワードおよびサービスの指定を求められます。ユーザー名、パスワードおよびサービスの情報は、SQL*Plus で入力したときと同じものを正確に入力してください。</div></div>

構成に関する問題を修正するための手順が失敗した場合、1-3 ページの「[データベースとオペレーティング・システムのデータを収集するための Oracle Data Gatherer の構成位置の理解](#)」での説明を参考にして、TNSNAMES.ORA のエントリが正しく構成されていることを確認してください。



# キーボード・ショートカット

この付録には、一般的な Windows のキーボード・ショートカットのリストが記載されています。Windows のキーボード・ショートカットおよびナビゲーションの完全なリストは、ご使用のオペレーティング・システムのドキュメントを参照してください。

表 C-1 一般的な Windows キーボード・ショートカット

キー	動作
[F1]	アクティブ・オブジェクトまたはウィンドウ全体のヘルプ情報を表示。
Windows キーまたは [Ctrl]+[Esc]	タスクバー上にある「スタート」メニューを開く。
[Ctrl]+[Alt]+[Del]	Microsoft Windows では、「プログラムの強制終了」ダイアログ・ボックスを開く。ここには、終了する対象のアプリケーションのリストと、「終了」、「シャットダウン」および「キャンセル」のコマンド・ボタンが表示されます。  Microsoft Windows NT では、「Windows NT のセキュリティ」ダイアログ・ボックスを開く。ここには「ワークステーションのロック」、「ログオフ」、「シャットダウン」、「パスワードの変更」、「タスク マネージャ」および「キャンセル」のオプションがあります。ログインしていない場合は、ログイン・ダイアログ・ボックスが開きます。
[Del]	選択した項目を削除。項目がファイルの場合、それらを「ごみ箱」に移します。
[Shift]+[Del]	選択した項目を削除。項目がファイルの場合、それらを「ごみ箱」に移さず、ただちに削除します。
[Ctrl]+[N]	新規作成」ダイアログ・ボックスを開く。（「ファイル」メニューの「新規作成」コマンドを選択することもできます。）
[Ctrl]+[O]	「ファイルを開く」ダイアログ・ボックスを開く。（「ファイル」メニューの「開く」コマンドを選択することもできます。）
[CTRL]+[P]	「印刷」ダイアログ・ボックスを開く。（「ファイル」メニューの「印刷」コマンドを選択することもできます。）

**表 C-1 一般的な Windows キーボード・ショートカット**

キー	動作
[Ctrl]+[S]	「ファイル名を付けて保存」ダイアログ・ボックスを開く。（「ファイル」メニューの「上書き保存」コマンドを選択することもできます。）
[Ctrl]+[X]	選択した項目を切り取り、クリップボードに移す。（「編集」メニューの「切り取り」コマンドを選択することもできます。）
[Ctrl]+[Insert] または [Ctrl]+[C]	選択した項目をクリップボードにコピー。（「編集」メニューの「コピー」コマンドを選択することもできます。）
[Shift]+[Insert] または [Ctrl]+[V]	クリップボードから、コピーした項目を貼り付ける。（「編集」メニューの「貼り付け」コマンドを選択することもできます。）
[Alt]+[Back Space] または [Ctrl]+[Z]	直前の操作を元に戻す。操作の中には元に戻せないもの（シャットダウンなど）もあることに注意してください。（「編集」メニューの「元に戻す」コマンドを選択することもできます。）
[Alt]+[Shift]+[Back Space]	直前の元に戻す操作を繰り返す。（「編集」メニューの「やり直し」コマンドを選択することもできます。）
Windows キー +[M]	すべてのウィンドウを最小化。キーボードのフォーカスは、デスクトップ上で直前に選択されていたアイコンに移ります。前に開いていたウィンドウを拡大し、最後に使用していたアプリケーションにフォーカスを戻すには、[Shift] を一緒に押します。

