



# Siebel パフォーマンス チューニングガイド

バージョン 8.0

2006 年 12 月

ORACLE®

Copyright © 2005, 2006, Oracle. All rights reserved.

このプログラム（ソフトウェアおよびドキュメントを含む）には、オラクル社およびその関連会社に所有権のある情報が含まれています。このプログラムの使用または開示は、オラクル社およびその関連会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と工業所有権に関する法律により保護されています。独立して作成された他のソフトウェアとの互換性を得るために必要な場合、もしくは法律によって規定される場合を除き、このプログラムのリバースエンジニアリング、逆アセンブル、逆コンパイル等は禁止されています。

このドキュメントに記載されている情報は、予告なしに変更されることがあります。ドキュメントになんらかの問題があるとお気づきの場合は、書面にて当社宛お知らせください。オラクル社およびその関連会社は、このドキュメントに誤りが無いことの保証は致し兼ねます。これらのプログラムのライセンス契約で許諾されている場合を除き、プログラムを形式、手段（電子的または機械的）、目的に関係なく、複製または転用することはできません。

製品のモジュールとオプション。このマニュアルには、オプションの（場合によってはライセンスを未購入の）モジュールの説明が含まれています。Siebel のサンプルデータベースには、これらのオプションのモジュールに関連するデータも含まれています。その結果、実際のソフトウェアの実装がこのマニュアルの説明と異なる場合があります。購入したモジュールについては、購入担当者か Siebel の営業員にお問い合わせください。

このプログラムが米国政府機関、もしくは米国政府機関に代わってこのプログラムをライセンスまたは使用する者に提供される場合は、次の注意が適用されます。

米国政府の権利。米国政府を顧客として引き渡されるプログラム、ソフトウェア、データベース、ならびに関連するドキュメントおよび技術データは、適用される連邦調達規制および各省庁固有の補足規制に従った「商用コンピュータソフトウェア」または「商用技術データ」です。そのため、ドキュメントおよび技術データを含む、本プログラムの使用、複製、開示、変更および翻案は、適用される Oracle 使用許諾契約に定められる使用許諾制約、および該当する範囲で FAR 52.227-19 の「Commercial Computer Software--Restricted Rights」(1987年6月)に定める権利に従うものとします。Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

このプログラムは、核、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への用途を目的としておりません。このプログラムをかかるとして使用する際、上述のアプリケーションを安全に使用するために、適切な安全装置、バックアップ、冗長性 (redundancy)、その他の対策を講じることは使用者の責任となります。万一かかるプログラムの使用に起因して損害が発生いたしましても、オラクル社およびその関連会社は一切責任を負いかねます。

Oracle、JD Edwards、PeopleSoft、および Siebel は米国 Oracle Corporation およびその子会社、関連会社の登録商標です。その他の名称は、他社の商標の可能性がります。

このプログラムは、第三者の Web サイトへリンクし、第三者のコンテンツ、製品、サービスへアクセスすることがあります。オラクル社およびその関連会社は第三者の Web サイトで提供されるコンテンツについては、一切の責任を負いかねます。当該コンテンツの利用は、お客様の責任になります。第三者の製品またはサービスを購入する場合は、第三者と直接の取引となります。Oracle は、以下の事項につき責任を負いません。(a) 第三者製品もしくはサービスの品質、または (b) 第三者との契約のいずれかの条件の履行（製品またはサービスの引渡しならびに購入した製品またはサービスに関する保証義務を含みます）。また、第三者との取引により損失や損害が発生いたしましても、オラクル社およびその関連会社は一切の責任を負いかねます。

# 目次

## 第 1 章： 新機能

## 第 2 章： Siebel のアーキテクチャとインフラストラクチャ

- パフォーマンスとスケーラビリティについて 14
- Siebel のアーキテクチャとインフラストラクチャについて 15
- Siebel ユーザーのリクエストのフローについて 19
- パフォーマンスチューニングの用語 21

## 第 3 章： パフォーマンスのための Siebel Application Object Manager のチューニング

- Application Object Manager について 24
- AOM インフラストラクチャ 24
- AOM の展開のパフォーマンスに影響する要因 26
- AOM の展開のトポロジに関する注意事項 28
- AOM のチューニングに関する最善の方法 29
  - CPU およびメモリ使用量のために AOM コンポーネントをチューニングする 29
  - AOM キャッシュのパラメータのチューニング 34
  - AOM のパフォーマンスに影響するその他のパラメータ 35
  - AOM のメモリコンシューマ 36
- AOM のデータベース接続プールの設定 37
  - AOM のデータベース接続について 37
  - データベース接続プール使用のガイドライン 38
  - デフォルトデータベース接続のプールの設定 41
  - 専用データベース接続のプールの設定 43
- AOM のスレッドプールの使用 45

## 第 4 章： パフォーマンスのための Siebel Server インフラストラクチャのチューニング

- AOM の SISNAPI 接続プールの設定 47
- Server Request Broker (SRBroker) のチューニング 49

## 第 5 章： パフォーマンスのための Siebel Web クライアントのチューニング

- Siebel クライアントについて 52
- Siebel Web クライアントのパフォーマンスに影響する要因 53
- Siebel Web クライアントのチューニングを行うための最善の方法 54
  - 十分な処理容量の Web サーバーおよびネットワークの提供 55
  - Web クライアントのパフォーマンスのテスト 55
  - 十分なクライアントハードウェアリソースの提供 56
  - システムコンポーネントのチューニング 56
  - 設定のガイドライン 57
  - ブラウザキャッシュの管理 57
  - 静的なファイルキャッシュの指定 58
  - ビューのレイアウトのキャッシュを使ったパフォーマンスの向上 60
  - メッセージバーに関連するパフォーマンスの管理 65
  - 標準の対話性を持つアプリケーションのビジーカーソルの設定 66

## 第 6 章： パフォーマンスのための Siebel Communications Server のチューニング

- Siebel Communications Server の概要 68
- セッションコミュニケーションインフラストラクチャ 69
- セッションコミュニケーションのパフォーマンスに影響する要因 71
- セッションコミュニケーションのトポロジに関する注意事項 72
- Session Communications のチューニングを行うための最善の方法 73
  - AOM コンポーネントのチューニング 74
  - CommSessionMgr コンポーネントのチューニング 74
  - キャッシュによる AOM サーバーリソースの節約 75
  - コミュニケーション設定のパフォーマンスの向上 76
  - セッションコミュニケーションのログの設定 76
  - セッション接続の可用性の向上 78
  - 画面ポップアップのパフォーマンスの向上 79
  - Siebel CTI Connect の画面ポップアップに関するパフォーマンスの向上 79
  - 活動の作成によるパフォーマンスへの影響の確認 80

Siebel Email Response のインフラストラクチャ	80
Siebel Email Response のパフォーマンスに影響する要因	81
Siebel Email Response のトポロジに関する注意事項	82
Siebel Email Response のチューニングを行うための最善の方法	83

## 第 7 章： パフォーマンスのための Siebel ワークフローのチューニング

Siebel ワークフローについて	87
ワークフローポリシーの監視	88
ポリシー頻度分析ビューの使用	88
ワークフローエージェント追跡ログの使用	89
ワークフローポリシーテーブルの監視	89
パフォーマンスのためのワークフローポリシーのチューニング	90
Siebel Server の負荷を管理するワークフローポリシーグループの作成	90
複数のワークフローモニターエージェントとワークフローアクションエージェント	91
複数の Siebel Server 上でのワークフローエージェントの実行	91
最適なワークフローポリシーグループのスリープ間隔の設定	91
ワークフローモニターエージェントとワークフローアクションエージェントに最適なアクション間隔を設定する	92
ワークフロープロセスのチューニング	93
Search Specification パラメータの使用の最小化	93
親ビジネスコンポーネントと子ビジネスコンポーネントに基づく条件の監視	94
ワークフローのパフォーマンスのために Siebel Business Applications を設定する	94
ワークフロープロセスのメモリのオーバーヘッドの監視	94
パフォーマンスのための Workflow Process Manager のチューニング	96
ビジネスサービスのキャッシュ	96
セッションのキャッシュ	96

## 第 8 章： パフォーマンスのための Siebel Configurator のチューニング

Siebel Configurator のインフラストラクチャ	100
Siebel Configurator のパフォーマンスに影響する要因	100
Siebel Configurator のトポロジに関する注意事項	101
AOM コンポーネント内での Siebel Configurator の実行	101
専用サーバー上での Siebel Configurator の実行	102
Siebel Configurator のチューニングを行うための最善の方法	103
Siebel Configurator のチューニング	104
Siebel Configurator ファイルシステムの場所の指定	105
カスタマイズ可能製品のモデルとクラスの定義	105

Siebel Configurator のキャッシュについて	106
Siebel Configurator のデフォルトのキャッシュ動作	107
Siebel Configurator のキャッシュ管理	108
Siebel Configurator のキャッシュの設定パラメータ	109
キャッシュパラメータのサイズ決定	112
Siebel Configurator キャッシュの管理	112
Siebel Configurator キャッシュ全体のリフレッシュ	113
製品の変更による Siebel Configurator キャッシュのリフレッシュ	113
製品クラスの変更による Siebel Configurator キャッシュの更新	114
製品クラスの変更による Siebel Configurator キャッシュのリフレッシュ	114
属性定義の変更による Siebel Configurator キャッシュの更新	115
属性定義の変更による Siebel Configurator キャッシュのリフレッシュ	115

## 第 9 章： パフォーマンスのための Siebel EAI のチューニング

Siebel Enterprise Application Integration について	117
Siebel EAI のチューニングを行うための最善の方法	118
IBM WebSphere MQ Transport パフォーマンスの向上	119
HTTP Inbound Transport のパフォーマンスの向上	121
EAI Siebel アダプタのパフォーマンス	121
仮想ビジネスコンポーネントのパフォーマンス	123
Workflow Process Manager のパフォーマンスの向上	124
Siebel EAI に関するその他の最善の方法	125

## 第 10 章： パフォーマンスのための Siebel EIM のチューニング

Siebel EIM について	128
EIM アーキテクチャ計画の要件	129
データベースのサイズ設定のガイドライン	129
データベースレイアウトのガイドライン（論理および物理）	130
EIM の使用計画	131
チームの定義	131
Siebel アプリケーションへのデータのマッピング	132
EIM プロセスのテスト	133
EIM を最適化するときの一般的なガイドライン	134
EIM プロセスを実装する際の推奨手順	135
EIM のパフォーマンスのトラブルシューティング	138
EIM のための SQL の最適化	138
USE INDEX HINTS パラメータと USE ESSENTIAL INDEX HINTS パラメータの使用	139
例：USE INDEX HINTS パラメータと USE ESSENTIAL INDEX HINTS パラメータの使用	139

USE INDEX HINTS と USE ESSENTIAL INDEX HINTS : データベースへのインデックスの 引き渡しの EIM 基準	141
SQLPROFILE パラメータの使用	142
EIM テーブルの追加インデックス	144
EIM テーブルに関する適切な統計の作成	145
初期実行でのインデックスの削除	146
バッチサイズの制御	146
EIM テーブルのレコード数の制御	147
USING SYNONYMS パラメータの使用	148
NUM_IFTABLE_LOAD_CUTOFF 拡張パラメータの使用	148
Docking: Transaction Logging の無効化	148
トリガーの無効化	149
EIM タスクの並列実行	149
<b>EIM を最適化するときのデータベースのガイドライン</b>	<b>150</b>
Microsoft SQL Server	150
Oracle データベース	153
IBM DB2 UDB	155
IBM DB2 UDB for z/OS	158
EIM のための IBM DB2 ローディングプロセス	158
IBM DB2 ローディングプロセスに関する一般的な推奨事項	159
<b>EIM を最適化するためのデータ管理ガイドライン</b>	<b>160</b>
<b>EIM を最適化するための実行パラメータガイドライン</b>	<b>161</b>
<b>EIM タスク時の Siebel Server の監視</b>	<b>161</b>
<b>第 11 章 : パフォーマンスのための Siebel Remote のチューニング</b>	
Siebel Remote について	163
Siebel Remote Server のコンポーネントのチューニング	164
Database Extract コンポーネントと Parallel Database Extract コンポーネントのスループットの 向上	164
Transaction Router コンポーネントのチューニング	165
Siebel Remote の展開におけるモバイル Web クライアントのチューニング	167
アプリケーション設定ファイルのパラメータの最適化	167
同期のための最善の方法	169
適切なルーティングモデルの選択	169

## 第 12 章：パフォーマンスのための顧客設定のチューニング

顧客設定のための一般的な最善の方法	172
その他の設定ガイドライン	172
生成された SQL のパフォーマンスの問題に関する分析	175
Siebel スクリプトに関する最善の方法	179
Siebel スクリプトに代わる宣言的な手段の使用	179
最適なパフォーマンスのための Siebel スクリプトガイドライン	180
データオブジェクト層についての最善の方法	183
マルチリンガル LOV クエリーとキャッシュのパフォーマンス	183
ソートおよび検索時のデータベースインデックスの管理	183
標準カラムの再使用	185
ビジネスオブジェクト層についての最善の方法	188
Cache Data プロパティによるビジネスコンポーネントのパフォーマンスの向上	188
アクティブなフィールドの数の制限	188
計算フィールドを使用するためのガイドライン	189
プロパティの使用によるピックリストのパフォーマンスの向上	190
プライマリ ID フィールドの使用によるパフォーマンスの向上	190
Check No Match プロパティによるパフォーマンスへの影響	191
ユーザーインターフェイスオブジェクト層についての最善の方法	192
グリッドレイアウトに関連するパフォーマンスの問題の解決	192
アプレットトグル使用時のパフォーマンスの維持	193

## 第 13 章：パフォーマンスのためのオペレーティングシステムのチューニング

Siebel Server のパフォーマンス向上のための Microsoft Windows のチューニング	196
すべての UNIX プラットフォーム用の Siebel Server のチューニング	197
すべての UNIX プラットフォーム用の Siebel Web Server Extension のチューニング	198
AIX 用 Siebel Business Applications のチューニング	199
AIX 用の IBM HTTP Server のチューニング	199
AIX 用の Siebel Server のチューニング	201
AIX 用のカーネル設定のチューニング	203
Solaris 用の Siebel Business Applications のチューニング	204
Solaris 用の Sun Java System Web Server のチューニング	204
Solaris 用のカーネル設定のチューニング	205
Solaris 用の Siebel Server のパフォーマンスの最大化	206
Solaris 用の AOM インスタンスのチューニング	207

HP-UX 用の Siebel Business Applications のチューニング	208
HP-UX 用の HP Apache Web Server のチューニング	208
HP-UX 用のカーネル設定のチューニング	210
HP-UX スケジューラの許可の設定	211

## 第 14 章 : Siebel ARM による Siebel アプリケーションの パフォーマンスの監視

Siebel Application Response Measurement について	213
Siebel ARM パラメータおよび変数について	214
Siebel ARM の有効化と設定	217
Siebel ARM に関する最善の方法	218

## 第 15 章 : Siebel ARM データの分析

Siebel ARM ファイルについて	219
Siebel ARM Analyzer Tool を使った Siebel ARM ファイルの分析	220
Siebel ARM Analyzer Tool について	221
パフォーマンス集計分析の実行	222
呼び出しグラフの生成の実行	223
ユーザーセッション追跡の実行	224
Siebel ARM データの CSV 変換の実行	225
Siebel ARM Analyzer 出力ファイルについて	225
パフォーマンス集計分析およびデータについて	226
呼び出しグラフの生成分析およびデータについて	232
ユーザーセッション追跡分析およびデータについて	235
Siebel ARM から CSV への変換データについて	237
Siebel ARM Query Tool を使った Siebel ARM ファイルの分析	238
Siebel ARM Query Tool について	238
Siebel ARM Query Tool の一般的なコマンド	239
Siebel ARM Query Tool の設定	240
Siebel ARM Query Tool の入力の設定	241
Siebel ARM Query Tool の出力の設定	242
Siebel ARM Query Tool でのフィルタの使用	245
Siebel ARM Query Tool による Siebel ARM データの集計	254
Siebel ARM Query Tool を使用したヒストグラムの生成	256
Siebel ARM Query Tool でのマクロの使用	257

## 索引



# 1

## 新機能

### Siebel パフォーマンスチューニングガイド、バージョン 8.0 で説明する新機能

表 1 に、Oracle の Siebel Business Applications バージョン 8.0 をサポートするためにこのバージョンのマニュアルに記載されている変更内容を示します。

表 1 Siebel パフォーマンスチューニングガイド、バージョン 8.0 で説明する新機能

トピック	説明
37 ページの「AOM のデータベース接続プールの設定」	トピックを改訂しました。データベース接続プールを有効にすると、複数のユーザーセッションが単一のデフォルトデータベース接続を共有することがなくなり、代わりにデータベース接続プールを共有します。
112 ページの「Siebel Configurator キャッシュの管理」	トピックを改訂しました。このリリースでは、カスタマイズ可能製品、製品クラス、および属性定義の変更内容により、Siebel Configurator キャッシュを一度の操作でリフレッシュできます。
150 ページの「EIM を最適化するときのデータベースのガイドライン」	トピックを改訂しました。このトピックには、IBM DB2 UDB データベースでEIMのパフォーマンスを最適化するための新たな推奨事項が記載されています。
163 ページの「パフォーマンスのための Siebel Remote のチューニング」	トピックを改訂しました。この章には、Siebel Remote のパフォーマンスを最適化するための新たな推奨事項が記載されています。
213 ページの「Siebel ARM による Siebel アプリケーションのパフォーマンスの監視」	新しい章です。この章では、Siebel Application Response Measurement (Siebel ARM) の機能の概要について説明します。
219 ページの「Siebel ARM データの分析」	新しい章です。この章では、新しいコマンドラインツールである Siebel ARM Query Tool を使用して、Siebel ARM データを分析する方法について説明します。  この章には、このガイドの以前のバージョンに記載されていた Siebel ARM Analyzer Tool の使用方法に関する情報も含まれています。



# 2

## Siebel のアーキテクチャと インフラストラクチャ

この章では、Oracle の Siebel Business Applications のアーキテクチャとインフラストラクチャの概要について説明するとともに、パフォーマンスおよびスケーラビリティのための Siebel アプリケーションのチューニングについて紹介します。ここで説明する内容は、次のとおりです。

- 14 ページの「パフォーマンスとスケーラビリティについて」
- 15 ページの「Siebel のアーキテクチャとインフラストラクチャについて」
- 19 ページの「Siebel ユーザーのリクエストのフローについて」
- 21 ページの「パフォーマンスチューニングの用語」

Siebel Business Applications の各エリアの設定方法について、このマニュアル内の他章に対する相互参照が含まれています。これらのエリアを最適にチューニングすることで、パフォーマンスとスケーラビリティのバランスを取ることができます。

Siebel Business Applications のアーキテクチャおよびインフラストラクチャについては、『Siebel Bookshelf』の次のマニュアルを参照してください。

- 『Siebel Deployment Planning Guide』
- 使用しているオペレーティングシステム用の『Siebel インストールガイド』
- 『Siebel システム管理ガイド』
- 『Configuring Siebel Business Applications』

**備考：** Siebel Business Applications のすべての実装は固有です。Siebel アプリケーションのアーキテクチャ、インフラストラクチャ、および設定はそれぞれのビジネスモデルに応じて異なる場合があります。

## パフォーマンスとスケーラビリティについて

このマニュアルでは、パフォーマンスとスケーラビリティを次のように定義します。

- **パフォーマンス**：一般に応答時間またはスループットによって測定される Siebel アプリケーションの実行能力。

たとえば、パフォーマンスの測定基準には、Siebel アプリケーションへのログインまたは Siebel Web クライアントで Siebel ビューの表示に必要な時間や、サーバーコンポーネントで一定時間内に処理できるトランザクション（リクエストとも呼ばれます）の量などがあります。

パフォーマンスを阻害する一般的な要因には、不十分なハードウェア、過度のネットワークのやり取り、過剰なカスタマイズ、不十分なネットワークインフラストラクチャなどが含まれます。

- **スケーラビリティ**：処理量が増大した場合でも良好なパフォーマンスを維持する Siebel アプリケーションの能力。

スケーラビリティは一般に、ハードウェアの観点から測定されます。たとえば、処理するユーザー数の増大に対応するために、既存のマシンに新しいプロセッサを追加したり（垂直方向のスケーラビリティ）、新しい Siebel Server マシンを追加したりした後（水平方向のスケーラビリティ）、受け入れ可能なパフォーマンスが維持されるかどうか基準になります。

スケーラビリティを阻害する一般的な要因には、柔軟性に欠けるアプリケーションモジュール構造や並列プロセスの実行不能などがあります。

パフォーマンスおよびスケーラビリティに関連する用語の定義については、[21 ページの「パフォーマンスチューニングの用語」](#)を参照してください。

## Siebel のアーキテクチャとインフラストラクチャについて

15 ページの図 1 に、Siebel Business Applications のアーキテクチャおよびインフラストラクチャの概要を示します。Siebel アプリケーションによっては、その展開方法が異なる場合があります。この図の各要素については、『Siebel Deployment Planning Guide』、『Siebel システム管理ガイド』、および使用しているオペレーティングシステム用の『Siebel インストールガイド』を参照してください。