---

# 用語集

## **MIB: 管理情報ベース (Management Information Base)**

通常関連する SNMP オブジェクトの ID (OID) の集合。

## **OID: SNMP オブジェクト ID (SNMP Object ID)**

ピリオド区切りの一連の数値で、a.b.c...x.y.z の形式。MIB の一部である情報項目に対する一意の識別子。OID には通常、それに関連付けられた名前を付けることができる。OID は本質的に階層型である。したがって、1.2.3 は 1.3 の前であるが、1.2 の後になる。たとえば、Oracle7 データベースが実行した物理読込みの数を含む OID は、次のようになる。

oraDbSysPhysReads, 1.3.6.1.4.1.111.4.1.1.1.8

## **Oracle プライベート MIB (Oracle Private MIB)**

Oracle 製品のみに固有の MIB。

## **RDBMS パブリック MIB (RDBMS Public MIB)**

Internet Engineering Task Force (IETF) によって合意された、リレーショナル・データベース用の標準 MIB。この MIB は、データベース名など、リレーショナル・データベースに対して一般に関連している様々な OID (たとえば、rdbmsDbName, 1.3.6.1.3.55.1.2.1.4 など) をサポートする。

## **SNMP: Simple Network Management Protocol**

OID を操作するネットワーク・プロトコル。Oracle のケースでは、2 種類の基本的な SNMP 操作がサポートされる。get oid は、oid の値をフェッチする。getnext oid は、oid 後の次の OID 値を取得する。

### **イベント (Event)**

イベントとは、Intelligent Agent によって監視されるデータベースまたはノードの上で起こりうる特定の条件。たとえば、データベースがダウンすると、DBDOWN イベントが発生する。イベントは次のどちらかの方法で検出できる。(1) 特定の条件を監視する Tcl スクリプトを定期的に実行する。(2) サードパーティ製品を利用して、イベントの発生を Agent に直接報告させる。

### **修正ジョブ (Fixit Job)**

イベントの発生によって引き起こされる特別な種類のジョブ。たとえば、使用済の表領域が 90% を超えたことが表領域満杯イベントによって検出されると、表領域にデータファイルを追加するための修正ジョブが自動的に実行される。

### **ジョブ (Job)**

一度限り、または繰り返しのスケジュールで実行できる Tcl スクリプトのこと。イベントが特定の条件を監視するのに対して、ジョブはなんらかのタスクを遂行することが求められる。ジョブの例には、バックアップやデータベースの起動がある。



---

# 索引

## A

---

### Agent

- Intelligent, 1-1
- プロセス, 1-1

### Agent との通信用関数

- OraTcl, 3-12

## C

---

catsnmp.sql スクリプト, 2-10

convertin, 3-13

convertout, 3-13

## D

---

dbsnmp.address, A-5

DBSNMP.POLLTIME, A-3

dbsnmp.spawnaddress, A-5

## G

---

GLOBAL\_DBNAME パラメータ, 2-9

## I

---

Intelligent Agent, 1-1, 1-2

- Tcl の使用, 3-10

- TCP プロトコル, A-5

- UNIX

  - setuid root としてのインストール, 2-5

  - インストール, 2-5

  - 起動と停止, 2-7

- Windows NT

  - インストール, 2-1

  - 起動と停止, 2-4

- インストール

  - UNIX, 2-5

  - Windows NT, 2-1

- 構成

  - Windows NT, 2-4

  - 必要なロールとユーザー, 2-10

  - ルート許可, 2-5

Intelligent Agent のデータ収集サービス

- UNIX

  - 起動と停止, 2-20

- Windows NT

  - 起動と停止, 2-20

## L

---

listener.ora ファイル, 2-9

## M

---

Management Information Base (MIB)