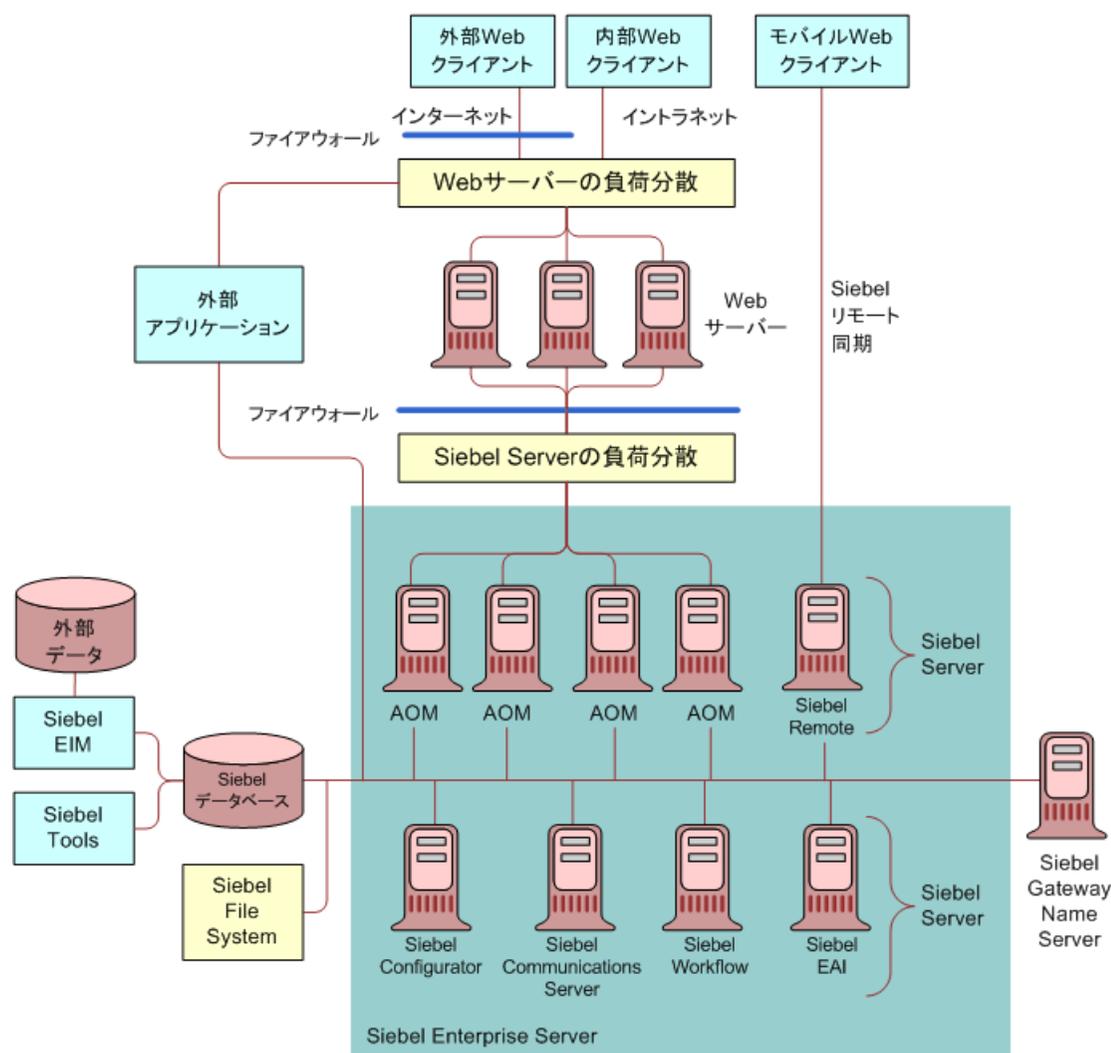


図 1 Siebel Business Applications の一般的なアーキテクチャ

## チューニングのための Siebel のアーキテクチャとインフラストラクチャエリア

次に、Siebel アプリケーションのアーキテクチャおよびインフラストラクチャの個別エリアに対するチューニングについて詳細に説明します。

これらのエリアの多くで、パフォーマンスを Siebel Application Response Measurement (Siebel ARM) を使って監視および分析できます。Siebel ARM については、[第 14 章「Siebel ARM による Siebel アプリケーションのパフォーマンスの監視」](#)を参照してください。

- **Siebel Application Object Manager (AOM)** : AOM は、Siebel Server 上に常駐する Siebel Server コンポーネントであり、Siebel Web クライアントおよび Web サーバー、または外部アプリケーションを通じて Siebel アプリケーションにアクセスするユーザーをサポートします。

AOM コンポーネントの実行方法により、パフォーマンスおよびスケーラビリティに大きな影響があります。一般的に、AOM をチューニングする目的は、システムのユーザーの増大に伴って、パフォーマンスの低下をわずかに抑えながら、またはパフォーマンスを低下させることなく、スケーラビリティを最大にすることです。

AOM コンポーネントは最適なパフォーマンスを実現するためにチューニングできますが、AOM コンポーネントや他のすべての Siebel Server コンポーネントが持つ能力は、最終的に CPU やメモリなどの Siebel Server マシンのリソースによって制限されます。

このエリアのチューニングについては、[第 3 章「パフォーマンスのための Siebel Application Object Manager のチューニング」](#)を参照してください。

- **Siebel Web クライアント** : エンドユーザーが Siebel アプリケーションの機能およびデータにアクセスする手段。Siebel Web クライアントでは、Web ブラウザを使用します。

Siebel Web クライアントのエンドユーザーが経験する応答時間は、AOM、ネットワークの帯域と遅延、Web サーバー、Siebel データベース、および Siebel アプリケーション設定 (Siebel リポジトリファイル内で指定されます) など、Siebel Enterprise 要素の設定およびチューニングの影響を受けます。また、キャッシュのブラウザ設定など、ローカルコンピュータのリソースおよび設定にも影響されます。

このエリアのチューニングについては、[第 5 章「パフォーマンスのための Siebel Web クライアントのチューニング」](#)を参照してください。[第 12 章「パフォーマンスのための顧客設定のチューニング」](#)も参照してください。

- **Siebel Communications Server** : Siebel Communications Server は、セッション通信 (音声通話など) や送受信の通信 (電子メールなど) を含めて、Siebel アプリケーションユーザーの数種類の通信活動をサポートするインフラストラクチャを提供しています。

Siebel Communication Server の処理によりエンドユーザーの応答時間が影響を受ける場合があります。またこの処理では、ユーザーセッションをサポートするための追加 AOM リソースが要求されることがあります。パフォーマンスとスケーラビリティは、サードパーティ製サーバーの設定と能力、および Siebel Server マシンのリソースと設定の影響を受けます。

このエリアのチューニングについては、[第 6 章「パフォーマンスのための Siebel Communications Server のチューニング」](#)を参照してください。

- **Siebel ワークフロー** : Siebel ワークフローは、イベントのエスカレーションおよび適切なパーティへの通知の自動化、作業のルーティングと割り当て、作業の処理、承認および推移のルールの強制など、ビジネスプロセスの自動化を行う対話式的環境です。

Siebel ワークフロー処理によりエンドユーザーの（同期リクエストの）応答時間が影響を受ける場合があります。またこの処理では、ユーザーセッションをサポートするための追加 AOM リソースが要求されることがあります。パフォーマンスおよびスケーラビリティは、Siebel Server マシンのリソースおよび設定の影響を受けます。

このエリアのチューニングについては、[第 7 章「パフォーマンスのための Siebel ワークフローのチューニング」](#)を参照してください。

- **Siebel Configurator** : Siebel Configurator は、Siebel アプリケーションのためのオーダー管理機能および製品設定機能をサポートします。

Siebel Configurator 処理によりエンドユーザーの（設定セッションの）応答時間が影響を受ける場合があります。またこの処理では、ユーザーセッションをサポートするための追加 AOM リソースが要求されることがあります。パフォーマンスおよびスケーラビリティは、Siebel Server マシンのリソースおよび設定の影響を受けます。

このエリアのチューニングについては、[第 8 章「パフォーマンスのための Siebel Configurator のチューニング」](#)を参照してください。

- **Siebel Enterprise Application Integration (Siebel EAI)** : Siebel EAI は、Siebel Business Applications を外部アプリケーションおよび内部アプリケーションと統合するためのコンポーネントと、Siebel アプリケーション間の送受信のインターフェイスを提供しています。

Siebel EAI 処理によりエンドユーザーの（リアルタイムインターフェイスの）応答時間が影響を受ける場合があります。またこの処理では、ユーザーセッションをサポートするための追加 AOM リソースが要求されることがあります。パフォーマンスおよびスケーラビリティは、Siebel Server マシンのリソースおよび設定の影響を受けます。

このエリアのチューニングについては、[第 9 章「パフォーマンスのための Siebel EAI のチューニング」](#)を参照してください。

- **Siebel Enterprise Integration Manager (Siebel EIM)** : Siebel EIM は、Siebel データベースとその他の企業データソース間においてデータを転送する Siebel EAI コンポーネントグループのサーバーコンポーネントです。

このエリアのチューニングについては、[第 10 章「パフォーマンスのための Siebel EIM のチューニング」](#)を参照してください。

- **Siebel Remote** : Siebel モバイル Web クライアント（通常、モバイル環境の切断モードでリモート操作される）は、Siebel Remote を使用すると、Siebel Server に接続し、更新されたデータやファイルを交換できます。このプロセスは、「同期」と呼ばれます。

このエリアのチューニングについては、[第 11 章「パフォーマンスのための Siebel Remote のチューニング」](#)を参照してください。

- **Siebel Tools** : Siebel Tools は、データオブジェクト、ビジネスオブジェクト、ユーザーインターフェイスオブジェクトの各層の要素を含めた Siebel アプリケーションの要素を設定するための統合開発環境です。また、Siebel Tools 環境内では Siebel スクリプト言語も管理されます。

Siebel Tools の設定およびスクリプトは、設定済みの Siebel アプリケーションのパフォーマンスとスケーラビリティに大きな影響を与えます。Siebel Tools を通じて行ったカスタマイズにより、特定の展開のパフォーマンスおよびスケーラビリティと元のインストールのパフォーマンスおよびスケーラビリティの相違の大きさがある程度まで決定します。

適切な設定では、Siebel データベースの処理が最適化されます。また、ユーザーセッションのサポートで不要なオーバーヘッドが付加されることがありません (Siebel Tools 自身は、実行時に Siebel アプリケーションに対して何らかの影響を与えることはありません)。

このエリアのチューニングについては、[第 12 章「パフォーマンスのための顧客設定のチューニング」](#)を参照してください。

- **オペレーティングシステム** : Microsoft Windows または UNIX オペレーティングシステムのチューニングについては、[第 13 章「パフォーマンスのためのオペレーティングシステムのチューニング」](#)を参照してください。

## Siebel ユーザーのリクエストのフローについて

図 2 に、Siebel Business Applications のアーキテクチャおよびインフラストラクチャ（一般的な図）内でユーザーリクエストが処理されるしくみと、パフォーマンスのチューニングが必要になる潜在的なエリアを示します。このデータフローの各部については、『Siebel システム管理ガイド』および『Siebel Bookshelf』のその他の関連マニュアルを参照してください。

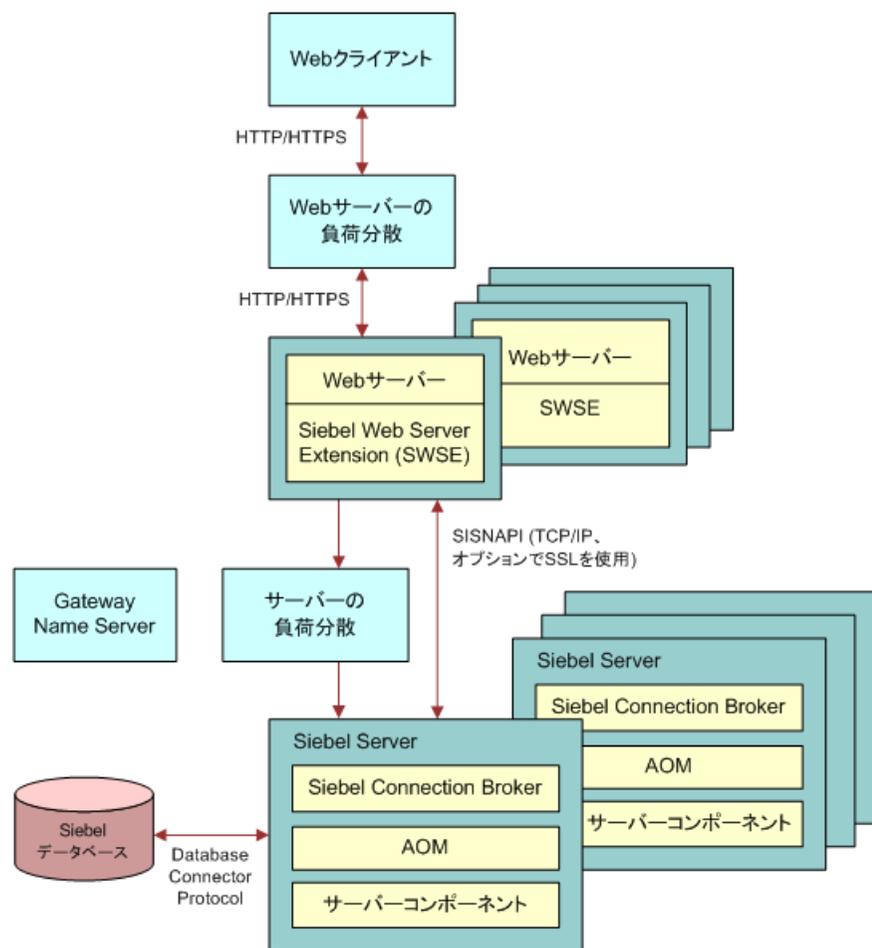


図 2 Siebel Business Applications 内の一般的なユーザーリクエストフロー

一般的な Siebel クライアントリクエストは、次に説明するような標準的なフローに従って、ユーザーの Siebel Web クライアントからシステム経由でやり取りされます。

- 1 ユーザーがリクエストを開始する操作を実行します。たとえば、ユーザーがサイトマップ内のリンクをクリックすると、特定のビューに移動します。リクエストは Web ブラウザおよび Siebel Web クライアントフレームワークによって生成されます。
- 2 リクエストは、既存または新規の HTTP 接続を使ってネットワークを通過します。リクエストはネットワークルーター、プロキシサーバー、キャッシュエンジン、その他のメカニズムを通過する可能性があります。
- 3 Web サーバーの負荷分散ソフトウェアが存在する場合、リクエストを評価し、リクエストを転送する最適な Web サーバーを決定します。次に、リクエストを Web サーバーに転送します。
- 4 Web サーバーは、HTTP リクエストを受け取り、それが Siebel アプリケーションリクエストであることを特定した上で、Web サーバーにインストールされている Siebel Web Server Extension (SWSE) にそのリクエストを転送します。
- 5 SWSE は HTTP メッセージを解析し、HTTP メッセージのコンテンツに基づいて SISNAPI メッセージを生成します。SWSE はまた、ユーザーセッション ID を取得するために受信したクッキーまたは URL を解析します。
  - Siebel の負荷分散機能を使用する場合、SWSE は Siebel Server にラウンドロビン方式でリクエストを転送します。
  - サードパーティ製の HTTP ロードバランサーを使用している場合、SWSE はロードバランサーにリクエストを転送します。ロードバランサーはユーザー設定のルーティングルールを使用して、Siebel Server にリクエストを転送します。

SISNAPI (Siebel Internet Session アプリケーションプログラミングインターフェイス) は、TCP/IP プロトコル上で動作するメッセージ形式です。Application Object Managers (AOM) と SWSE 間のネットワーク通信で使用されます。

- 6 Siebel Server 上では、AOM が SISNAPI メッセージを受信および処理します。情報を取得するためにデータベースクエリーが必要な場合は、AOM が SQL 文を構築し、データベース接続を通じて Siebel データベースにリクエストを送信します。

データベースリクエストは、データベースコネクタに固有のプロトコル形式によりデータベース接続を経由します。

- 7 データベースは、SQL 文を実行し、データを AOM に返します。AOM は、メッセージを作成元の Web サーバーに転送します。サードパーティ製の HTTP ロードバランサーを使用している場合、メッセージは Web サーバーに到達する前にロードバランサーを通過する可能性があります。
- 8 Web サーバー上の SWSE は、SISNAPI メッセージを受け取り、それを変換して HTTP に戻します。次に、HTTP メッセージを Web サーバーに転送します。この時点ではメッセージは Web ページコンテンツの形式になっています。
- 9 Web サーバーの負荷分散を実装している場合、Web ページコンテンツを元の HTTP 接続経由でエンドユーザーの Web ブラウザに転送します。
- 10 Web ブラウザおよび Siebel Web クライアントのフレームワークは、返却されたメッセージを処理して表示します。

## パフォーマンスチューニングの用語

表2に、Siebel Business Applicationsのパフォーマンスとチューニングに関連する各用語の定義を示します。「パフォーマンス」と「スケーラビリティ」の定義については、14ページの「パフォーマンスとスケーラビリティについて」を参照してください。

Application Object Manager (AOM) コンポーネントのチューニングに関連する、これらの用語と概念（同時ユーザーと待ち時間を含む）については、26ページの「AOMの展開のパフォーマンスに影響する要因」を参照してください。

表 2 パフォーマンスチューニングの用語

用語	定義
同時ユーザー	特定の時点で、SiebelアプリケーションまたはAOMプロセスなど特定要素のアクティブな使用およびアクセスを行うアプリケーションユーザーの数。
遅延	ネットワークパケットがネットワークインフラストラクチャを経由して移動するネットワーク伝送中に発生する遅れ。
待ち時間	ユーザー操作間の待機時間。たとえば、ユーザーが [取引先] 画面に移動し、別の操作を実行するまで 10 秒間データをレビューする場合、待ち時間は 10 秒となります。  平均待ち時間は、特に AOM でパフォーマンスおよびスケーラビリティのチューニングを行う場合に、非常に重要な要素です。待ち時間値を正しく予測すると、実際の負荷レベルに近い予測負荷レベルが得られます。
プロセス	オペレーティングシステム (OS) のプロセス。たとえば、AOM などの Siebel Server コンポーネントは、マルチスレッドプロセスと呼ばれる複数の OS プロセスから成ります。
マルチスレッドプロセス (MT server)	1プロセスあたりに複数のスレッド (タスク) をサポートする、マルチスレッド Siebel Server コンポーネント上で実行するプロセス。AOM コンポーネントは、スレッドをサポートするマルチスレッドプロセスを実行します。
タスク	Siebel Server コンポーネントにより実行可能な Siebel アプリケーションの作業単位概念。一般的に、Siebel のタスクはスレッドとして実装されます。
スレッド	所定の作業単位を実行するオペレーティングシステムの機能。スレッドは、大半の Siebel Server コンポーネントのタスクを実装する目的で使用されます。マルチスレッドプロセスでは、ユーザーセッションのサポートなどの作業を行うために複数のスレッドを実行できます。
応答時間	エンドユーザーが経験する、Siebel アプリケーションがユーザーリクエストに回答するのにかかる時間。応答時間は、すべてのサーバー処理と、処理の伝送遅延によって費やされる合計時間です。応答時間は、リクエストに関連する処理と、このユーザーリクエストに影響を及ぼす可能性のあるその他のリクエストの処理に基づきます。
スループット	一定時間内に処理可能なオペレーション数またはトランザクション数。一般的に、1 秒あたりのトランザクション数 (TPS) で表されます。



# 3

## パフォーマンスのための Siebel Application Object Manager のチューニング

この章では、Siebel Application Object Manager (AOM) コンポーネントの構造とオペレーション、およびオペレーションの最適化のために必要になるチューニングについて説明します。ここで説明する内容は、次のとおりです。

- 24 ページの「Application Object Manager について」
- 24 ページの「AOM インフラストラクチャ」
- 26 ページの「AOM の展開のパフォーマンスに影響する要因」
- 28 ページの「AOM の展開のトポロジに関する注意事項」
- 29 ページの「AOM のチューニングに関する最善の方法」
- 37 ページの「AOM のデータベース接続プールの設定」
- 45 ページの「AOM のスレッドプールの使用」

Siebel Server および AOM インフラストラクチャについて、および Siebel Web Client については、『Siebel Bookshelf』の次のマニュアルを参照してください。

- 『Siebel Deployment Planning Guide』
- 『Siebel システム管理ガイド』
- 使用しているオペレーティングシステム用の『Siebel インストールガイド』

## Application Object Manager について

「Application Object Manager」という用語は、Siebel Web クライアントおよび Web サーバーを通じて Siebel アプリケーションにアクセスするユーザーをサポートできる Siebel Server コンポーネントを意味します。

Siebel Business Applications または Siebel Industry Applications の各基本アプリケーションには、異なる AOM コンポーネントが用意されています。次に例を示します。

- Call Center Object Manager (SSCObjMgr\_enu) は、米語環境における Siebel Call Center の AOM です。
- Sales Object Manager (SSEObjMgr\_enu) は、米語環境における Siebel Sales の AOM です。
- eService Object Manager (eServiceObjMgr\_enu) は、米語環境における Siebel eService の AOM です。

**備考:** Siebel アプリケーションを実行する可能性があるインストール済み言語ごとに、個別に AOM が提供されています。たとえば、フランス語の Call Center Object Manager は SSCObjMgr\_fra です。

Siebel Server の AOM コンポーネントを適切に設定すると、メモリと CPU リソースを効率的に使用でき、Siebel データベース、Siebel Web Server Extension (SWSE)、および Siebel Enterprise のその他のコンポーネントと効率的に通信できます。

AOM コンポーネントのマルチプロセス、マルチスレッドモデルは、広範な同時 Siebel アプリケーションユーザーを持つ展開をサポートするスケーラビリティを備えています。

AOM の全体的なパフォーマンスは、エンドユーザーが経験する応答時間に大きな影響を与えます。

## AOM インフラストラクチャ

AOM コンポーネントは、Siebel Server 上の「マルチスレッドプロセス」として実装されます。実行時には、親プロセスが AOM 設定に従って 1 つ以上のマルチスレッドプロセスを開始します。

各プロセスは、複数のユーザーセッションを（タスクとして）ホストできます。さらにそれらのユーザーセッションは、プロセス内でスレッドとして実装されます。このスレッドは、特定のユーザーセッション専用のスレッドとして使用される場合や、複数のユーザーセッションにより共有可能なプールとして使用される場合があります（プロセスごとに、プロセスのコア機能を実行するための専用スレッドが少数開始します）。

システムにログインするユーザーが増えると、それに伴ってユーザーをホストするための追加プロセスのインスタンスが作成されることがあります。

- この章では、スレッドプールを使用する場合を除いて、「スレッド」という用語を「タスク」と同じ意味で使用します。詳しくは、[45 ページの「AOM のスレッドプールの使用」](#)を参照してください。
- また、「マルチスレッドサーバー (multithreaded server)」または「MT サーバー (MT server)」という用語は、「マルチスレッドプロセス」(複数のスレッドをサポートするプロセス)と同じ意味を持ちます。たとえば、AOM パラメータ MaxMTServers および MinMTServers の名前は、マルチスレッドプロセスを意味しています。

インタラクティブモードで動作する AOM コンポーネントは、アプリケーションのユーザーインターフェイス (UI) が常駐する Siebel Web Client セッションの処理を行います。AOM タスクは、Siebel ビジネスオブジェクトおよびデータオブジェクトを管理し、クライアントセッションのビジネスロジックを実行します。

一般に、各 AOM タスクは Web ブラウザ内で実行する Siebel Web クライアントからのリクエストに応じて開始し、クライアントの接続が切断されると終了します。

## AOM と他のモジュールの通信

各 AOM タスクは、Siebel Server インフラストラクチャ機能を使って、Siebel データベース、Web サーバー (SWSE 経由)、その他の Siebel Enterprise Server コンポーネントと通信します。

- Siebel データベースとの通信では、データベース接続が使用されます。データベース接続は、最適なパフォーマンスを得るために管理およびチューニングを行うことができます。またオプションで、データベース接続の接続プールを設定できます。

データベース接続プールの設定については、[37 ページの「AOM のデータベース接続プールの設定」](#)を参照してください。

- Siebel Connection Broker (SCBroker) との通信には、オペレーティングシステム内部のメカニズムを使用します。SCBroker は SWSE から SISNAPI 接続リクエストを受け取り、実行するタスクを最少に抑えながら、それぞれの接続リクエストを AOM マルチスレッドプロセスに転送します。接続が転送されると、リクエストは SWSE から AOM に直接渡されます。

SCBroker のチューニングについては、使用しているオペレーティングシステム用の『Siebel インストールガイド』および『Siebel システム管理ガイド』の負荷分散の項を参照してください。

- Siebel Web Server Extension との通信では、TCP/IP プロトコル上で動作するメッセージング形式である SISNAPI (Siebel Internet Session API) が使用されます。SISNAPI 接続は、Secure Sockets Layer (SSL) に基づく暗号化と認証を使用するために設定できます。

SISNAPI 通信のチューニングについては、[47 ページの「AOM の SISNAPI 接続プールの設定」](#)を参照してください。

- その他の Siebel Enterprise Server コンポーネント (他の Siebel Server を含みます) との通信では、Server Request Broker (SRBroker) 経由で SISNAPI が使用されます。

SRBroker のチューニングについては、[49 ページの「Server Request Broker \(SRBroker\) のチューニング」](#)を参照してください。

## AOM のチューニングについて

AOM コンポーネントに対する直接的または間接的なチューニング作業には、次のいくつかまたはすべてが含まれます。

- Siebel Enterprise Server 設定ユーティリティを使ってシステムの各要素を設定します。
- Siebel Server Manager を使って、Enterprise Server、Siebel Server、または AOM コンポーネントのパラメータをチューニングします。これらのパラメータは、Siebel Gateway Name Server 上のディレクトリ内の siebns.dat ファイルに保存されます。
- 各 Siebel Server 上のコンポーネントグループおよびコンポーネントを選択的に有効にします。必要なコンポーネントグループおよびコンポーネントのみを有効にします。
- Siebel Web Server Extension 上の eapps.cfg ファイル内のパラメータをチューニングします。このファイルは、Web サーバマシン上の Siebel Web Server Extension インストールディレクトリ内の bin サブディレクトリにあります。
- Siebel Call Center の uagent.cfg など、アプリケーション設定ファイル内のパラメータをチューニングします。このファイルは、Siebel Server インストールディレクトリの bin/language サブディレクトリに含まれます。このファイルの特定のセクション ([InfraUIFramework] など) のパラメータは、米語環境の Siebel Call Center の SCCObjMgr\_enu など、関連する AOM によって読み取られます。

このマニュアルの他の数章では、Siebel Communications Server、Siebel Configurator などの他のモジュールの使用に関連する AOM チューニングについて説明します。

## AOM の展開のパフォーマンスに影響する要因

AOM の展開を計画する場合、または既存の展開のパフォーマンスをトラブルシューティングする場合、パフォーマンスを決定付けたり、パフォーマンスに影響を与えたりするいくつかの要因について考慮する必要があります。

AOM を設定するタスクの焦点となる要因は、「パフォーマンスドライバ」とも呼ばれます。AOM のパフォーマンスドライバには、同時ユーザー数や平均待ち時間などが含まれます。ハードウェアリソースなど、その他の重要な要因により、全体的な容量または 1 サーバーあたりの容量の上限が決まります。

以降の項では、最適なパフォーマンスとスケーラビリティを実現および維持するのに役立つ情報とガイドラインを提供します。

これらの要因は、最初に AOM を設定するとき、特に AOM コンポーネントパラメータ MaxTasks、MaxMTServers、および MinMTServers の値を指定する場合に重要です。これらのパラメータについては、[29 ページの「CPU およびメモリ使用量のために AOM コンポーネントをチューニングする」](#)を参照してください。

### 同時ユーザー

同時ユーザーの数とは、同時にサポートされるユーザーセッションの総数です。また、これには、匿名ブラウザユーザーをサポートするセッションも含まれます。計画およびチューニングを行う目的で、次の複数のレベルで同時ユーザー数（または総ユーザー数）を考慮する必要があります。

- 展開全体（企業）
- 各 Siebel Server
- 各サーバー上の各 AOM コンポーネント
- 各 AOM コンポーネントのマルチスレッドプロセス

たとえば、特定の Siebel Server マシンが AOM コンポーネントを専用で実行する場合、Siebel Server あたりの同時ユーザーの最大数は、平均待ち時間、ハードウェアリソース、および Siebel アプリケーションの展開の特性に依存します。

設定面では、AOM の同時ユーザーの最大数は、MaxTasks パラメータの値によって制限されます。この有効最大数は、この AOM のマルチスレッドプロセス数とハードウェアリソースによっても制限されます。

平均待ち時間やその他の要因に応じて、一般的に各マルチスレッドプロセス（AOM 内のプロセス）は最大で約 100 の同時ユーザーをサポートします。MaxMTServers パラメータを使って、ピーク負荷時に必要な最大同時ユーザー数に対応する十分なマルチスレッドプロセスを設定します。

**備考：** 複雑または特殊な Object Manager コンポーネントの中には、サポートする同時ユーザー数が少ないものもあります。たとえば、Siebel Industry Applications の一部である Siebel eCommunications や、Siebel Configurator の Object Manager は、通常、約 25 人の同時ユーザーをサポートします。Siebel Configurator の Object Manager（Siebel Product Configuration Object Manager）については、[第 8 章「パフォーマンスのための Siebel Configurator のチューニング」](#)を参照してください。

## 待ち時間

待ち時間とは、Siebel アプリケーションでユーザーが実行する操作間の平均経過時間のことです。待ち時間には、ユーザーが顧客との対話、アプリケーションへのデータ入力、および他のアプリケーションでの作業に必要とする時間が含まれます。

想定される待ち時間は、マルチスレッドプロセスでサポート可能な同時タスク数に直接関係します。

ユーザーベースの一般的な使用パターンに基づいて、平均待ち時間を決定します。アプリケーションを設定した後、重要プロセスのクリックストリーム分析を実行し、各クリックによって表されるユーザー操作間の時間を測定します。Siebel Server Manager の `list statistics` コマンドを使用して、平均待ち時間を算出することもできます。

各操作（[新規] のクリックなど）間の平均時間、および各トランザクション（新しい連絡先を作成するためのすべての手順の実行など）間の総合的な平均時間を考慮します。マウスボタンのクリックは、リクエストが Siebel アプリケーションインフラストラクチャに送信されるまで操作とみなされません。これらのすべての要因に基づいて、総合的な平均待ち時間を算出します。

MaxMTServers パラメータを設定する数式では、待ち時間 30 秒に基づいて、比率 100（プロセスあたり 100 タスク）を仮定しています。この数式は [29 ページの「CPU およびメモリ使用量のために AOM コンポーネントをチューニングする」](#) で説明しています。

比率 100 は、約 3 人のユーザーがまったく同じ時間に操作することを想定しています（ $100/30 = \text{約 } 3.3$ ）。一般的に、各マルチスレッドプロセスは、パフォーマンスの低下を最小限に抑えながら、約 3 つのプロセスを同時に処理できることが観測されています。

待ち時間を長くすると、1 つのマルチスレッドプロセスが 100 を超える同時タスクをサポートできる場合があり、待ち時間を短くすると、サポートできる同時タスクも少なくなります。たとえば、ユーザー操作間の待ち時間が 15 秒である場合、1 プロセスあたり約 50 タスクをサポートできます（ $15 * 3.3 = \text{約 } 50$ 、または  $50/15 = \text{約 } 3.3$ ）。

## Siebel アプリケーションの展開の性質

使用中の Siebel アプリケーションとその他のモジュール、Siebel アプリケーションの設定方法、アプリケーションの展開方法、その他の要因もまた、AOM のパフォーマンスとサポート可能な同時ユーザー数に影響します。これらの要因のいくつかを次に示します。

- 従業員アプリケーション（Siebel Call Center など）、顧客アプリケーション（Siebel eService など）、パートナーアプリケーション（Siebel PRM など）、またはこれらのアプリケーションの併用をサポートするかどうか。一般的に、従業員アプリケーションの対話性は高く、顧客アプリケーションの対話性は標準的です。
- グローバル環境で複数の言語を使って Siebel ソフトウェアを展開するかどうか。
- Siebel Tools を使って行う変更など、どのような種類のアプリケーション設定の変更をどの程度行うか。詳しくは、[第 12 章「パフォーマンスのための顧客設定のチューニング」](#) を参照してください。

サポートできる同時タスク数は、カスタマイズのレベルまたは AOM でサポートするアプリケーションのプロセスオートメーションの用途に応じて異なります。このマニュアルの推奨事項は、実行される操作が標準的または一般的であると仮定しています。展開や使用するモジュールによっては、単一のユーザーの操作によって開始された操作が比較的複雑で、他の多くの操作より多くのリソースを必要とする場合があります。

- Siebel Configurator（製品設定）または Siebel CTI（コールセンターエージェント用のコンピュータ電話統合）のような特殊な機能を使用するかどうか。また、このような機能を展開するかどうか。さらに、このような機能を使用するユーザーベースの割合。Siebel Configurator または Siebel CTI は、特殊な機能の例にすぎません。

## ハードウェアリソース

各 Siebel Server マシンのハードウェアリソース、特に CPU とメモリが各 AOM コンポーネントでサポートできる同時ユーザー数を左右する要因になります。たとえば、4 ウェイマシンは 2 ウェイマシンの 2 倍のリソースを持ち、2 倍の同時ユーザー数をサポートできる可能性があります。AOM のパフォーマンスにとって重要なハードウェアリソースを以下にいくつか示します。

- **CPU** : CPU の性能と、1 サーバマシンあたりの CPU の数。
- **メモリ** : RAM の容量。また、過度のページングを発生させることなくユーザーに対応できるかどうか。

ディスク I/O、その他のネットワーク容量も重要なハードウェア要因ですが、AOM のチューニングに影響することはありません。これらの要因は、Siebel データベースおよび Siebel File System のパフォーマンスに大きく影響します。

AOM コンポーネントをサポートするために投入できるマシンの総数により、同時ユーザーの総数が決定します。

## AOM の展開のトポロジに関する注意事項

Siebel アプリケーションは、さまざまなトポロジまたはシステムレイアウトを使用して展開できます。AOM は、全体的な展開の一部にすぎませんが、Siebel アプリケーションユーザーをサポートする直接的かつ中心的な役割を果たします。

Siebel Server を実行するマシンの数と、AOM コンポーネントを実行するマシンの数を決定する必要があります。状況によっては、同じ Siebel Server 上で複数のコンポーネントを実行することを選択すべき場合があります。

**備考** : AOM コンポーネントは、一般的に Siebel Server マシンのリソースを消費する主要なプロセスです。この章で説明するチューニングに関する注意事項は一般的に、使用可能なリソースについて強く競合する AOM マシン上で追加コンポーネントを実行していないことを前提としています。

トポロジに関する注意事項について詳しくは、『Siebel Deployment Planning Guide』を参照してください。

## AOM のチューニングに関する最善の方法

ハードウェアリソースを最適に使用し、システムを適切に設定することにより、パフォーマンス面の目標を達成するために役立ちます。リソースと要件について慎重に考慮し、システムパフォーマンスを継続的にテストおよび監視すべきです。

詳しくは、『Siebel システム管理ガイド』とその他の資料を参照してください。チューニングの計算を行う前に、[26 ページの「AOM の展開のパフォーマンスに影響する要因」](#)で説明する全体的なシステムについて、および注意事項について理解しておく必要があります。

AOM のチューニングについて詳しくは、次の項を参照してください。

- [29 ページの「CPU およびメモリ使用量のために AOM コンポーネントをチューニングする」](#)
- [34 ページの「AOM キャッシュのパラメータのチューニング」](#)
- [35 ページの「AOM のパフォーマンスに影響するその他のパラメータ」](#)
- [36 ページの「AOM のメモリコンシューマ」](#)

## CPU およびメモリ使用量のために AOM コンポーネントをチューニングする

ここでは、AOM コンポーネントのチューニングを行うために必要になる背景知識とガイドライン、特にパラメータ MaxTasks、MaxMTServers、および MinMTServers の設定について説明します。

これらのパラメータの設定により、特定のユーザー負荷および操作が課せられているシステムのパフォーマンスが決まります。パラメータを設定することにより、Siebel Server インフラストラクチャを通じてサーバーの容量を制御できます。またこの設定は、各サーバーの全体的な容量に直接影響します。

MaxTasks、MaxMTServers、および MinMTServers のパラメータの設定は、[26 ページの「AOM の展開のパフォーマンスに影響する要因」](#)で説明する、サーバーの真の容量を決定する要素となります。

AOM コンポーネントをチューニングする場合に難しいのは、サーバーマシンでユーザーの応答時間に対する影響を最小限に抑えながら（パフォーマンス）、できる限り多くのユーザーをホストできる（スケーラビリティ）正しいパラメータ設定を見つけ出すことです。

## MaxTasks、MaxMTServers、および MinMTServers について

次に、AOM パラメータ MaxTasks、MaxMTServers、および MinMTServers について説明します。これらのパラメータは、Siebel Server Manager を使って設定します。詳しくは、『Siebel システム管理ガイド』を参照してください。

マルチスレッドプロセス、スレッド、および関連する概念については、[24 ページの「AOM インフラストラクチャ」](#)を参照してください。

- **MaxTasks (Maximum Tasks)** : この Siebel Server の AOM 上で同時に実行できるタスク (スレッド) の総数を指定します。この数を超えるタスクを追加リクエストの処理のために開始することはできません。
- **MaxMTServers (Maximum MT Servers)** : この AOM 上で同時に実行できるマルチスレッドプロセスの最大数を指定します。この数を超えるマルチスレッドプロセスを追加リクエストの処理のために開始することはできません。
- **MinMTServers (Minimum MT Servers)** : 親プロセスの開始時にこの AOM 上で開始するマルチスレッドプロセスのデフォルトの最小数を指定します。親プロセスは Siebel Server Manager により明示的に開始される場合もあります。また、コンポーネントの状態が最後に Running に設定されたときに Siebel Server が起動した場合は、親プロセスが自動的に開始します。MinMTServers を 0 に設定すると、AOM コンポーネントが無効になります。

多数のユーザーがログインすると、対応するセッションを処理するための新しいタスクと、追加タスクをサポートするための新しいマルチスレッドプロセスが開始します。タスクとプロセスは、AOM の負荷分散動作に従って、タスクおよびマルチスレッドプロセスの最大数に達するまで追加されます。詳しくは、[31 ページの「AOM パラメータの設定の効果」](#)を参照してください。

**備考:** MaxTasks、MaxMTServers、および MinMTServers は、さまざまな Siebel Server コンポーネントに適用される汎用パラメータです。ただし、この章で説明する特定の動作は AOM コンポーネントに適用されます。詳しくは、『Siebel システム管理ガイド』を参照してください。

これらのパラメータには、相互に次のような関係があります。

- MaxMTServers および MinMTServers には、一般的に同じ値が設定されます。これにより、ユーザーのログインによって新しいマルチスレッドプロセスが開始する場合のパフォーマンス低下を回避できます。MaxMTServers の値は、MinMTServers 以上にする必要があります。  
  
一般的に、親プロセスの開始時に、事前にすべてのマルチスレッドプロセスを開始しておいても問題にはなりません。マルチスレッドプロセス自体を実行するためのメモリのオーバーヘッドは、そのスレッドのオーバーヘッドは別として、最小限で済みます。
- 比率 MaxTasks/MaxMTServers により、ある特定のマルチスレッドプロセスで同時に実行可能なスレッド (タスク) の最大数が決定します。詳しくは、[26 ページの「AOM の展開のパフォーマンスに影響する要因」](#)の待ち時間についての説明を参照してください。

## AOM パラメータの設定の効果

ここでは、MaxTasks パラメータ、MaxMTServers パラメータ、および MinMTServers パラメータの特定の設定例で AOM がどのように動作するかについて説明します。より実際的な例については、32 ページの「AOM のパラメータ値を計算する式」を参照してください。

たとえば、MaxTasks = 500 および MaxMTServers = 5 の場合、比率 MaxTasks/MaxMTServers = 100 になります。つまり、この AOM 上ではマルチスレッドプロセス内で最大 100 スレッド（タスク）を実行できることを意味します。

一般的に、MinMTServers には MaxMTServers と同じ設定が指定されます。しかしこの例では、MinMTServers = 4 を前提としています。この場合、デフォルトで 4 個のマルチスレッドプロセスが開始し、合計で 400 個の同時スレッドを処理できます。

ユーザーがサーバー上のアプリケーションを起動すると、同時スレッド数が増加するとともに、次の処理が発生します。

- 同時スレッド数が増加してもそれが 400 未満であれば、この AOM でデフォルトで開始された 4 つのマルチスレッドプロセスにスレッドが分散されます。このしくみは、AOM コンポーネント内部での負荷分散の形態です
- 同時スレッド数が 400 に達している状態で新しいリクエストが届くと、この AOM の第 5 のマルチスレッドプロセスが開始します。そして、AOM の 5 つのマルチスレッドプロセス間でスレッドが分散されるようになります。
- AOM の同時スレッド数が 500 に達すると、それ以上のクライアントセッションリクエストを処理できなくなります。既存のマルチスレッドプロセスでさらなるスレッドを開始できないこと、および AOM でさらなるマルチスレッドプロセスを開始できないことがその理由です。このような AOM の状態を「マックスアウト」といいます。

ユーザーがログアウトしたとき、またはセッションタイムアウトが強制されたときに AOM の負荷が再び低下すると、スレッドが解放されます。状況によっては、スレッドが完了したマルチスレッドプロセスもタイムアウトして終了する場合があります（MaxMTServers が MinMTServers より大きい場合にのみ行われます）。

## AOM パラメータを設定するためのガイドライン

ここでは、AOM コンポーネントの MaxTasks パラメータ、MaxMTServers パラメータ、および MinMTServers パラメータを設定するための式とガイドラインについて説明します。

**備考：** 32 ページの「AOM のパラメータ値を計算する式」で示す 2 つの計算式の全要素は、展開に応じて異なります。AOM でサポートできる同時ユーザー数は、プロセッサの数、セッションタイムアウト設定、平均待ち時間などの要因に依存します。

一般的に、AOM は Siebel Server マシン上で大量のリソースを使用する唯一のコンポーネントです。複数のサーバーコンポーネントを実行する場合、または非 Siebel モジュールを実行する場合、このマシン上の AOM でサポートできる同時スレッド数は少なくなります。