- SNMP, 1-3

msgtxt, 3-14

msgtxt1, 3-14

## N

---

NLS 環境, 1-2

NLS の問題とエラー・メッセージ

- 説明, 3-11

NMI.LOG\_DIRECTORY, A-4

NMI.LOG\_FILE, A-4

NMI.TRACE\_DIRECTORY, A-4

NMI.TRACE\_FILE, A-4

NMI.TRACE\_FILECNT, A-4

NMLTRACE\_FILESIZE, A-4  
NMLTRACE\_LEVEL, A-4

## O

---

oemevent, 3-22  
oraautocom, 3-15  
oracancel, 3-15  
Oracle Names, 2-9  
Oracle Server メッセージ  
    oramsg, 3-5  
    説明, 3-5  
oraclose, 3-15  
oracols, 3-16  
oracommit, 3-16  
oradbsnmp, 3-17  
orafail, 3-17  
orafetch, 3-18  
orainfo, 3-19  
orajobstat, 3-19  
oralogoff, 3-20  
oralogon, 3-20  
oramsg 要素, 3-5  
oraopen, 3-21  
oraplexec, 3-21  
orareadlong, 3-22  
orareporevent, 3-22  
oraroll, 3-24  
orasleep, 3-24  
orasnmp, 3-24  
orasql, 3-25  
orastart, 3-26  
orastop, 3-27  
oratab ファイル, 2-14  
OraTcl  
    Agent との通信用関数, 3-12  
    RDBMS 管理用関数, 3-11  
    SNMP アクセス用関数, 3-12  
    SQL および PL/SQL 関数, 3-11  
    キャラクタ・セット変換およびエラー処理用関数,  
        3-12  
    コマンドとパラメータ, 3-11  
    ジョブとイベント・スクリプト, 3-3  
    説明, 3-3  
    汎用ユーティリティ関数, 3-12  
    変数, 3-12  
oritime, 3-27

orawritelong, 3-27

## R

---

RDBMS 管理用関数  
    OraTcl, 3-11  
root.sh, 2-5  
root.sh の実行成功の確認, 2-6

## S

---

services.ora, 2-13, 2-14, A-1  
setuid root, 2-5  
SNMP  
    Agent, 1-3  
    snmp\*.ora ファイルのパラメータ, A-2  
    snmp\_ro.ora, 2-13, 2-14, A-1  
    snmp\_rw.ora, 2-13, 2-14, A-1  
    SNMP.CONNECT.service\_name.world.PASSWORD,  
        A-2  
    SNMP.CONNECT.service\_name.world.USER, A-2  
    SNMP.CONTACT.service\_name.world, A-3  
    SNMP.INDEX.service\_name.world, A-2  
    snmp.ora  
        パラメータ, A-2  
SNMP アクセス用関数  
    OraTcl, 3-12  
SQL\*Net  
    構成ファイル, A-1  
SQL および PL/SQL 関数  
    OraTcl, 3-11

## T

---

Tcl  
    Agent が実行するスクリプト, 1-2  
Tcl 言語  
    Web サイト, 3-3  
    ジョブとイベント・スクリプト, 3-1  
    説明, 3-2  
Tcl スクリプト  
    説明, 3-1  
    例, 3-3  
TCP プロトコル, Intelligent Agent, A-5  
trigevent, 3-8

## い

---

インストール  
Intelligent Agent, 1-2

## き

---

キーボード・ショートカット, 一般的な Windows プラットフォーム, C-1  
キャラクタ・セット  
Agent, 3-5  
OraTcl の変換およびエラー処理用関数, 3-12

## け

---

言語作業環境  
NLS の問題とエラー・メッセージ, 3-11

## こ

---

構成ファイル  
services.ora, A-1  
snmp\_ro.ora, A-1  
snmp\_rw.ora, A-1

## さ

---

作成  
Windows NT ユーザー・アカウント, 2-2

## し

---

修正ジョブ  
trigevent, 3-8  
ジョブとイベント・スクリプト  
Tcl 言語, 3-1

## す

---

スクリプト  
ジョブおよびイベント, 3-1

## と

---

トラブルシューティング  
Agent, B-2

## は

---

配列, 3-8  
バッチ ジョブとしてログオン, 2-2  
パラメータ  
OraTcl, 3-12  
汎用ユーティリティ関数  
OraTcl, 3-12

## ふ

---

プラットフォーム固有のインストール・マニュアル,  
1-2