次の一般的な手順に従って、これらのパラメータ値の設定方法を決定します。

- 前述した平均待ち時間、その他の要因に基づいて、同時ユーザーの総数を決定します。
- AOM を実行する Siebel Server マシンでサポートされる同時ユーザー数を決定します。次に説明する式では、同時ユーザー数は対象ユーザー数に必要に応じて匿名ブラウザユーザー数を加えた数（該当する場合）です。
- 同時ユーザーをサポートするために必要な Siebel Server マシンの数を決定します。この作業は通常、Siebel エキスパートサービスまたはプラットフォームベンダーによって行われます。

- 値を次の式に代入した上で、その他の条件に合わせて値を調整します。特に次の環境を設定します。
  - MaxMTServers の算出値が整数ではない場合は、最も近い整数に切り上げます。
  - MaxMTServers の値を調整した後に、MaxTasks/MaxMTServers の比率が整数ではない場合は、比率が整数になるように MaxTasks の値を切り上げます。
- 実際に必要な匿名ブラウザユーザーの数を測定するなど、初期パラメータ設定をテストした後、必要に応じてさらに設定を調整します。

## AOM のパラメータ値を計算する式

次に、AOM コンポーネントのパラメータ値を計算する式を示します。

- $\text{MaxTasks} = \text{target\_number\_of\_users} + \text{anon\_browser\_users}$
- $\text{MaxMTServers} = (\text{target\_number\_of\_users} + \text{anon\_browser\_users})/100$
- $\text{MinMTServers} = \text{MaxMTServers}$

最初の計算をした後に、必要に応じて MaxMTServers の値を最も近い整数に切り上げて、MaxTasks/MaxMTServers の余り (X) を算出し、MaxTasks に (MaxMTServers - X) を加算します。これは、MaxTasks/MaxMTServers の比率を整数にするために行います。

**備考：** MaxMTServers 式の値 100 は、1 マルチスレッドプロセスに対する同時タスクの比率を表しています。100 は、経験則から導き出された値にすぎません。詳しくは、次の説明を参照してください。

次に、この式の変数について説明します。

- $\text{target\_number\_of\_users}$  = AOM でサポートする同時ユーザーセッションの最大数 (アプリケーションにログイン中のユーザー)。  
同時ユーザーの最大数は、AOM の MaxTasks パラメータ値、実行中のマルチスレッドプロセスの数 (MaxMTServers および MinMTServers によって決定) によって制限され、さらにハードウェアリソースによって事実上の制限を受けます。
- $\text{anon\_browser\_users}$  = 匿名ブラウザユーザーに専用に割り当てられた AOM 上のセッションの数 (非ログインユーザーをサポートするスレッド数)。
  - 高い対話性を持つアプリケーション (一般的には Siebel Call Center などの従業員アプリケーション) では、匿名ブラウザユーザーがサポートされないため、この数は要因には含まれません。
  - 標準的な対話性を持つアプリケーション (通常は Siebel eService などの顧客アプリケーション) の場合、匿名ブラウザユーザーはおそらく対象ユーザー数の 25% 程度です。
- 100 = AOM 上の各マルチスレッドプロセスでサポート可能な同時スレッドのおおよその最大数。100 は、経験則から導き出された値です。それぞれの展開に適した値を使用します。

**備考：** AOM の使用に関するシナリオの大半で、マルチスレッドプロセスに対するスレッドの比率として 100 は適切です。ただし、展開に必要なとされるユーザーの操作と操作の間の待ち時間が 30 秒より短い場合、またはスレッドあたりの負荷が平均より大きい場合、各マルチスレッドプロセスでサポートされる同時スレッドは少なくなります。逆に、待ち時間が長い場合、または平均負荷が小さい場合、サポートできる同時スレッドは多くなります。マルチスレッドプロセスあたりのスレッドの比率と、待ち時間の関係については、26 ページの「AOM の展開のパフォーマンスに影響する要因」を参照してください。

## AOM パラメータの設定例

待ち時間など他の要因に加えて、MaxTasks、MaxMTServers、および MinMTServers の計算は、前述の *target\_number\_of\_users* および *anon\_browser\_users* の仮定に依存します。次に、Siebel Call Center および Siebel eService の設定例を示します。

### Siebel Call Center の設定例

Siebel Call Center では、たとえば待ち時間が 30 秒、*target\_number\_of\_users* = 500 と仮定します。このアプリケーションでは、*anon\_browser\_users* の影響はありません。パラメータの値は次のとおりです。

$$\text{MaxTasks} = 500$$

$$\text{MaxMTServers} = 500/100 = 5$$

$$\text{MinMTServers} = \text{MaxMTServers} = 5$$

### Siebel eService の設定例

Siebel eService では、たとえば待ち時間が 30 秒、*target\_number\_of\_users* = 500 と仮定します。各自の実装によって、*anon\_browser\_users* は *target\_number\_of\_users* の約 25% (つまり 125) にします。暫定的なパラメータ値は次のとおりです。

$$\text{MaxTasks} = (500 + 125) = 625$$

$$\text{MaxMTServers} = (500 + 125)/100 = 6.25 = 7 \text{ (切り上げ)}$$

$$\text{MinMTServers} = \text{MaxMTServers} = 7$$

MaxTasks の値を調整します。変数 X は、(625/7) の余りで、2 です。MaxTasks に (MaxMTServers - X) を加算します：625 + (7 - 2) = 625 + 5 = 630。したがって、パラメータ値の最終的な計算値は次のとおりです。

$$\text{MaxTasks} = 630$$

$$\text{MaxMTServers} = \text{MinMTServers} = 7$$

## AOM キャッシュのパラメータのチューニング

AOM では、そのメモリ使用量に影響するキャッシュがいくつか使用されています。AOM キャッシュをチューニングすると、AOM のパフォーマンスとメモリ使用量に影響します。次に、AOM によって使用される設定可能な主なキャッシュをいくつか示します。

- SQL カーソルキャッシュ
- SQL データキャッシュ

### SQL カーソルキャッシュ

SQL カーソルキャッシュは、DSMaxCachedCursors パラメータによって設定されます。このキャッシュは、データベース接続プールとともにマルチスレッドコンポーネント（AOM など）上で有効にできます。

この値は、データベース接続ごとの SQL カーソルの数を表します。Siebel Server マシンがそのメモリ容量の限界に達する前に CPU 能力の限界に達する可能性が高い AOM (Siebel Employee Relationship Management など) では、おそらく DSMaxCachedCursors パラメータのデフォルト値である 16 が適切です。このようなアプリケーションは、「CPU 負荷の高い」アプリケーションと呼ばれることがあります。

Siebel Server マシンがその CPU 能力の限界に達する前にメモリ容量の限界に達する可能性が高い AOM (Siebel Call Center など) では、DSMaxCachedCursors をより低い値に設定します (0 が適切な場合さえあります)。このようなアプリケーションは、「メモリ負荷の高い」アプリケーションと呼ばれることがあります。

一般的に、特定の AOM コンポーネントを実行する Siebel Server マシン上の CPU リソースおよびメモリリソースの使用可能性に基づいて、この値を設定する必要があります。このパラメータを設定する場合のトレードオフは、SQL カーソルのキャッシュにメモリを割り当てると SQL カーソルの作成の必要性が少なくなる反面、メモリのコストが大きくなることです。

カーソルあたりのメモリ要件は、クエリーのサイズ、データベース接続のタイプ、行のサイズ、クエリーが返す行数などの要因によって異なります。キャッシュされたカーソルの有用性は、それぞれが表すクエリーの一意性に依存します。一般に、ほとんどの Siebel アプリケーションのクエリーは一意であり、キャッシュされたカーソルの再利用のメリットはありません。

一般に、接続プールを使用して多くのユーザーがデータベース接続を共有する場合、利用可能メモリに余裕があるかぎり、キャッシュするカーソルの数を増やす必要があります。データベース接続プールについては、[37 ページの「AOM のデータベース接続プールの設定」](#)を参照してください。

### SQL データキャッシュ

SQL データキャッシュは、DSMaxCachedDatasetsPerProcess パラメータと DSMaxCachedDataSets パラメータによって設定されます。このパラメータによって、次の 2 つのタイプのデータキャッシュを管理できます。

- ほとんどの場合に役立つグローバルデータキャッシュ。このキャッシュは、DSMaxCachedDatasetsPerProcess によって制御されます。デフォルト値は 16 です。
- データベース接続プールの有無にかかわらず、有効にできる接続あたりのデータキャッシュ。このキャッシュは、DSMaxCachedDataSets によって制御されます。デフォルト値は 16 です。

CPU 負荷の高い AOM (Siebel Employee Relationship Management など) では、DSMaxCachedDatasetsPerProcess と DSMaxCachedDataSets はデフォルト値が適しています。

メモリ負荷の高い AOM (Siebel Call Center など) では、DSMaxCachedDatasetsPerProcess と DSMaxCachedDataSets を低い値に設定できます (ゼロにすることもできます)。

一般的に、特定の AOM コンポーネントを実行する Siebel Server マシン上の CPU リソースおよびメモリリソースの使用可能性に基づいて、この値を設定する必要があります。このパラメータを設定する場合のトレードオフは、SQL データセットのキャッシュにメモリを割り当てると SQL カーソルの作成の必要性が少なくなる反面、メモリのコストが大きくなることです。

SQL カーソルキャッシュの説明も参照してください。

## AOM のパフォーマンスに影響するその他のパラメータ

この項では、AOM のパフォーマンスに影響するその他のパラメータを設定するためのガイドラインを示します。

- **MemProtection** : AOM コンポーネントに対して MemProtection パラメータを「FALSE」に設定すると、パフォーマンスが向上することがあります。

このパラメータを「TRUE」(デフォルト)に設定すると、各トランザクションは多数のシリアル化された mprotect ステートメントを発行し、その結果、Siebel Server マシンのパフォーマンスが低下することがあります。

MemProtection パラメータは表示されません。[コンポーネントパラメータ] ビュータブで [隠す] をクリックして表示します。または、次のように Siebel Server Manager のコマンドラインバージョンを使用して設定することもできます。

```
change param MemProtection=False for comp component_alias_name server  
siebel_server_name
```

ここで、各項目は次のとおりです。

*component\_alias\_name* は設定する AOM コンポーネントのエイリアス名で、たとえば Call Center Object Manager のドイツ語バージョンでは SCCObjMgr\_deu です。

*siebel\_server\_name* は、コンポーネントを設定している Siebel Server の名前です。

MemProtection パラメータを「FALSE」に設定すると、パフォーマンスだけでなくスケーラビリティも向上する場合があります。

**備考** : MemProtection を「FALSE」に設定できるのは、AIX プラットフォームと HP-UX プラットフォームのみです。

- **DSPreFetchSize と DSMaxCursorSize** : DSPreFetchSize パラメータと DSMaxCursorSize パラメータは、IBM DB2 UDB for z/OS および OS/390 上の Siebel 実装のみで設定してください。これらのパラメータの設定については、『Implementing Siebel Business Applications on DB2 UDB for z/OS and OS/390』を参照してください。

他のすべてのデータベースでは、これらのパラメータを -1 に設定します。

- **EnableCDA** : AOM コンポーネントで Siebel Advisor またはブラウザベースの Siebel Configurator をサポートする必要がない場合は、Siebel Call Center の uagent.cfg など、アプリケーション設定ファイル内の [InfraUIFramework] セクションでこのパラメータを「FALSE」に設定します。

## AOM のメモリコンシューマ

ここでは、前述したキャッシュ以外の AOM コンポーネント内の主なメモリコンシューマについて説明します。ここで説明するいくつかのトピックについては、[第 12 章「パフォーマンスのための顧客設定のチューニング」](#)を参照してください。

- **データベースクライアントライブラリ**：データベースクライアントライブラリには、メタデータ、接続、カーソル、およびデータを対象にした独自のキャッシュがあります。このようなキャッシュの中には、[37 ページの「AOM のデータベース接続プールの設定」](#)で説明されているように、Siebel データベース接続プールを使用してサイズを縮小できるものがあります。
- **スクリプト**：ビジネスコンポーネント、アプレット、またはビジネスサービスで定義されたスクリプトは、スクリプトが初めて呼び出されるときに AOM のメモリに読み込まれます。  
Siebel eScript では、サーバーメモリおよびその他のリソースを適切に使用するために、各リリースに対して最適化された設定に従ってガーベジコレクションが実行されます。
- **重い設定**：アプリケーションに重い設定が適用されていると、パフォーマンスに影響します。

次に、AOM 内の他のメモリコンシューマを示します。

- **移動パターン**：アプリケーション内でのさまざまな移動のシナリオにより、グローバルキャッシュの使用効果がなくなる場合があります。
- **セッションタイムアウト**：セッションタイムアウト値を大きくすると、サーバー上で同時にアクティブになるセッション数が増え、したがってメモリの使用量が増大します。セッションタイムアウト値を小さくすると、ログインがより頻繁になる場合があります。
- **AOM ごとのユーザー数**：AOM あたりのユーザー数が増加すると、ユーザー間でのグローバルリソースの共有が増えることとなります。この AOM 上でのユーザーあたりのメモリ使用量は減少しますが、全体としてのメモリ使用量は増加します。
- **ビュー上のアプレットの数**：ビュー上で設定するアプレットが増えると、同時に必要になるビジネスコンポーネントが増え、したがって全体的なメモリ使用量が増大します。
- **PDQ サイズ**：PDQ（定義済みクエリー）リスト内の項目リストは、現在のビジネスオブジェクトのサーバー上に保持されます。このリスト内の項目数が多いほど、消費されるメモリが増えます。また、PDQ 文字列のサイズもメモリ使用量の決定要素となります。

## AOM のデータベース接続プールの設定

この項では、AOM のデータベース接続の設定オプション、特にデータベース接続プールについて説明します。次のトピックが含まれています。

- 37 ページの「AOM のデータベース接続について」
- 38 ページの「データベース接続プール使用のガイドライン」
- 41 ページの「デフォルトデータベース接続のプールの設定」
- 43 ページの「専用データベース接続のプールの設定」

**備考：**顧客は、RDBMS に各自のニーズを満たす十分なデータベース接続数があるかどうかを確認する必要があります。利用可能な接続の総数は、RDBMS、オペレーティングシステムのプラットフォーム、およびその他の要因の制限を受けることがあります。接続プールを設定する前に、AOM が使用できるデータベース接続の数を確認します。Siebel 以外のコンポーネントによる RDBMS のパフォーマンスおよびデータベース接続の使用については、この項の対象範囲外です。

## AOM のデータベース接続について

ここでは、非プール接続およびプール接続を含めて、AOM コンポーネントのデータベース接続の概要について説明します。以降の項では、さまざまなタイプのデータベース接続プールの設定のガイドラインと処理手順を示します。

### 非プールデータベース接続について

データベース接続をプールしない場合、データベース接続数は AOM セッション数に対応します。つまり、データベース接続はプールされません。非プールデータベース接続を使用するために、特別の AOM 設定は必要ありません。プールを設定しない場合、ユーザーセッションが終了するとデータベース接続は閉じられます。

- **非プールのデフォルトデータベース接続：**非プールデータベース接続では、セッションログイン中にユーザーのデータベース資格情報を使用してデータベース接続が確立されます（LDAP などの外部認証システムが使用されている場合は、ユーザーのデータベース資格情報と Siebel 資格情報が一致しない場合があります）。

このデータベース接続がセッションにバインドされます。またこれは、読み取り、書き込み、更新、および削除の各処理で使用されるデフォルトのデータベース接続になります。

このマニュアルでは、このような接続をデフォルトデータベース接続と呼びます。この接続は、この項の後半で説明するように、プールすることもできます。

- **非プールの専用データベース接続：**セッション中に AOM の外部トランザクション管理機能を使用する専用機能が起動されると、この専用用途のために第 2 のデータベース接続が開かれます。

このデータベース接続はセッションにもバインドされ、セッションにより実行される外部的な制御下の全トランザクションで使用されます。Siebel EAI コンポーネントは、外部トランザクション管理を行う専用コードの例です。

このマニュアルでは、このような接続を専用データベース接続と呼びます。この接続は、この項の後半で説明するように、プールすることもできます。

## プールデータベース接続について

非プールデータベース接続で説明した 2 タイプと同じデータベース接続を、AOM コンポーネントに対して設定することもできます。

- **プールされたデフォルトデータベース接続**：このデータベース接続はプールして、共有（多重化）、持続設定、またはその両方をサポートできます。
  - 共有接続では、同じデータベース接続上の複数の SQL 文についてデータベース処理を多重化（共有）することにより、複数のユーザーセッションを同時にサポートできます。共有接続を使用すると、同じ接続数で、より多くのユーザーをサポートできます。
  - 永続接続はプールされますが、必ずしも共有されるわけではありません。永続接続を使用すると、データベース接続の作成コストを軽減してパフォーマンスを向上できます。すべての共有接続は永続接続です。

詳しくは、[38 ページの「データベース接続プール使用のガイドライン」](#) および [41 ページの「デフォルトデータベース接続のプールの設定」](#) を参照してください。

- **プールされた専用データベース接続**：このデータベース接続は一度に単一のセッション専用割り当てられ、専用の用途で使用されます。このような接続をプールすると、永続性は提供できますが共有は行われません。この接続を永続的にプールすると、データベース接続の作成コストを軽減してパフォーマンスを向上できます。

**備考**：専用データベース接続ではなく、デフォルトデータベース接続にプールを設定すると、それぞれの専用データベース接続は、要求されたトランザクションが完了した時点で閉じられます。

詳しくは、[38 ページの「データベース接続プール使用のガイドライン」](#) および [43 ページの「専用データベース接続のプールの設定」](#) を参照してください。

## データベース接続プール使用のガイドライン

次のガイドラインに従って、データベース接続プールを使用する必要があるかどうかを確認し、接続プールの展開方法を検討してください。

次に示すデータベース接続プールのための AOM パラメータの設定については、[41 ページの「デフォルトデータベース接続のプールの設定」](#) および [43 ページの「専用データベース接続のプールの設定」](#) を参照してください。

### データベース接続プールの使用を考慮する状況

各自の展開において、次のいずれか 1 つ以上が当てはまる場合にのみ、データベース接続プールの実装を検討します。

- 非プールデータベース接続を使用した場合に、RDBMS が専用ユーザー接続の必要数をサポートできない。デフォルトデータベース接続を共有で使用するためにプールすると、必要な接続数を減らすことができます。
- AOM を実行する Siebel Server マシンのメモリリソースが不足している。デフォルトデータベース接続を共有で使用するためにプールすると、同時ユーザーあたりの AOM メモリの必要量を減らすことができます。
- データベース認証以外の認証である LDAP などの外部認証を使用する展開で、データベースサーバーでの接続の新規作成が低速になる。データベース接続のプールは、接続が共有されているかどうかにかかわらず、永続プールによってログインなどの操作を高速化できます。

- 特殊な処理のために、頻繁なログインを必要とする Siebel Server コンポーネントを使用する。データベース接続を共有せずに永続的にプールすると、このようなコンポーネントには大きな効果があります。
  - Siebel Configurator で Siebel Product Configuration Object Manager コンポーネント(日本語版のエイリアス eProdCfgObjMgr\_jpn)を使用する場合、永続的な接続プールを設定することを強くお勧めします。Siebel Configurator については、第 8 章「パフォーマンスのための Siebel Configurator のチューニング」を参照してください。

**備考:** Siebel アプリケーションを実行できるインストール済み言語ごとに、個別の Object Manager が提供されています。たとえば、フランス語版の Call Center Object Manager は SCCObjMgr\_fra です。
  - セッションレスモードで実行する EAI Object Manager などの他の一部のコンポーネントでも、永続的な接続プールを設定すると有効な場合があります。

**備考:** コンポーネントに ModelCacheMax パラメータを設定してセッションキャッシュを設定する場合に、永続的な接続プールを設定すると多少の効果が期待できます。たとえば、セッションキャッシュは一般に Workflow Process Manager に設定します。Siebel ワークフローのセッションキャッシュについては、96 ページの「セッションのキャッシュ」を参照してください。

## データベース接続プール使用のガイドライン

データベース接続プールを使用する場合は、次のガイドラインに従ってください。

- MaxTasks/MaxSharedDbConns の値は、2:1 などの低い比率から開始します。

**備考:** 3:1 より高い比率を使用する場合は、Siebel エキスパートサービスに問い合わせてください。
- ユーザーシナリオの平均待ち時間が短い(操作が速い)場合、MaxTasks/MaxSharedDbConns の比率を小さくします。待ち時間が長い場合は、より大きな比率をサポートできます。

待ち時間が 30 秒の場合、10:1 より大きい比率は使用しないでください。待ち時間が 15 秒の場合、5:1 より大きい比率は使用しないでください。この項で説明するその他の要素によっても、実用的な限界値が決まります。

待ち時間については、26 ページの「AOM の展開のパフォーマンスに影響する要因」を参照してください。
- 時間のかかるクエリーを最小限に抑えます。データベース接続は、一度に 1 つのデータベース操作しか処理できません。したがって、ユーザーセッションにおいてデータベース接続プールで使用可能なすべての接続を使用して時間のかかるクエリーを実行すると、データベース接続を要求する新しいユーザーセッションは、データベース接続が利用可能になるまでブロックされます。

たとえば、3 秒かかるクエリーが実行されると、同じデータベース接続を必要とするユーザーは、クエリー操作が完了するまでキューに置かれてブロックされます。

時間のかかるクエリーは、それを開始したユーザーがブラウザを終了した場合でも RDBMS で実行が続行されます。

データベース接続プールを使用する場合、各自の環境でデータベースアクセスを最適化することが重要です。時間のかかるクエリーの実行を回避できない場合、データベース全体の応答時間を監視し、MaxTasks/MaxSharedDbConns の比率を小さくして、満足できる応答時間を確保してください。

時間のかかるクエリーは、エンドユーザーが並べ替え可能またはクエリー可能なフィールドがインデックスに含まれるように調整するか、インデックス処理されていないフィールドを表示しないようにアプリケーションユーザーインターフェイスを設定するか、または時間のかかるクエリーを実行する操作を避けるようにユーザーを指導することによって、最小限に抑えることができます。インデックス処理の Siebel アプリケーションのパフォーマンスへの影響については、[183 ページの「ソートおよび検索時のデータベースインデックスの管理」](#)を参照してください。

- データベース接続作成のコストを考慮します。このコストは認証方法によって異なります。

展開でデータベース認証を使用する場合、認証目的のために各ログインでデータベース接続が作成されます。その後、接続の総数が限界値より少ない場合、この接続は共有接続プールに解放されます。プールが一杯の場合、接続は閉じられます（終了します）。したがって、プールが一杯であり、接続が使用可能な場合であっても、依然として新たなセッションログインが行われるたびに新しい接続が一時的に作成されます。このような接続について、データベース接続の割り当てを決定するときに考慮する必要があります。

ただし、外部認証では、永続的な接続プールを使用してデータベース接続作成のコストを減らすことができます。永続的な接続プールを使用して作成されたデータベース接続は、共有される場合もされない場合も永続的です。永続的だが共有されていないプールされたデフォルトデータベース接続では、MaxSharedDbConns を MaxTasks - 1 に設定します。

認証オプションについては、『Siebel セキュリティガイド』を参照してください。

- 専用データベース接続の接続プールを設定するには、次のようにデフォルトデータベース接続のプールを設定する必要があります。
  - 接続プールを共有データベース接続（MaxSharedDbConns が -1 または 0）として設定しなければ、作成されたそれぞれの専用データベース接続は、個々のユーザーセッション専用になります。MinTrxDBConns の値は無視されます。
  - 共有データベース接続プールを設定する場合（MaxSharedDbConns の値は 0 より大きく、MaxTasks より小さい）、専用データベース接続はユーザーセッション専用にはなりません。この種の接続は、MinTrxDBConns の設定に基づいて処理されます。
    - MinTrxDBConns が -1 または 0 の場合、その接続を必要とするトランザクションが終了すると、それぞれの専用データベース接続は閉じられます（削除されます）。
    - MinTrxDBConns の値が 0 より大きい場合、その接続を必要とするトランザクションが終了すると、それぞれの専用データベース接続は接続プールに戻されます。
- Siebel データベース接続プールを MTS または Oracle 10g の多重化機能と同時に使用することはできません。

## デフォルトデータベース接続のプールの設定

デフォルトデータベース接続は、ほとんどの AOM 処理で使用されます。

### デフォルト接続をプールするためのパラメータの設定

この項では、MaxSharedDbConns パラメータ (DB Multiplex - Max Number of Shared DB Connections) と MinSharedDbConns パラメータ (DB Multiplex - Min Number of Shared DB Connections) を使用してデフォルトデータベース接続のプールの有効/無効を切り替える方法を説明します。

- 接続プールを有効にするには、MaxSharedDbConns パラメータと MinSharedDbConns パラメータを MaxTasks - 1 未満で、1 以上の正の整数値に設定します。マルチスレッドプロセス内のセッション数が 1 プロセスあたりに許容される共有接続の最大数を超えると、複数のユーザーセッションにより接続が共有されます。
  - MaxSharedDbConns は、各マルチスレッドプロセスでプールされているデータベース接続の最大数を設定します。
  - MinSharedDbConns は、AOM でマルチスレッドプロセスごとに維持することを試みる、使用可能なプールされているデータベース接続の最小数を設定します。

MinSharedDbConns の設定は、MaxSharedDbConns の設定以下であることが必要です。AOM の使用パターンに応じて、これらのパラメータを同じ値に設定することもできますし、データベース接続リソースの節約に役立つ場合であれば MinSharedDbConns により小さな値を設定することも可能です。
- 永続的で共有されたデータベース接続プールを設定するには、適切な MaxTasks/MaxSharedDbConns の比率を使用して MaxSharedDbConns を設定します。この比率によって、共有の有効性が決まります。MaxTasks/MaxSharedDbConns の比率は、2:1 (またはそれ未満) から開始します。この比率の例では、2 つのユーザータスクが同じデータベース接続を共有します。
- 永続的な共有されていないデータベース接続プールを設定するには、MaxSharedDbConns を MaxTasks - 1 に設定します。
- 接続プールを無効にする場合は、MaxSharedDbConns および MinSharedDbConns を -1 (デフォルト値) に設定します。

MaxSharedDbConns と MinSharedDbConns はエンタープライズベースで AOM コンポーネントごとに定義されます。このパラメータは、InfraDatasources タイプの指定サブシステムに含まれます。これらのパラメータで制御するデータベース接続は、マルチスレッドプロセス間で共有されません。個々のマルチスレッドプロセスでデータベース接続の実際の最大数は、比率 MaxSharedDbConns/MaxMTServers によって決まります。

**備考：** MaxSharedDbConns と MinSharedDbConns の動作は、各マルチスレッドプロセスに対して共有される専用データベース接続の数を指定する MinTrxDBConns とは異なります。詳しくは、43 ページの「専用データベース接続のプールの設定」を参照してください。

## デフォルト接続をプールするための設定例

たとえば、次のようなパラメータの設定を考えます。

```
MaxTasks = 500
MaxMTServers = 5
MinMTServers = 5
MaxSharedDbConns = 250
MinSharedDbConns = 250
```

この設定で、AOM コンポーネントは最大 500 タスク（スレッド）をサポートできます。この 500 タスクは、5 つのマルチスレッドプロセスで 100 タスクずつに分散されます。各マルチスレッドプロセスは、最大で 50 の共有データベース接続を持ち、それぞれが最大 2 タスクに対応します。

## プールされたデフォルト接続を割り当てる方法

ユーザーが AOM にログインすると、ユーザー認証のためのデータベース接続が確立され、データベースまたは外部認証システムがユーザーを認証すると、そのデータベース接続は破棄されます（閉じられます）。正常に認証された後に、AOM の接続マネージャは SQL 文の接続プールを確認します。この接続プールが空の場合、接続マネージャは接続を追加します。

ユーザーが SQL 文を初期化するたびに、接続マネージャは接続プールに使用できる接続があるかどうかを確認します。接続マネージャは、次のいずれかの方法を使用して、SQL 文のための接続を予約します。

- SQL 文の処理に使用できる接続がある場合、接続マネージャはその接続を SQL 文の実行に割り当てます。

接続マネージャは、この接続を SQL 文の実行が終わるまで予約します。SQL 文が終了すると、接続マネージャは接続を解放します。ユーザーのログアウトまたはセッションタイムアウトによってユーザーセッションが終了すると、接続マネージャはユーザーセッションが使用した接続を確認します。この接続を他のユーザーセッションが参照していない場合、およびデータベース接続プールで使用できる接続数が MinSharedDbConns に指定された値を超えている場合、接続マネージャはこの接続を閉じてデータベース接続プールから解放します。

- 接続プールに使用できる接続がない場合、接続マネージャは新しい接続を作成して SQL 文の実行に割り当てます。接続マネージャは、接続プールの接続数が MaxSharedDBConns と等しくなるまで接続を追加します。

## 専用データベース接続のプールの設定

共有されない専用データベース接続は、複数の AOM 処理にわたるトランザクションが必要になる Siebel EAI など、特殊な Siebel コンポーネントによって主に使用されます。専用接続は、BEGIN TRANSACTION および END TRANSACTION による処理で使用されます。

### 専用接続をプールするためのパラメータの設定

ここでは、MinTrxDBConns (DB Multiplex - Min Number of Dedicated DB Connections) パラメータを使用して専用接続プールを有効または無効にする方法を説明します。

- MinTrxDBConns は、各マルチスレッドプロセスの専用データベース接続の最小数を設定します。接続は、必要になるまで作成されません。この最小数は、プールのすべての接続が作成された後の専用接続プールの最小サイズです。
  - 専用接続プールを有効にするには、MinTrxDBConns に 1 以上の正の整数値を設定します。デフォルトデータベース接続のプールも設定する必要があります。
  - 専用接続プールを無効にするには、MinTrxDBConns に -1 (デフォルト値) を設定します。
- 専用接続の最大数には明示的な制限がありません。しかし、セッションよりも多い専用接続が存在することは事実上不可能です。平均的には、接続の数はセッションよりもかなり少数です。

MinTrxDBConns はエンタープライズベースで AOM コンポーネントごとに定義されます。このパラメータは、InfraDatasources タイプの指定サブシステムに含まれます。このパラメータで制御するデータベース接続は、マルチスレッドプロセス間で共有されません。個々のマルチスレッドプロセスに対する専用データベース接続の実際の最小数は、MinTrxDBConns の値として指定します。

**備考：** MinTrxDBConns の動作は、MaxSharedDbConns および MinSharedDbConns とは異なります。MaxSharedDbConns と MinSharedDbConns は、すべての AOM プロセスで使用できる共有データベース接続の数を指定し、MinTrxDBConns は、AOM プロセスあたりの専用データベース接続の数を指定します。詳しくは、41 ページの「デフォルトデータベース接続のプールの設定」を参照してください。

## 専用接続をプールするための設定例

たとえば、42 ページの「デフォルト接続をプールするための設定例」で示したパラメータの設定に加えて、次のパラメータの設定を仮定します。

```
MinTrxDBConns = 5
```

この設定では、各マルチスレッドプロセスが少なくとも 5 つの専用データベース接続を持ちます。したがって、AOM 上で 5 つのマルチスレッドプロセスすべてが実行中である場合、AOM 用に少なくとも 25 の専用接続が存在することになります。

## プールされた専用接続を割り当てる方法

AOM が起動するとき、専用接続プールは空です。トランザクションを開始するためのリクエストが行われると、AOM が専用接続プールのデータベース接続を要求します。接続が使用可能な場合、その接続がプールから削除され、排他使用のためにセッションに割り当てられます。

コミットまたはキャンセルによりトランザクションが完了すると、セッションからプールに専用接続が返されます。MinTrxDBConns で指定した数を超える接続がプールにすでに含まれている場合は、専用接続が閉じます。プールの接続がこの数以下である場合は、接続がプール内に維持されます。

## プールされた専用接続を割り当てるシナリオ

たとえば、MinTrxDBConns を 2 に設定した場合を考えます。この場合、専用接続は次のように処理されます。

- ユーザー 1 がトランザクション 1 を開始します。専用接続プールは空であるため、新しい接続が作成されます。トランザクション 1 が完了すると、この接続がプールに返されます。
- ユーザー 2 がトランザクション 2 を開始します。トランザクション 1 が依然として実行中である場合は、新しい専用接続が作成されます。トランザクション 1 が完了している場合は、トランザクション 2 が最初のデータベース接続を使用します。
- 2 つの専用接続が作成された場合は、AOM が終了するまで 2 つの接続がプール内に維持されます。

## AOM のスレッドプールの使用

AOM コンポーネントでは、オプションでスレッドプールを使用するように設定できます。次に説明するように AOM スレッドプールを有効にすると、スレッド間で複数のタスクがプールおよび多重化（共有）されます。

AOM スレッドプールを使用することにより、大きい負荷がかかっている複数 CPU のマシン、たとえば、8 基以上の CPU を搭載し CPU 能力の 75% を超えて実行中のマシン上でパフォーマンスとスケーラビリティを向上させることができます。

**備考：** AOM スレッドプールは、小規模のサーバーマシン、または比較的小さな能力で実行するマシンではお勧めできません。

### AOM のスレッドプールについて

AOM のマルチスレッドプロセスごとのプールサイズは、パラメータ UseThreadPool、ThreadAffinity、MinPoolThreads、および MaxPoolThreads の設定の組み合わせにより決定します。

AOM スレッドプールにより、ユーザーのログインまたはログアウト時、あるいはタイムアウト時のセッションスレッドの作成およびクローズで投入されるシステムリソースの使用量が削減されます。スレッドプールを使用していない場合は、セッションリクエストの要求に伴い必要に応じてセッションスレッドが作成されます。しかしスレッドプールを使用した場合は、セッションの終了時にセッションスレッドがクローズされるのではなく、プール内に解放されるので、以降のセッションで使用できます。

**備考：** ただし、スレッドプールを使用すると、タスク内のコンテキストの切り換えなど、独自のオーバーヘッドが発生します。このような理由から、スレッドを多重化しないでスレッドのプールを行うこと（つまり、ThreadAffinity = TRUE であるにもかかわらず、UseThreadPool = TRUE に設定すること）を避けるように強くお勧めします。

ThreadAffinity = FALSE であるので、ある種のデータベース接続や SISNAPI 接続で可能なように、スレッドが多重化されます。各スレッドは、任意のある時点で 1 つ以上のユーザーセッション（タスク）に割り当てられることがあります。

### AOM スレッドプールの設定

スレッドプールを使用するには、AOM 上で次のパラメータを設定します。

- UseThreadPool = TRUE（デフォルトは FALSE）
- ThreadAffinity = FALSE（デフォルトは FALSE）
- MinPoolThreads = *min\_number\_threads\_in\_pool*（デフォルトは 0）  
*min\_number\_threads\_in\_pool* は、AOM スレッドプール内のスレッドの最小数を表します。
- MaxPoolThreads = MinPoolThreads（デフォルトは 0）

**備考：** MaxPoolThreads には、MinPoolThreads 以上の値を指定する必要があります。この要件を除けば、どのような値を指定してもかまいません。

MinPoolThreads および MaxPoolThreads の適切な値を決定するには、まず穏やかな設定で開始した後、システムパフォーマンスを監視しながら、展開に応じて多重化をより大きくします。たとえば、まず次のような式から開始します（1 スレッドあたり 2 タスクに基づく式です）。

$$\text{MinPoolThreads} = \text{MaxPoolThreads} = (\text{MaxTasks}/\text{MaxMTServers})/2$$

続いて、次の式を使って、1 スレッドあたり 5 タスクに引き上げます。

$$\text{MinPoolThreads} = \text{MaxPoolThreads} = (\text{MaxTasks}/\text{MaxMTServers})/5$$

たとえば、MaxTasks = 525 および MaxMTServers = 5 の場合、次のようになります。

$$\text{MinPoolThreads} = \text{MaxPoolThreads} = (525/5)/5 = 105/5 = 21$$

また、MaxTasks = 725 および MaxMTServers = 5 の場合は、次のようになります。

$$\text{MinPoolThreads} = \text{MaxPoolThreads} = (725/5)/5 = 145/5 = 29$$

**備考：** 必要に応じて、待ち時間の値を調整します。待ち時間の値を半分にする場合は、プール内のスレッド数を 2 倍にします。

# 4

## パフォーマンスのための Siebel Server インフラストラクチャのチューニング

この章では、Siebel Application Object Manager (AOM) コンポーネントの構造とオペレーション、およびオペレーションの最適化のために必要になるチューニングについて説明します。ここで説明する内容は、次のとおりです。

- 47 ページの「AOM の SISNAPI 接続プールの設定」
- 49 ページの「Server Request Broker (SRBroker) のチューニング」

Siebel Server および AOM インフラストラクチャについて、および Siebel Web Client については、『Siebel Bookshelf』の次のマニュアルを参照してください。

- 『Siebel Deployment Planning Guide』
- 『Siebel システム管理ガイド』
- 使用しているオペレーティングシステム用の『Siebel インストールガイド』
- 『Siebel システム管理ガイド』

### AOM の SISNAPI 接続プールの設定

ここでは、Siebel Server の SISNAPI 接続を管理する方法について説明します。

TCP/IP プロトコル上で動作する SISNAPI (Siebel Internet Session アプリケーションプログラミングインターフェイス) は、AOM と Siebel Web Server Extension (SWSE) 間のネットワーク通信で使用されるメッセージ形式であり、Web サーバーにインストールされます。SISNAPI 接続は、Secure Sockets Layer (SSL) に基づく暗号化と認証を使用するために設定できます。

AOM コンポーネントの各マルチスレッドプロセスは、SWSE によって管理される SISNAPI 接続プールを使用します。このプロセスにより、各接続上で多数のクライアントセッションが多重化 (共有) されます。

各クライアントセッションリクエストにより、SessPerSisnConn パラメータによって定義された一連の接続が作成されるまで、新しい接続が開かれプールに追加されます。次に、既存のプール接続上で以降のリクエストが多重化されます。SISNAPI 接続は、次のいずれかのイベントが発生するまで維持されます。

- Web サーバープロセスが終了する
- AOM コンポーネントが終了する
- パラメータ SISNAPI Connection Maximum Idle Time (エイリアス ConnIdleTime) の値に到達する  
このパラメータについて詳しくは、『Siebel システム管理ガイド』を参照してください。
- ファイアウォールによって接続がタイムアウトになる

SISNAPI 接続セット上でトラフィックを多重化することにより、オープンネットワーク接続数を減らすことができます。

AOM のマルチスレッドプロセスごとの SISNAPI 接続プールサイズは、MaxTasks、MaxMTServers、および SessPerSisnConn の設定の組み合わせにより決定します。

SessPerSisnConn は、単一 SISNAPI 接続上で多重化できるセッション数を指定します。AOM コンポーネントの SessPerSisnConn は、デフォルトで 20 に設定されます。このデフォルト値は、大半の展開で適切であり、通常は変更する必要がありません。この設定は、待ち時間の計算に応じて異なる場合があります。詳しくは、[26 ページの「AOM の展開のパフォーマンスに影響する要因」](#)を参照してください。

MaxTasks および MaxMTServers の設定については、[29 ページの「CPU およびメモリ使用量のために AOM コンポーネントをチューニングする」](#)を参照してください。

実際には、AOM のマルチスレッドプロセスごとの SISNAPI 接続数は次の式によって決定します。

$$(\text{MaxTasks}/\text{MaxMTServers})/\text{SessPerSisnConn} = \text{SISNAPI\_conn\_per\_process}$$

ここで、SISNAPI\_conn\_per\_process は、マルチスレッドプロセス 1 個あたりの SISNAPI 接続数を表します。

たとえば、次のようなパラメータ値を考えます。

$$\text{MaxTasks} = 600$$

$$\text{MaxMTServers} = 5$$

$$\text{SessPerSisnConn} = 20$$

この場合、式は次のようになります。

$$(600/5)/20 = 120/20 = 6$$

また、MaxTasks = 540 で SessPerSisnConn を 18 に設定した場合は、式は次のようになります。

$$(540/5)/18 = 108/18 = 6$$

いずれの場合でも、各 SWSE からの AOM マルチスレッドプロセスごとに 6 つの SISNAPI 接続が作成およびプールされます。各 Web サーバーと SWSE にはそれぞれ 6 つの接続セットが存在する可能性があるため、AOM プロセスに対する接続の最大総数は、Web サーバー数の 6 倍です。前者の例では、接続ごとに 20 セッションが多重化されます。後者の例では、接続ごとに 18 セッションが多重化されます。

**備考：** 一般に、前述の式に使用する数値は、結果が整数になるように調整することをお勧めします。そのためには、MaxTasks、MaxMTServers、および SessPerSisnConn を定義する方法を変更する必要があります。

一部の Object Manager コンポーネントは、AOM コンポーネントではありません。このようなコンポーネントの設定は、AOM コンポーネントでの適切な設定とは異なる場合があります。EAI Object Manager については、[第 9 章「パフォーマンスのための Siebel EAI のチューニング」](#)を参照してください。

## Server Request Broker (SRBroker) のチューニング

Server Request Broker (SRBroker) コンポーネントは、AOM からバッチコンポーネントにリクエストをルーティングするなど、Siebel Server コンポーネント間でリクエストをルーティングします。SRBroker は、バッチコンポーネント間のリクエストも処理します。コンポーネントが同じマシン上で実行しているか異なるマシン上で実行しているかを問わず、SRBroker が使用されます。

AOM コンポーネントから発行されたサーバーリクエストは常に SRBroker コンポーネントに送られ、リクエストに伴って実行すべき処理が次の要領で決定されます。

- 宛先コンポーネントが同じ Siebel Server 上で実行している場合、SRBroker はそのコンポーネントにリクエストを渡します。宛先コンポーネントのインスタンスが複数実行中の場合、SRBroker は各コンポーネントインスタンスにラウンドロビン方式でリクエストを渡します。
- 宛先コンポーネントが同じ Siebel Server 上では実行していない場合、SRBroker は別のマシン上で実行する SRBroker にリクエストを渡します。宛先コンポーネントが複数の Siebel Server 上で実行している場合、SRBroker は各サーバーにラウンドロビン方式でリクエストを渡します。

SRBroker のデフォルトパラメータ値は、ほとんどの展開で適切に機能します。必要に応じて、MaxTasks パラメータ値を調整します (デフォルト値は 100)。MaxTasks は、Siebel Server 上で実行可能な SRBroker スレッド (タスク) の最大数を決定します。必要に応じて、MaxTasks の値を、Siebel Server で実行するバッチコンポーネントの数と、エンタープライズ内の Siebel Server の数の和に、10 (オーバーヘッド用) を加えた値に設定します。

MaxMTServers および MinMTServers は、Siebel Server 上で実行可能な SRBroker のマルチスレッドプロセスの最大数と最小数を決定します。各マルチスレッドプロセスは、最大で MaxTasks/MaxMTServers 個のスレッドを実行できます。MaxMTServers と MinMTServers は、デフォルト値の 1 にしておく必要があります。この値を大きくしてもパフォーマンスは向上せず、メリットはありません。

**注意:** SRBroker コンポーネントの MaxTasks パラメータの値に前述の要件を満たさない設定を行うと、リクエストが失敗する場合があります。バッチコンポーネントに送信されたリクエストの処理方法の詳細については、以下のセクションの HonorMaxTasks パラメータの説明を参照してください。HonorMaxTasks は、SRBroker コンポーネントまたは Server Request Processor (SRProc) コンポーネントに設定しても効果はありません。

SRBroker コンポーネントと SRProc コンポーネントについては、『Siebel システム管理ガイド』を参照してください。

## バッチコンポーネントの HonorMaxTasks パラメータについて

Workflow Process Manager などのバッチコンポーネントの HonorMaxTasks パラメータは、デフォルトで「FALSE」に設定されています（この設定をお勧めします）。この設定では、最大タスク容量に到達したバッチコンポーネントに SRBroker によってルーティングされたリクエストは、メモリ内のキューに置かれ、タスクが処理可能になった時点で処理されます。このようなタスクをキューに置くと、MaxTasks 値に到達したリクエストがバッチコンポーネントで失敗する可能性を最小に抑えることができます。

次のシナリオでは、バッチコンポーネントの HonorMaxTasks を「TRUE」に設定することを考えます。

- 非同期リクエストを処理するバッチコンポーネントでは、コンポーネントを実行するサーバーのリソースレベルが異なるために、コンポーネントの MaxTasks の設定値が異なる場合は、HonorMaxTasks を「TRUE」に変更することを検討します。その場合、設定値の大きなサーバーがより多くのリクエストを処理することになります。ただし、コンポーネントが最大タスク容量で実行されていない場合の効果は、明確に確認できないことがあります。
- バッチコンポーネントにクラッシュまたはハングの問題が発生する場合は、コンポーネントメモリのキューにリクエストを置かないことをお勧めします。HonorMaxTasks が「TRUE」の場合、各リクエストの成功または失敗の状態は正しくレポートされます。このオプションの方法は、一時的な手段としてのみ使用します。コンポーネントのクラッシュまたはハングの問題が発生する場合は、Siebel 技術サービスに問い合わせ解決してください。

96 ページの「パフォーマンスのための Workflow Process Manager のチューニング」も参照してください。

# 5

## パフォーマンスのための Siebel Web クライアントのチューニング

この章では、Siebel Web クライアントのパフォーマンスおよびスループットに影響する設定オプションについて説明するとともに、最適なパフォーマンスとスケーラビリティを実現、維持するためのクライアントのチューニングに関するガイドラインを示します。次のトピックが含まれています。

- 52 ページの「Siebel クライアントについて」
- 53 ページの「Siebel Web クライアントのパフォーマンスに影響する要因」
- 54 ページの「Siebel Web クライアントのチューニングを行うための最善の方法」

詳しくは、『Siebel Bookshelf』の次のマニュアルを参照してください。

- 『Siebel Deployment Planning Guide』
- 使用しているオペレーティングシステム用の『Siebel インストールガイド』
- 『Siebel システム管理ガイド』
- 『Siebel セキュリティガイド』
- Siebel SupportWeb 上の「System Requirements and Supported Platforms & Miscellaneous Documentation」

このマニュアルの次の項は、Siebel Web クライアントのパフォーマンスにも関係します。

- Application Object Manager (AOM) のパフォーマンスに関する注意事項については、[第 3 章「パフォーマンスのための Siebel Application Object Manager のチューニング」](#)を参照してください。
- Siebel アプリケーションの設定に関連するパフォーマンスの注意事項については、[第 12 章「パフォーマンスのための顧客設定のチューニング」](#)を参照してください。

## Siebel クライアントについて

Siebel クライアントは、Siebel Business Applications を実行し、1 つまたは複数のサーバーを通じてデータやサービスにアクセスするコンピュータです。Siebel クライアントを使用すると、Siebel アプリケーションによって管理される情報にアクセスできます。Siebel のすべての展開には、1 つまたは複数のタイプの Siebel クライアントが含まれます。クライアントは組み合わせて展開できます。

このマニュアルで取り上げる Siebel Business Applications のクライアントタイプは、Siebel Web クライアントです。このクライアントは、エンドユーザーのクライアントコンピュータ上で標準のサードパーティ製ブラウザ内で実行されます。他の常駐ソフトウェアをインストールする必要はありません。

ブラウザは、HTTP を使って WAN、LAN、または VPN、あるいはインターネット経由で Web サーバーに接続します。Siebel クライアントは Web サーバー経由で Siebel Server 上の Application Object Manager (AOM) コンポーネントに接続します。さらにこの AOM コンポーネントが Siebel アプリケーションビジネスロジックを実行し、データにアクセスします。Siebel データベースのデータがアクセスされます。また、仮想ビジネスコンポーネントおよびさまざまな統合メソッドを通じて他のデータソースのデータがアクセスされる場合もあります。

Siebel Business Applications アーキテクチャのユーザーインターフェイス層だけがクライアントのコンピュータにあります。

Siebel Web クライアントとその他のクライアントタイプについて、およびサポートされるブラウザとブラウザ設定については、使用しているオペレーティングシステム用の『Siebel インストールガイド』と、『Siebel システム管理ガイド』を参照してください。

## Siebel Web クライアントのパフォーマンスに影響する要因

Siebelアプリケーションのパフォーマンスに関する注意事項には、Siebelクライアントのパフォーマンスには影響せず、サーバー上での処理またはチューニングに関する活動だけが対象になるものがあります。しかしその他にも、直接または間接に Siebel クライアントのパフォーマンスに影響するこの種の要因が数多く存在します。この章では、Siebel Web クライアントのパフォーマンスに最も直接的に関係する要因をいくつか取り上げます。

Siebel クライアントのパフォーマンスは、さまざまな要因に依存します。次に、そのような要因のいくつかをまとめておきます。これらのトピックについては、『Siebel Bookshelf』の該当マニュアルまたは Siebel SupportWeb を参照してください。

### 複数の Siebel モジュールのサポートについて

従業員アプリケーションおよび顧客アプリケーションはさまざまな要件と属性を持ち、異なるブラウザやその他の関連技術を使用することがあります。

- Siebel Call Center などの従業員アプリケーションは、高い対話性モードを使用し、サポートされる Microsoft Internet Explorer ブラウザ内だけで実行可能です。
- Siebel eService や Siebel eSales などの顧客アプリケーションは、標準対話性モードを使用し、広範なブラウザおよびブラウザバージョン内で実行可能です。

すべての Siebel アプリケーションは、共通のアーキテクチャ要素を数多く持ちます。複数のアプリケーションが同じ Siebel リポジトリファイル (SRF) を使用できます。各アプリケーションは、独自の AOM コンポーネントを使用します。それぞれの使用のシナリオで要件が満たされることを確認するために、各アプリケーションの複数のインスタンスを定義、設定、およびテストすることが必要になる場合があります。

Siebelアプリケーションのパフォーマンスは、Siebel Toolsまたはカスタムブラウザスクリプトを使ったアプリケーションの設定に左右されます。57 ページの「[設定のガイドライン](#)」を参照してください。

クライアントのパフォーマンスもまた、展開する Siebel モジュールに応じて変化します。Siebel クライアントのパフォーマンスは、その使用機能に大きく依存します。したがって、Siebel モジュールのパフォーマンスの特性はさまざまです。

モジュールによっては、特殊な処理要件が追加されます。たとえば、Siebel CTI は Communications Session Manager (CommSessionMgr) コンポーネントを使用し、クライアント内でコミュニケーションツールバーと画面ポップアップの表示をサポートします。サーバーおよびローカルリソースはこの機能をサポートします。

国内または世界全体のオフィスに分散するユーザーをサポートする場合、パフォーマンスに影響する可能性がある特殊な展開要因を考慮することが必要になります。

### ローカルマシンのリソースについて

各ユーザーのローカルマシンで使用可能なリソースは、Siebel SupportWeb 上の「System Requirements and Supported Platforms & Miscellaneous Documentation」に記載されている推奨レベルを満たしているか、それを上回ることが必要です。パフォーマンス向上の基準のいくつかは、使用可能なリソースに直接依存します。

## Siebel Web クライアントのチューニングを行うための最善の方法

設定の変更を展開する前に、ビジネス要件が満たされるかどうか、および設計フェーズで想定したクライアントのパフォーマンスが得られるかどうかを確認するために、ハードウェアのリソースと要件を慎重に考慮する必要があります。

このマニュアル、特に第 12 章「パフォーマンスのための顧客設定のチューニング」、および『Siebel Bookshelf』の他の関連マニュアルに掲載されているガイドラインを確認してください。

時間経過に伴ってデータベース属性は変化します。システムのパフォーマンスを継続的にテストおよび監視することを強くお勧めします。

時間経過に伴って最適に実行するシステムを維持するには、展開済みのアプリケーションの拡張またはその他の変更を計画する必要があります。

次に、パフォーマンスおよびスケーラビリティに関する目標を達成するために考えられる活動をいくつか示します。

- システムトポロジを調整する
- Siebel Tools で Siebel アプリケーションを設定する
- Siebel Server コンポーネント、特に AOM を設定する
- ローカルマシン上で使用可能なハードウェアリソースを調整する
- サーバマシンまたはクライアントマシン上でオペレーティングシステム設定を調整する
- Web サーバーまたはネットワークの設定を調整する
- Web ブラウザの設定を調整する
- Siebel アプリケーションのユーザー設定を調整する

## 十分な処理容量の Web サーバーおよびネットワークの提供

Web サーバーが適切に設定されていて、パフォーマンスの要件が満たされていることを確認します。[58 ページの「静的なファイルキャッシュの指定」](#)も参照してください。

ネットワークの容量（帯域幅）がパフォーマンスの要件を満たすことを確認します。ネットワークの帯域幅を決定する場合には、次のような要因を考慮します。

- **アプリケーションの設定**：大規模で複雑なビューでは、より大きなテンプレート、より多くのコントロール、およびより大量のデータを Web サーバーからクライアントに送信することが必要になります。帯域幅が問題になる場合、1 つのビューの最適なサイズとレイアウトを決定するためにユーザーにより行われる操作を考慮することが大切です。

たとえば頻繁に使用されるビューでは、表示されるフィールドの数を削減します。高い対話性を持つクライアントの場合は、リストアプレット内で必要なカラムをユーザーが決定できます。特定のカラムセットを前提とするのではなく、ユーザーが必要に応じてカラムセットを調整できます。基本設定で必要なカラムの最小数を指定します。

詳しくは、[第 12 章「パフォーマンスのための顧客設定のチューニング」](#)を参照してください。

- **ビューのレイアウトのキャッシュ**：High Interactivity モードでは、管理者がローカルにキャッシュされるビューの数を決定できます。ハードウェアでより多くのビューをキャッシュできる場合は、それに応じて値を調整します。

ビューがキャッシュされている場合、以降のアクセス時にデータを更新する必要がありますが、Web テンプレートを再ロードする必要はありません。これにより、全体的な使用感が大きく向上します。

詳しくは、[60 ページの「ビューのレイアウトのキャッシュを使ったパフォーマンスの向上」](#)を参照してください。

- **ログイン**：高い対話性を持つクライアントでは一般に、最初のログインが最も負荷のかかる処理です。クライアントインフラストラクチャは、最初のログイン時にアプリケーションのメインコンポーネントをキャッシュします。このため、以降のログインで必要になるリソースははるかに少なくなります。キャッシュされたオブジェクトは、キャッシュをオフにするか、または新しいバージョンのアプリケーション設定が使用可能になるまで、クライアントコンピュータ上に留まります。

## Web クライアントのパフォーマンスのテスト

Siebel エキスパートサービスは、設定済みの Siebel アプリケーションの属性に関する既知の情報に基づいて、一般的なガイダンスを提供します。しかし、一般的なデータに基づくことを前提としているため、顧客テストを実施することをお勧めします。実際の状況は、ユーザーにより行われる操作のシナリオに応じて異なる可能性があります。最も高い割合の活動から成る最も一般的なシナリオをいくつか選択します。使用される全体的な帯域幅を収集します。

ウォームビュー（すでにアクセスおよびキャッシュされたビュー）をテストします。このテストでは、対象アプリケーションが通常使用されること、ユーザーがログイン後にログオフして新しいセッションを開始するまで 1 回で 4～8 時間使用するアプリケーションであることが必要です。

低帯域幅の接続を共有する数人のユーザーによって必要とされる帯域幅を見積もる場合、ユーザーによって行われる操作のシナリオを慎重に考慮した上で、それに応じた計画を作成します。ネットワークのパフォーマンスに関する最悪のシナリオ（すべてのユーザーが同時に Enter キーを押したり、新しいビューにアクセスしたりした場合）について計画しないでください。おそらく、ネットワークを同時に使用するユーザーは、実際には少数です。

詳しくは、[第 14 章「Siebel ARM による Siebel アプリケーションのパフォーマンスの監視」](#)を参照してください。

## 十分なクライアントハードウェアリソースの提供

高い対話性を持つアプリケーションで最適なクライアントのパフォーマンスを得るには、エンドユーザー（一般的には従業員）に十分なハードウェアリソースを提供します。要件は、それぞれの展開に応じて異なる場合があります。

クライアントマシンで使用可能なメモリが増大するにつれて、キャッシュ可能なビューの数が増えます。詳しくは、次の資料を参照してください。

- 57 ページの「ブラウザキャッシュの管理」
- 58 ページの「静的なファイルキャッシュの指定」
- 60 ページの「ビューのレイアウトのキャッシュを使ったパフォーマンスの向上」

クライアントマシン上のプロセッサ（CPU）の速度は、Siebel アプリケーションのユーザーインターフェイスの描画速度に影響します。

Siebel Call Center などの従業員アプリケーションによって使用される高い対話性を持つクライアントで最高のパフォーマンスを得るために、Microsoft Internet Explorer の最新のサポートバージョンをテスト対象に含めることが一般的に推奨されます。多くの場合、最新バージョンには、フィックスおよびパフォーマンスの強化機能が含まれています。

Siebel eService などの顧客アプリケーションによって使用される標準的な対話性を持つクライアントで最高のパフォーマンスを得るには、サポートするブラウザ、プロセッサ速度、予想されるインターネット接続速度など、顧客環境の最小限の機能を決定する必要があります。顧客アプリケーションでは、広範な顧客環境をサポートする必要があります。したがって、一般的にはこのようなアプリケーションの複雑さを最小限に抑える必要があります。

Siebel クライアントハードウェアおよびその他のプラットフォームの要件と推奨事項については、Siebel SupportWeb 上の「System Requirements and Supported Platforms & Miscellaneous Documentation」を参照してください。

Siebel アプリケーションのブラウザ設定については、『Siebel システム管理ガイド』を参照してください。

## システムコンポーネントのチューニング

全体的なエンドユーザーのパフォーマンスは、Web サーバーと Siebel データベースサーバーのやり取りに加えて、クライアント上のあらゆる処理の影響を受けます。全体的なパフォーマンスを向上させる余地がないかどうかについて、該当するあらゆるエリアを調べます。

ほとんどの場合、Siebel Server コンポーネントに関連するパフォーマンスのチューニングを行うときは AOM に焦点を合わせる必要があります。詳しくは、第 3 章「パフォーマンスのための Siebel Application Object Manager のチューニング」を参照してください。

Siebel ARM を使って、Siebel インフラストラクチャを経由するトランザクションを監視します。長い時間と大量のリソースを必要とするエリアに注目し、チューニングの余地がないかどうかについてさらに調査します。

たとえば、カスタム設定を使用する場合に、データベースインスタンスが最適化されない複雑な SQL 文が意図に反して作成されることがあります。Siebel Tools で若干の設定を調整するか、データベースをチューニングすることにより、クライアントのパフォーマンスおよび Siebel Server 上のアプリケーションのスケラビリティがともに向上する可能性があります。

Siebel Tools については、第 14 章「Siebel ARM による Siebel アプリケーションのパフォーマンスの監視」を参照してください。

## 設定のガイドライン

Siebel クライアントで最高のパフォーマンスを得るには、すべての顧客設定のイニシアチブを慎重に評価する必要があります。設定を変更する場合は常に、設定それ自体のコストと、想定されるパフォーマンスへの効果の両面から正当化されることが必要です。

また、アプリケーションの管理タスクがアプリケーションのパフォーマンスに影響する場合もあるため、この点についても慎重に評価しなければなりません。

第 12 章「パフォーマンスのための顧客設定のチューニング」や『Siebel Bookshelf』の他のマニュアルに掲載されているガイドラインに従ってください。

## ブラウザキャッシュの管理

ユーザーが Siebel ビューへのログインまたはアクセスを行ったときのパフォーマンスを向上させる目的で、数タイプの Siebel アプリケーション要素がブラウザキャッシュ内に保存されます。

**備考：** パフォーマンスの測定時に、ビューのレイアウトのキャッシュや他タイプのキャッシュを考慮する必要があります。たとえば、Siebel ビューのレイアウトをキャッシュから取得した場合の方がビューのレイアウトがキャッシュされておらずシステムからビューのレイアウトを取得した場合よりもパフォーマンスに優れています。詳しくは、60 ページの「ビューのレイアウトのキャッシュを使ったパフォーマンスの向上」を参照してください。

キャッシュの使用量は、使用されるブラウザ、実行中のアプリケーション、およびアプリケーションの設定に応じて異なります。たとえば、高い対話性を持つアプリケーションは、標準的な対話性を持つアプリケーションよりもブラウザキャッシュを使用します。

高い対話性を持つアプリケーションでは、ブラウザを閉じているときを含めて、ブラウザキャッシュをオンにしておくことが一般的にユーザーに推奨されます。Microsoft Internet Explorer では次の設定が推奨されます。

- [ツール] > [インターネットオプション] の順に選択します。[詳細設定] タブをクリックします。[セキュリティ] オプションで、[ブラウザを閉じたとき、[Temporary Internet Files] フォルダを空にする] チェックボックスをオフにします。

**備考：** この設定を行わない場合、60 ページの「ビューのレイアウトのキャッシュを使ったパフォーマンスの向上」で説明する永続的なビューのレイアウトのキャッシュとプレロードが機能しません。

- [ツール] > [インターネットオプション] の順に選択します。[詳細設定] タブをクリックします。[セキュリティ] オプションで、[暗号化されたページをディスクに保存しない] チェックボックスをオフにします。

**備考：** この設定を行わない場合、暗号化されたビュー（SSL により暗号化されたビュー）に対して、60 ページの「ビューのレイアウトのキャッシュを使ったパフォーマンスの向上」で説明する永続的なビューのレイアウトのキャッシュとプレロードが機能しません。

- [ツール] > [インターネットオプション] の順に選択します。[全般] タブの [設定] をクリックします。[保存しているページの新しいバージョンの確認] で、[自動的に確認する] を選択します。

- またブラウザのキャッシュは、インターネット一時ファイルのフォルダのサイズにより影響を受けます。この設定には、[ツール] > [インターネットオプション] からアクセスできます。[全般] タブ、[設定] の順にクリックし、このフォルダで使用するディスク容量を指定します。

**備考：** インターネット一時ファイルのフォルダのサイズを 0 に設定すると、60 ページの「ビューのレイアウトのキャッシュを使ったパフォーマンスの向上」で説明する永続的なビューのレイアウトのキャッシュとプレロードが無効になります。

Siebel アプリケーションのブラウザ設定については、『Siebel システム管理ガイド』を参照してください。

またブラウザ内のキャッシュは、静的なファイルキャッシュを管理する Web サーバーの設定の影響も受けます。詳しくは、58 ページの「静的なファイルキャッシュの指定」を参照してください。

## 静的なファイルキャッシュの指定

また、ブラウザキャッシュの動作は、静的なファイルキャッシュの Web サーバー設定の影響を受けます。適切な設定を行うことにより、イメージファイル、JavaScript ファイル、スタイルシートファイルなどの更新されることがまれなファイルをブラウザ上でキャッシュできます。静的なファイルをキャッシュすることで、ネットワークの使用が抑制され、Siebel Web クライアントの応答時間が向上します。

Siebel Web テンプレートファイルのキャッシュについては、60 ページの「ビューのレイアウトのキャッシュを使ったパフォーマンスの向上」の永続的なビューのキャッシュに関する説明を参照してください。

一部の静的なファイルが実際には定期的に更新される場合があるので、静的なファイルの古いバージョンがキャッシュから提供される危険があります。したがって、コンテンツに適切な有効期限を指定しておく必要があります。一般的には、7 日間の失効期間がおそらく適切です。

静的なファイルの更新がまれである場合は、より長い期間を指定し、失効の頻度を減らすことができます。静的なファイルの更新がより頻繁である場合は、より短い期間を指定し、失効の頻度を増やすことができます。

Microsoft インターネットインフォメーションサービス (IIS)、IBM HTTP Server (IHS)、および Sun Java System Web Server 上で静的なファイルのキャッシュを設定する手順に従います。設定を有効にするには、Web サーバーを再起動する必要があります。詳しくは、サードパーティ製 Web サーバーのベンダーが提供するマニュアルを参照してください。

サポートされる Web サーバーとそのバージョンについては、Siebel SupportWeb 上の「System Requirements and Supported Platforms & Miscellaneous Documentation」を参照してください。

## Microsoft IIS の静的なファイルのキャッシュ

Microsoft IIS では、次の手順に従って静的なファイルのキャッシュとコンテンツの失効期間を指定します。

### Microsoft IIS 上で静的なファイルのキャッシュを指定するには

- 1 Web サーバマシン上で、[スタート] > [設定] > [コントロールパネル] > [管理ツール] を選択します。
- 2 Internet Service Manager を実行します。
- 3 Internet Service Manager で、[既定の Web サイト] を右クリックします。
- 4 [既定の Web サイトのプロパティ] で、[HTTP ヘッダー] タブをクリックします。
- 5 [コンテンツに有効期限を設定する] チェックボックスをオンにします。
- 6 [有効期間] を選択し、値 7 を指定するか (7 日後に静的なファイルが失効します)、展開に応じて別の適切な値を指定します。

## IBM HTTP Server の静的なファイルキャッシュ

IBM HTTP Server (IHS) では、次の手順に従って静的なファイルのキャッシュとコンテンツの失効期間を指定します。

### IBM HTTP Server 上で静的なファイルのキャッシュを設定するには

- 1 Web サーバマシン上で、編集のために httpd.conf ファイルを開きます。このファイルは、Web サーバのインストールディレクトリに含まれています。
- 2 次の行が含まれていること、コメントアウトされていないことを確認します。  
LoadModule expires\_module modules/mod\_expires.so
- 3 次の行が存在しない場合は、[ステップ 2](#) で示したファイルの行の下に次の行を追加します。7 日ではなく、展開に応じて適切な別の値を指定することもできます。

```
#####  
ExpiresActive On  
  
<IfModule mod_expires.c>  
ExpiresByType image/gif "access plus 7 days"  
ExpiresByType image/jpeg "access plus 7 days"  
ExpiresByType application/x-javascript "access plus 7 days"  
ExpiresByType text/css "access plus 7 days"  
</IfModule>  
  
#####
```

- 4 ファイルを保存します。

## Sun Java System Web Server の静的なファイルのキャッシュ

Sun Java System Web Server で、適切な手順に従って静的なファイルのキャッシュとコンテンツの失効期間を指定します。たとえば Sun Java System Web Server 6.0 では、次の手順に従います。

### Sun Java System Web Server 上で静的なファイルのキャッシュを設定するには

- 1 ブラウザから、Web サーバーの管理ページ（たとえば [http://web\\_server\\_name/8080](http://web_server_name/8080)）に接続します。
- 2 サーバーを選択し、[管理] をクリックします。
- 3 右上隅にある [クラスマネージャ] のリンクをクリックします。
- 4 最上段のタブから [コンテンツ管理] タブをクリックします。
- 5 左側のタブ領域で、[キャッシュコントロール指示] のリンクをクリックします。
- 6 [キャッシュコントロール応答指示] で、[年齢上限 (秒)] を選択し、7 日間のキャッシュの失効期間として 604800 (秒) を入力します。
- 7 [適用] をクリックして変更を適用します。

## ビューのレイアウトのキャッシュを使ったパフォーマンスの向上

ブラウザ内でビューのレイアウトのキャッシュ(レイアウトのキャッシュまたはビューのキャッシュとも呼ばれます)を使用すると、高い対話性を持つアプリケーションでビューのアクセスのパフォーマンスが向上します。ブラウザ上で次のキャッシュを行うことにより、Siebel アプリケーションセッション内でビューの描画速度が向上します。

- ビューの解釈時に使用される (テンプレートからの) Static HTML。
- コントロールの描画のためにクライアント上で生成される Dynamic HTML。

キャッシュを適切に設定することにより、Siebel クライアントセッションのパフォーマンスとネットワーク使用を最適化できます。キャッシュの動作は、[57 ページの「ブラウザキャッシュの管理」](#)で説明した考慮事項の影響を受けます。

Siebel Web クライアントでは、使用されるビューのレイアウトのキャッシュが 2 種類あります。これらのキャッシュタイプは連携して機能するもので、システムとして設定する必要があります。

- **ブラウザメモリ内でのキャッシュ**：詳しくは、[61 ページの「メモリ内でビューのレイアウトをキャッシュする」](#)を参照してください。
- **永続的なキャッシュ (ローカルディスク上のブラウザのキャッシュディレクトリ)**：詳しくは、[62 ページの「永続的なビューのレイアウトのキャッシュ」](#)を参照してください。

**備考**：ビューをキャッシュできるかどうかは、[65 ページの「ビューでレイアウトのキャッシュが可能であるかどうか確認する」](#)で説明する基本要件に左右されます。

## メモリ内でビューのレイアウトをキャッシュする

メモリ内でビューのレイアウトをキャッシュすることにより、ビューのレイアウトを保存するための HTML フレームが複数、ブラウザ上に作成されます。これらの HTML フレームの数がビューのキャッシュサイズを表します。あるビューを表示すると、そのビューのレイアウトを含む HTML フレームのサイズが使用可能なブラウザ領域のすべて (100%) を占有するように設定され、他のフレームが非表示になります (つまり、サイズが領域の 0% を占有するように設定されます)。

ビューのキャッシュサイズについては、62 ページの「[ビューのキャッシュサイズの設定](#)」を参照してください。

ビューのレイアウトのキャッシュで使用されるロジックは次のとおりです。

- ブラウザのメモリキャッシュ内ですでにレイアウトが使用可能であるビューにユーザーが移動すると、そのビューを含む HTML フレームが表示され、現在表示されているフレームが非表示になります。
- ブラウザのメモリキャッシュ内にレイアウトが含まれないビューにユーザーが移動すると、使用可能ないずれかの HTML フレームによりビューのレイアウトがメモリにロードされます。ビューのレイアウトは、可能な場合は永続的なキャッシュからロードされます。メモリ内のビューのレイアウトは、ビューのキャッシュサイズの設定に従ってキャッシュされます。
- 永続的なキャッシュ内にビューのレイアウトが現在保存されていない場合、サーバーからロードされます。またこのビューのレイアウトは、永続的なキャッシュにも保存されます。メモリ内のビューのレイアウトは、ビューのキャッシュサイズの設定に従ってキャッシュされます。

詳しくは、62 ページの「[永続的なビューのレイアウトのキャッシュ](#)」を参照してください。

**備考：** 高い対話性を持つフレームワークでは、Siebel アプリケーションユーザーインターフェイスのサーバーからの取得と、データベースレコードの取得を分離します。ビューに表示するデータベースレコードは、常にサーバーから取得します。

メモリキャッシュには、ユーザーが以前にアクセスし、ビューのキャッシュとして使用可能なビューのレイアウトが含まれています。ビューのキャッシュが一杯の状態では別のビューがアクセスされた場合は、最初にアクセスされたビューがキャッシュから削除されます。このように、メモリのキャッシュのコンテンツは LRU (Least Recently Used) 方式で管理されます。

ユーザーがビューに移動すると、HTML フレームがメモリにロードされます。起動ビュー (キャッシュに格納可能なビュー) をキャッシュするには、ユーザーがそのビューに 2 回アクセスする必要があります。つまり、別のビューにアクセスした後、そのビューに 2 回目のアクセスを行う必要があります。

**備考：** Siebel Call Center の Home Page View (WCC) などのホームページビューとして特に作成されたビューは、標準的な対話性を持つビューであり、キャッシュできません。

ビューのキャッシュフレームワークの設計では、たとえば以前にキャッシュされたビューをすべて削除するブラウザの更新機能を実行することでキャッシュされたビューを含むフレームが削除された場合に、ユーザーが次にアクセスしたキャッシュ可能なビューからレイアウトのキャッシュの再ロードが開始されます。

起動時に、最近アクセスされたビューのレイアウトがディスク上の永続的なブラウザキャッシュからメモリキャッシュにプレロードされる場合があります。この動作は、ViewPreloadSize パラメータを使って設定します。詳しくは、62 ページの「[永続的なビューのレイアウトのキャッシュ](#)」および 63 ページの「[キャッシュされたビューのメモリへのプレロード](#)」を参照してください。

## ビューのキャッシュサイズの設定

ブラウザのメモリキャッシュでは、次に説明するように、ビューのレイアウトのキャッシュサイズをユーザーごとにビューのキャッシュサイズのユーザー設定によって管理します。

**備考：** ビューのキャッシュサイズを 1 に設定すると、ビューのキャッシュがオフになります。64 ページの「ビューのレイアウトのキャッシュを無効にする」で説明するように、EnableViewCache パラメータを FALSE に設定しても同じ効果が得られます。

### ビューのレイアウトのキャッシュサイズを設定するには

- 1 アプリケーションレベルのメニューから、[表示] > [ユーザー設定] の順に選択します。
  - 2 [表示] ドロップダウンリストから、[行動] を選択します。
  - 3 [キャッシュサイズを表示] フィールドで、ドロップダウンリストから値を選択するか、または値を入力します。  
[キャッシュサイズを表示] のデフォルト値は 10 です。この場合は、Siebel ビューのレイアウトを表示するために 10 の HTML フレームをメモリ内にキャッシュすることを指定しています。ある時点でこれらのフレームの 1 つが表示されます。
- ユーザーが多数のビューにアクセスする場合で、クライアントマシンに十分なメモリが搭載されているときは、この値を低く設定しすぎると十分なキャッシュが行われな可能性がります。
  - またこの値を高く設定しすぎると、マシン上での使用メモリが大きくなりすぎることパフォーマンスが低下する可能性があります。

## 永続的なビューのレイアウトのキャッシュ

永続的なレイアウトのキャッシュでは、ローカルクライアントのディスク上のブラウザキャッシュにビューのレイアウトが保存されます。次に、同じセッションまたは以降のセッションで後ほどこのビューがアクセスされると、保存されたレイアウトが再使用されます（同じセッションで後ほどこのビューにアクセスされた場合は、可能であればブラウザのメモリキャッシュからビューのレイアウトにアクセスされます）。

永続的なビューのレイアウトのキャッシュにより、セッション間でサーバーから生成する必要があるページ数が減少し、パフォーマンスを向上させることができます。

WebTemplateVersion パラメータは、Siebel Web Engine がブラウザのキャッシュに保存されたビューのレイアウトを使用するか、または新しいビューのレイアウトを構築するかを指定します。このパラメータは、Siebel Call Center の uagent.cfg などのアプリケーション設定ファイルの [InfraUIFramework] セクションに含まれます。このファイルは、AOM を実行する Siebel サーバマシン上に存在します。

Siebel ビューの Web テンプレートを変更する場合、設定ファイル（存在する場合）に WebTemplateVersion パラメータを追加し、その値を 1 に設定します。次に例を示します。

```
[InfraUIFramework]
WebTemplateVersion = 1
```

以降、いずれかの Web テンプレートを変更するたびに、このパラメータの値が 1 ずつ増えます。これにより、サーバー上の Web テンプレートからビューのレイアウトが強制的にロードされます。

ビューが要求されると、Siebel Web Engine が WebTemplateVersion パラメータの値をカプセル化するチェックサム値を URL に含めます。

- パラメータ値と、URL にカプセル化された値が一致する場合、このビューのレイアウトが更新されていないものとみなされます。可能な場合は、永続的なキャッシュに保存されたビューのレイアウトを使用できます。
- 値の一致が検出されない場合、サーバーから新しいビューのレイアウトがロードされます。おそらく、サーバー上の Web テンプレートの方がブラウザの永続的なキャッシュに保存されたものよりも新しいビューのレイアウトです。

### キャッシュされたビューのメモリへのプレロード

最近アクセスされたビューでは、ユーザーがログインしたときにブラウザ上の永続的なキャッシュに格納されたビューのレイアウトがブラウザのメモリにプレロードされる場合があります。プレロード可能なビューの数は、永続的なキャッシュのコンテンツに応じて異なり、各ユーザーの [キャッシュサイズを表示] の設定により制限されます。

ログイン時のパフォーマンスを向上させるために、起動時にメモリにプレロードされるビューのレイアウトの数をさらに制限することが有効な場合があります。このためには、ViewPreloadSize パラメータを使用します。

**備考：** ViewPreloadSize がユーザーセッションに影響するのは、View Cache Size 値よりも小さい正の整数値を設定した場合に限られます。このパラメータを設定しない場合のデフォルトの動作は、View Cache Size の値から 1 引いた数のビューのレイアウトをプレロードすることです (View Cache Size で指定したフレームの 1 つは、必要な場合にアプリケーションの起動ビューのために予約されます)。

ViewPreloadSize は、Siebel Call Center の uagent.cfg などのアプリケーション設定ファイルの [InfraUIFramework] セクションに追加する必要があります。このファイルは、AOM を実行する Siebel サーバマシン上に存在します。たとえば、次のように設定します。

```
[InfraUIFramework]
ViewPreloadSize = 5
```

ViewPreloadSize を 0 に設定した場合、ビューのレイアウトはメモリにプレロードされません。ユーザーが単一のビューにアクセスした後、再びログアウトする場合など、Siebel アプリケーションにユーザーが頻繁にログインするシナリオでは、ログイン時のパフォーマンスが複数のビューのプレキャッシュよりも重要になります。このような状況では、ViewPreloadSize を 0 に設定した方が良いでしょう。

## ビューのレイアウトのキャッシュを無効にする

Siebel Call Center の uagent.cfg など、アプリケーション設定ファイルの [InfraUIFramework] セクションで EnableViewCache パラメータを「FALSE」に変更することにより、アプリケーションユーザーに対するビューのレイアウトのブラウザメモリキャッシュを無効にできます。たとえば、次のように設定します。

```
[InfraUIFramework]
EnableViewCache = FALSE
```

**備考：** 一般に、EnableViewCache を TRUE に設定することをお勧めします。一部のユーザーがビューのレイアウトのキャッシュを必要としない場合は、62 ページの「ビューのキャッシュサイズの設定」で説明したように View Cache Size を 1 に設定できます。

EnableViewCache を FALSE に設定すると、ブラウザのメモリによるビューのレイアウトのキャッシュだけが無効になります。永続的なビューのレイアウトのキャッシュは無効になりません。

## 現在のビューのレイアウトがロードされた方法の確認

アプリケーションの実行中に現在のビューがどのように取得されたかを確認するには、そのビューに移動し、Shift キーを押して、[ヘルプ] > [ビュー情報] を選択します。これにより、現在のビューの [キャッシュモード] に、アプリケーションでビューのレイアウトが取得された方法が示されます。次のような値が表示されます。

- **[キャッシュなし]**: ビューのレイアウトはキャッシュされませんでした (また、ビューのレイアウトはキャッシュできません)。
- **[メモリー]**: ビューのレイアウトはブラウザのメモリキャッシュから取得されました。
- **[サーバー]**: ビューのレイアウトは、Siebel Server および Web サーバーから取得されました。ビューがキャッシュ可能である場合で、別のビューにアクセスした後そのビューに戻ると、[キャッシュモード] の値が [メモリー] に変更されます。
- **[ディスク]**: ビューのレイアウトはブラウザのディスクキャッシュ (永続的なキャッシュ) から取得されました。ビューがキャッシュ可能である場合で、別のビューにアクセスした後そのビューに戻ると、[キャッシュモード] の値が [メモリー] に変更されます。

キャッシュをクリアしないで作業を続ける時間が長くなるほど、アクセスすることがまれなビューがサーバーからではなくブラウザの永続的なキャッシュから取得される可能性が高くなります。

## ビューでレイアウトのキャッシュが可能かどうか確認する

Siebel のすべてのビューでレイアウトのキャッシュが可能ではありません。データ依存の動的なレイアウトまたはコントロールを持つアプレットを含むビューは、キャッシュできません。高い対話性をサポートするアプレットだけがビューのレイアウトのキャッシュで使用できます。

レイアウトのキャッシュは、アプレットを実装する C++ クラスの機能です。キャッシュの可否は、各アプレットのクラスオブジェクト定義のプロパティによって決定されます。Siebel Tools を使って、クラスオブジェクト定義の High Interactivity Enabled プロパティの値を確認し、このクラスのアプレットでレイアウトのキャッシュがサポートされるかどうかを判別します。キャッシュが可能なビューでは、そのビュー内の全アプレットのクラスオブジェクトで High Interactivity Enabled 値に 2 または 4 (キャッシュ可能) を設定する必要があります。

クラスの High Interactivity Enabled プロパティの設定については、『Siebel Object Types Reference』を参照してください。

ビューのレイアウトのキャッシュはまた、次の状況で無効になります。

- いずれかのアプレットでパーソナライゼーションルールが定義されている場合
- いずれかのアプレットが動的なトグルアプレットである場合
- いずれかのアプレットが階層リストアプレットまたはエクスプローラ (ツリー) アプレットである場合
- ビューのテンプレート (たとえばエクスプローラビューなど) 内で HTML フレームが使用されている場合

## メッセージバーに関連するパフォーマンスの管理

Siebel Call Center などの従業員アプリケーションには、メッセージバー機能が含まれます。メッセージバーでは、表示テキストを継続的に更新するためにネットワークリソースとクライアントマシン上のローカルリソースが必要になります。

- 展開でメッセージバー機能が不要の場合は、メッセージバー機能をオフにすることで処理リソースを節約できます。
- メッセージバーを必要とするユーザーが一部である場合は、[ツール] > [ユーザー設定] > [メッセージの一括送信] からユーザーがこの機能をオフにできるように指定できます。

メッセージバーを使ったメッセージの一括送信については、『Siebel アプリケーション管理ガイド』を参照してください。

## 標準の対話性を持つアプリケーションのビジーカーソルの設定

パラメータ `EnableSIBusyCursor` が「TRUE」（デフォルト）に設定されている場合、Siebel アプリケーションは現在のリクエストを保存し終えるまで新しいリクエストを受け付けることはできません。新しいリクエストの例には、レコードのドリルダウンの試みがあります。

このパラメータのデフォルト設定により、ユーザーがページのロード中に別のリンクをクリックできないため、JavaScript エラーが発生するのを防ぐことができます。`EnableSIBusyCursor` を「FALSE」に設定すると、標準対話性モードを使用する Siebel アプリケーションのネットワーク帯域幅の利用状況を改善させることができます。砂時計アイコンの表示が無効になり、ユーザーは現在のリクエストが保存される前に、他のリンクをクリックできます。

アプリケーション設定ファイルの `[InfraUIFramework]` セクションの `EnableSIBusyCursor` の値を設定します。

# 6

## パフォーマンスのための Siebel Communications Server のチューニング

この章では、Siebel Communications Server およびその関連モジュールの選択した機能についてパフォーマンスとスループットに影響する問題を解説するとともに、このモジュールで最適なパフォーマンスとスケーラビリティを実現および維持するためのチューニングのガイドラインを示します。ここで説明する内容は、次のとおりです。

- 68 ページの「Siebel Communications Server の概要」
- 69 ページの「セッションコミュニケーションインフラストラクチャ」
- 71 ページの「セッションコミュニケーションのパフォーマンスに影響する要因」
- 72 ページの「セッションコミュニケーションのトポロジに関する注意事項」
- 73 ページの「Session Communications のチューニングを行うための最善の方法」
- 80 ページの「Siebel Email Response のインフラストラクチャ」
- 81 ページの「Siebel Email Response のパフォーマンスに影響する要因」
- 82 ページの「Siebel Email Response のトポロジに関する注意事項」
- 83 ページの「Siebel Email Response のチューニングを行うための最善の方法」

この章で取り上げる機能には、セッションコミュニケーション（一般的には Siebel CTI）および Siebel Email Response が含まれます。コミュニケーションに関連する他のモジュールについては取り上げません。

この章のトピックについては、『Siebel Bookshelf』の次のマニュアルを参照してください。

- 『Siebel Communications Server 管理ガイド』
- 『Siebel eMail Response 管理ガイド』
- 『Siebel システム管理ガイド』
- 『Siebel Smart Answer Administration Guide』

## Siebel Communications Server の概要

Siebel Communications Server は、Siebel アプリケーションユーザーのためのさまざまなコミュニケーション活動をサポートするインフラストラクチャを提供します。

セッションコミュニケーションのパフォーマンスのチューニングについては、[69 ページの「セッションコミュニケーションインフラストラクチャ」](#) およびそれ以降の項を参照してください。

Siebel Email Response のパフォーマンスのチューニングについては、[80 ページの「Siebel Email Response のインフラストラクチャ」](#) およびそれ以降の項を参照してください。

- **セッションコミュニケーション**：マルチチャネルのコミュニケーションツールバーを使用するコンタクトセンターエージェントにインタラクティブな（セッション）コミュニケーションをサポートします。
    - Siebel CTI Connect やサードパーティ製品などの CTI ミドルウェアによりサポートされるコンピュータレフォニーインテグレーションを使って音声通話の発着信を行います。
    - 受信電子メールメッセージを受け取ります（Siebel Email Response）。
  - **受信コミュニケーション**：サードパーティの電子メールサーバーとの統合、および受信電子メール（Siebel Email Response を使用した場合）の処理をサポートします。
  - **送信コミュニケーション**：送信コミュニケーションのために、電子メールサーバーやワイヤレスメッセージプロバイダーなど、さまざまなサードパーティのコミュニケーションシステムへの統合をサポートします。
    - Siebel Email Response を使って返信電子メールを送信するエージェントをサポートします。
    - Siebel アプリケーションユーザーに Send Email および Send Fax の各コマンドをサポートします（Send Page もサポートしますが、この場合は Page Manager サーバーコンポーネントを使用します）。
    - コミュニケーションリクエストを使って送信コミュニケーションコンテンツ（電子メール、ファックス、またはポケベル）を送信するユーザーをサポートします。リクエストは、『Siebel Communications Server 管理ガイド』で説明されているユーザーインターフェイスを通じてプログラムからまたは手動で作成および送信できます。
- Siebel モジュールの多くが送信コミュニケーションのためにワークフローを通じてビジネスサービスメソッドを起動します。

## セッションコミュニケーションインフラストラクチャ

セッションコミュニケーションでは、Communications Server コンポーネントを使ってコンタクトセンターエージェントまたはその他のユーザーがインタラクティブなコミュニケーションの作業項目を処理できます。たとえば Siebel CTI では、この機能をサポートすることにより、エージェントがコミュニケーションツールバーを使って音声通話を処理できます。

セッションコミュニケーションのパフォーマンスの問題を防止または解決するには、セッションコミュニケーションをサポートするインフラストラクチャを理解することが大切です。

この項、および次の項では、セッションコミュニケーションのパフォーマンスについて取り上げます。

- [71 ページの「セッションコミュニケーションのパフォーマンスに影響する要因」](#)
- [72 ページの「セッションコミュニケーションのトポロジに関する注意事項」](#)
- [73 ページの「Session Communications のチューニングを行うための最善の方法」](#)

### 重要な Siebel Server コンポーネント

Siebel Server 環境内で、セッションコミュニケーションは主に次のコンポーネントによりサポートされます。

- **Communications Session Manager (CommSessionMgr)** : このサーバーコンポーネントは、音声通話などのインタラクティブなコミュニケーション作業項目を管理します。
- **Application Object Manager (AOM)** : このサーバーコンポーネントは、コミュニケーション作業項目を処理するユーザー（エージェント）を含めて、Siebel Web クライアントを使用するエンドユーザーのアプリケーションセッションを管理します。エージェントからのインタラクティブなコミュニケーションリクエストは、一般的に AOM を経由します。

AOM については、[第 3 章「パフォーマンスのための Siebel Application Object Manager のチューニング」](#)を参照してください。

- **Server Request Broker (SRBroker)** : このサーバーコンポーネントは、AOM と、CommSessionMgr をはじめとする他のいくつかの Siebel Server コンポーネントとの間のコミュニケーションを処理します。

たとえば、Siebel CTI エージェントがコミュニケーションツールバーから電話をかけると、リクエストが AOM から SRBroker を経由して CommSessionMgr に送られます。

CommSessionMgr が AOM と同じマシン上で実行しているか、異なるマシン上で実行しているかを問わず、SRBroker が使用されます。このシナリオについては、[72 ページの「セッションコミュニケーションのトポロジに関する注意事項」](#)を参照してください。

SRBrokerについては、[49ページの「Server Request Broker\(SRBroker\)のチューニング」](#)を参照してください。

## その他の Siebel Server コンポーネント

Siebel Server 環境およびコミュニケーションインフラストラクチャでは、次のコンポーネントも使用される場合があります。

- **Communications Configuration Manager (CommConfigMgr)** : コミュニケーション設定データのキャッシュのためにオプションで使用されます。
- **Communications Inbound Receiver (CommInboundRcvr)** : 詳しくは、[80 ページの「Siebel Email Response のインフラストラクチャ」](#) を参照してください。
- **Communications Inbound Processor (CommInboundProcessor)** : 詳しくは、[80 ページの「Siebel Email Response のインフラストラクチャ」](#) を参照してください。
- **Communications Outbound Manager (CommOutboundMgr)** : 電子メールまたは他のタイプのメッセージを送信します。

## Siebel 製品モジュール

Siebel CTI または Siebel Email Response に加えて、セッションコミュニケーションでは次の Siebel 製品モジュールが使用される場合があります。

- **Siebel CTI Connect** : このモジュールは、CTI ミドルウェア、コミュニケーションドライバ、およびサンプルのコミュニケーション設定データから成ります。Siebel CTI Connect は、サードパーティの CTI ミドルウェア Intel NetMerge (以前の Dialogic CT Connect) を基盤とします。Siebel CTI Connect については、『Siebel Business Applications Third-Party Bookshelf』に含まれる Intel のマニュアルを参照してください。
- **Siebel Smart Answer** : このモジュールは、電子メールおよび検索リクエストのコンテンツを分析し、自動返信を返すか、またはユーザーに承認を求める 1 つ以上の返信を提示します。

Siebel Smart Answer は、Banter のサードパーティ製品に基づいています。『Siebel Smart Answer Administration Guide』、および『Siebel Business Applications Third-Party Bookshelf』にある Banter のマニュアルを参照してください。

詳しくは、[85 ページの「Siebel Smart Answer のパフォーマンス」](#) を参照してください。

## サードパーティ製品のモジュール

CTI ミドルウェア / ドライバ / 設定、ルーティング製品、プレディクティブダイヤラー、インタラクティブ音声応答モジュール、電子メールサーバー、ファックスサーバーなど、サードパーティ製品のモジュールが使用される場合があります。サポートされる電子メールサーバーについては、Siebel SupportWeb 上の「System Requirements and Supported Platforms & Miscellaneous Documentation」を参照してください。

**備考** : Siebel CTI Connect を使用しない場合で Siebel CTI を使用するには、サードパーティの CTI ミドルウェアパッケージを入手し、ベンダーとの共同作業でそのモジュールを統合する必要があります。

## セッションコミュニケーションのパフォーマンスに影響する要因

ここでは、セッションコミュニケーションの展開について、パフォーマンスを向上させる要因や、パフォーマンスに影響する要因を説明します。

エージェントは、その展開に応じて電話の発信 (Siebel CTI)、電子メールメッセージ (Siebel Email Response)、他のコミュニケーションチャネルの作業項目、またはこれらの組み合わせを処理できます。

- **1 時間あたりに処理された電話の着信数**：コミュニケーションインフラストラクチャにより 1 時間あたりに（またはその他の時間あたりに）処理された電話の着信数（またはその他タイプの作業項目数）。
- **1 時間あたりに処理された電話の発信数**：コミュニケーションインフラストラクチャによって 1 時間あたりに（またはその他の時間あたりに）処理された電話の発信数（プレディクティブダイヤラーによる電話の発信では、Communications Server によって応対および処理された電話だけが対象になります）。

- **1 分あたりのユーザーコミュニケーションのアクション数（負荷）**：コミュニケーションに関連する 1 分あたりのユーザーアクションの平均数とそのユーザーアクション間の平均待ち時間。一般的に、コミュニケーションに関連するアクションは、コミュニケーションツールバーを使って実行するアクションです。

待ち時間がより長くなると、それに伴って Siebel データベースおよび Siebel Server 上の負荷が軽減します。待ち時間は、全体的なシステム負荷において重要な要因です。待ち時間の見積りでは、実際のユーザーによる使用に近似させる必要があります。

待ち時間および AOM チューニングについては、[第 3 章「パフォーマンスのための Siebel Application Object Manager のチューニング」](#)を参照してください。

- **同時コミュニケーションユーザーの数（エージェント）**：セッションコミュニケーション機能の同時ユーザー数。一般的にはコンタクトセンターエージェントの数です。この数値は、AOM 上の同時ユーザーの総数に占める割合です。

また、これらの機能をエージェントが使用する手法、1 エージェントあたりの作業項目の平均発着信数、そしてこれらの要因が組織のサービスの目的にどのように関連するかについて理解しておく必要があります。エージェントによっては、ACD キューから多数の作業項目を受け取るか、または多数の作業項目を開始することがあります。たとえば、スーパーバイザーまたはその他のユーザーがエージェントとして定義されていても、エスカレートされた作業項目だけを受け取る場合があります。

同時ユーザーおよび AOM チューニングについては、[第 3 章「パフォーマンスのための Siebel Application Object Manager のチューニング」](#)を参照してください。

- **顧客データの量**：顧客データの合計量。

データ量は、画面ポップアップの検索の実行、作業項目のルーティング、顧客ダッシュボードへの入力など、さまざまな目的でのデータ取得の速度に影響します。データ量がエージェントの経験する応答時間に直接影響することも少なくありません。このデータ量は現実的な値であることが必要です。また、実際の条件に合わせてデータベースをチューニングする必要があります。

これらの要因と他の多数の要因、たとえば平均電話時間、エージェントの電話間の平均時間などがコンタクトセンターエージェントの経験するシステムパフォーマンスに影響します。エージェントは、一般的な応答時間、画面ポップアップの応答時間、その他の知覚されるパフォーマンスの測定基準に関心を持ちます。

### サードパーティ製品の注意事項

展開に影響するあらゆる要件について、該当するサードパーティのマニュアルを確認してください。次に例を示します。

- CTI ミドルウェアソフトウェアによっては、1 つのコンタクトセンターサイトで対応可能なエージェント数が制限される場合があります。
- ACD キュー、プレディクティブダイヤラー、その他のモジュールとの統合が設定やネットワークトラフィックなどに影響する場合があります。
- (ACDスイッチとCTIミドルウェア間の)テレフォニーリンクの能力がパフォーマンスに影響する場合があります。

## セッションコミュニケーションのトポロジに関する注意事項

一般に、CommSessionMgrなどのセッションコミュニケーションのSiebel Communications Serverコンポーネントは、AOM を実行している同じ Siebel Server マシン上で実行します。しかし、AOM とは異なるマシン上でCommSessionMgr を実行しなければならない場合もあります。次に、これらのオプションについて詳細に説明します。

CTI ミドルウェアは一般に、各コンタクトセンター施設にあるサーバー上で実行されます。

### AOM マシン上での CommSessionMgr の実行

一般に、セッションコミュニケーションの Siebel Communications Server コンポーネントは、AOM を実行している同じ Siebel Server マシン上で実行します。このようなトポロジでは、AOM 負荷分散メカニズムにより Communications Server の負荷を間接的に分散できます。CommSessionMgr の負荷はかなり軽いために、単独では専用マシン上で実行する必要はありません。

エージェントが接続するすべての AOM で Enable Communication パラメータを「TRUE」に設定します。Siebel Server の負荷分散機能を使用する場合、リクエストの配信先になるすべての AOM を同じ方法で設定する必要があります。

### 専用マシン上での CommSessionMgr の実行

AOM コンポーネントとは別のマシン上で CommSessionMgr を実行することが必要な場合があります。

CommSessionMgr は、CTI ミドルウェアのコミュニケーションドライバを実行する同じマシン上で実行する必要があります。ドライバを特定のオペレーティングシステムプラットフォームで実行する必要がある場合、そのプラットフォームのマシン上に Siebel Server をインストールし CommSessionMgr を実行しなければなりません (コミュニケーションドライバは、Siebel SupportWeb 上の「System Requirements and Supported Platforms & Miscellaneous Documentation」に記載した、サポートされるいずれかの Siebel Server プラットフォームで実行する必要があります)。

異なるプラットフォームを使ったマシン上で AOM コンポーネント (Call Center Object Manager) を実行している場合は、CommSessionMgr を実行するマシンを指定するために、コミュニケーション設定内で CommSessionMgr および RequestServer をはじめとするいくつかのパラメータを設定します。このコミュニケーション設定のユーザーをサポートする AOM からのコミュニケーションセッションリクエストはすべて、専用マシン上の CommSessionMgr コンポーネントにルーティングされます。

詳しくは、74 ページの「CommSessionMgr コンポーネントのチューニング」を参照してください。これらのパラメータについては、『Siebel Communications Server 管理ガイド』を参照してください。

## Session Communications のチューニングを行うための最善の方法

ハードウェアリソースを最適に使用し、システムを適切に設定することにより、パフォーマンス面の目標を達成するために役立ちます。リソースと要件について慎重に考慮し、システムパフォーマンスを継続的にテストおよび監視する必要があります。

詳しくは、『Siebel Communications Server 管理ガイド』、『Siebel システム管理ガイド』、Siebel CTI Connect のマニュアルまたは関連するサードパーティのマニュアル、その他の資料を参照してください。

次に、パフォーマンスおよびスケーラビリティに関する目標を達成するために考えられる活動をいくつか示します。ただしこれらに限定されるわけではありません。

- システムトポロジを調整します。詳しくは、72 ページの「セッションコミュニケーションのトポロジに関する注意事項」を参照してください。
- AOM コンポーネントを設定します。詳しくは、74 ページの「AOM コンポーネントのチューニング」を参照してください。
- CommSessionMgr および関連コンポーネントを設定します。詳しくは、74 ページの「CommSessionMgr コンポーネントのチューニング」を参照してください。
- コミュニケーション設定、コミュニケーションドライバ設定などを変更します。以降の項で説明する活動の多くが同様の性質を備えています。

時間経過に沿って最適なシステムの実行を維持するには、受信するコミュニケーション量、ユーザー数などの変化について計画しておく必要があります。CTI ミドルウェアが受信するコミュニケーション量およびユーザー数の予測される増大に対応できることを確認します。コミュニケーション量およびユーザー数の増大に対応する場合に、より多くの AOM コンポーネントおよび CommSessionMgr コンポーネントを実行するために、追加ハードウェアが必要になる可能性があります。

## AOM コンポーネントのチューニング

CommSessionMgr コンポーネントと CommConfigMgr コンポーネントが使用するリソースは、それが動作する Siebel Server リソースのうちのわずかな割合でしかありません。CommSessionMgr コンポーネントまたは CommConfigMgr コンポーネントが存在する場合でも、AOM のパフォーマンスは全体的なシステムのパフォーマンスに最も大きな影響を与えます。

エージェントセッションの AOM メモリの要件は、多くの要因に左右されます。セッションコミュニケーションを使用するエージェントの AOM メモリの使用量は、他のユーザー（コミュニケーション設定内でエージェントとして定義されないユーザー）よりも大きくなります。

また AOM のチューニングは、コミュニケーション設定のキャッシュ方法に左右されます。[75 ページの「キャッシュによる AOM サーバーリソースの節約」](#) も参照してください。

AOM のチューニングについては、[第 3 章「パフォーマンスのための Siebel Application Object Manager のチューニング」](#) を参照してください。

## CommSessionMgr コンポーネントのチューニング

CommSessionMgr コンポーネントでは、MaxTasks パラメータにより一度に処理可能なコミュニケーションイベントの最大数を指定します。

特に CommSessionMgr が各 AOM マシン上で実行する場合に、MaxTasks、MinMTServers、および MaxMTServers の各パラメータのデフォルト値は適切であることが普通です。

専用 Siebel Server マシンを使って CommSessionMgr コンポーネントを実行する場合、CPU やメモリなどのサーバーリソースの使用を最適化するためにこれらのパラメータをより高い値に設定した方が適切なこともあります。[72 ページの「セッションコミュニケーションのトポロジに関する注意事項」](#) も参照してください。

## キャッシュによる AOM サーバーリソースの節約

各エージェントセッションでコミュニケーション設定データをより速くロードし、AOM 上でのサーバーリソースに対する要求を軽減するために、2つのキャッシュメカニズムを使用できます。

これらのキャッシュメカニズムは、ともに使用することもできますし、個別に使用することも可能です。詳しくは、『Siebel Communications Server 管理ガイド』を参照してください。

- **CommConfigCache パラメータ (AOM)** : AOM 上で CommConfigCache パラメータを「TRUE」に設定すると、最初のエージェントがログインしたときにコミュニケーション設定データがキャッシュされます。設定データは、AOM が再起動されるまでキャッシュされます。同じコミュニケーション設定に関連付けられたエージェントでは、各エージェントセッションで同じキャッシュ済みデータが使用されます。74 ページの「AOM コンポーネントのチューニング」も参照してください。
  - 以降のエージェントのログインでは、設定データがデータベースからではなくキャッシュからロードされるので、パフォーマンスが向上します。
  - また、AOM のスケーラビリティも向上します。これは、エージェントセッション間で AOM メモリ内の設定データが共有され、したがってエージェントセッション数が増大してもサーバーリソースの使用が抑制されるためです。
- **CommConfigMgr サーバーコンポーネントおよび CommConfigManager パラメータ (AOM)** : CommConfigMgr サーバーコンポーネントは最初のエージェントがログインしたときにコミュニケーション設定データをキャッシュします。AOM 上で CommConfigManager パラメータを TRUE に設定すると、このサーバーコンポーネントが有効になります。
  - 以降のエージェントのログインでは、設定データがデータベースからではなくキャッシュからロードされるので、パフォーマンスが向上します。
  - CommConfigMgr コンポーネントを使用することによりデータをキャッシュして、エージェントセッションごとにログインプロセスを高速化し、メモリの使用量を減らすことができます。これは、コンポーネントが AOM コンポーネント上ですでにキャッシュされた設定データを使用するためです。
  - AOM の CommConfigCache パラメータと連携して CommConfigMgr コンポーネントを使用する必要はないが、それらと一緒に使用する場合、AOM に対してだけでなく、エンタープライズレベルでコミュニケーション設定データがキャッシュされます。これらのメカニズムを個別に使用する場合と比較すると、全体的なパフォーマンスが向上します。

## コミュニケーション設定のパフォーマンスの向上

セッションコミュニケーションの展開時に、コミュニケーション設定の作成、従業員のエージェントとしての定義、各エージェントへの設定の関連付けを行います。これらの作業の方法がパフォーマンスとスケーラビリティに影響します。

複数の物理サイト間で多数のエージェントをサポートする展開では、設定内のエージェントをグループ化する条件を指定する必要があります。

たとえば、同じ国または市外局番内など、特定の場所のエージェントでは、パラメータ `DialingFilter.RuleN` を使って番号フィルターを定義すると良い場合があります。そして他セットのエージェントには、おそらく別の番号フィルターが適切です。

さらに、スイッチ、テレセット、または CTI ミドルウェアの設定はコミュニケーション設定を反映して選択され、おそらくそれは物理的な場所によって異なります。

単一の場所のユーザーにだけ適用するコミュニケーション設定を定義すると便利な場合があります。この方法では、コミュニケーション設定、テレセット、その他の要素を定義する手順が簡素化されるだけでなく、AOM のメモリや CPU などのサーバーリソースに対する要求も軽減できます。

コンタクトセンター間で電話の転送や同様の機能をサポートする必要がある場合、その他にも考慮すべき設定の問題が存在します。

コミュニケーション設定とエージェントの定義については、『Siebel Communications Server 管理ガイド』を参照してください。

## セッションコミュニケーションのログの設定

パフォーマンスの監視またはチューニングの一環として、ログのデータを分析する場合があります。詳しくは、[第 14 章「Siebel ARM による Siebel アプリケーションのパフォーマンスの監視」](#)を参照してください。

より高レベルのログでは、システムエラーまたはパフォーマンスの問題を解決するために役立つより詳細なデータが得られます。これは、システムのテスト時に適しています。しかし本番システムでは、パフォーマンスを向上させるためにログのレベルを下げる必要があります。

セッションコミュニケーションに該当するログ関連のパラメータを次にまとめます。AOM コンポーネントは、コミュニケーションツールバー、画面ポップアップなどの使用をはじめとするユーザーのクライアントセッションに関連するアクティビティをログに記録します。CommSessionMgr は、コミュニケーションドライバのコマンドやイベントなど、このコンポーネントに関連するアクティビティをログに記録します。

AOM と CommSessionMgr のログは、ユーザーごとに個別のファイルに書き込まれます。一般的に、これらのログメカニズムはともに、同じファイルセットに書き込みます（ただし必ずしもそうであるわけではありません）。これにより、特定のユーザーセッションのセッションコミュニケーションに関連する問題の監視およびトラブルシューティングがより簡単になります。

これらのログパラメータについては、『Siebel Communications Server 管理ガイド』を参照してください。

## AOM ログパラメータ

セッションコミュニケーションのアクティビティのログを記録する AOM パラメータを次に示します。

- **CommLogFile** : ログファイルの名前を指定します (デフォルト値は SComm.log)。エージェントセッションごとに、個別のログファイルが SComm\_username.log の形式で作成されます。
- **CommLogDebug** : ログファイルに詳細を含めるかどうかを指定します。FALSE に設定すると、パフォーマンスが向上します。
- **CommMaxLogKB** : ログファイルの最大サイズを指定します。
- **CommReleaseLogHandle** : ログファイルのハンドルを定期的に関断することを指定します。デフォルト設定である TRUE でより優れたパフォーマンスが得られます。

## CommSessionMgr ログパラメータ

セッションコミュニケーションのアクティビティのログを記録する CommSessionMgr パラメータを次に示します。

- **LogFile** : ログファイルの名前を指定します (デフォルト値は SComm.log)。エージェントセッションごとに、個別のログファイルが SComm\_username.log の形式で作成されます。
- **LogDebug** : ログファイルに詳細を含めるかどうかを指定します。FALSE に設定すると、パフォーマンスが向上します。
- **MaxLogKB** : ログファイルの最大サイズを指定します。
- **ReleaseLogHandle** : ログファイルのハンドルを定期的に関断することを指定します。デフォルト設定である TRUE でより優れたパフォーマンスが得られます。

## Siebel CTI Connect ドライバのログパラメータ

セッションコミュニケーションのアクティビティをログに記録する Siebel CTI Connect のコミュニケーションドライバには、次のようなパラメータがあります。

- **Driver:DriverLogFile** : ログファイルの名前を指定します (デフォルト値は ctc.log)。ドライバセッションで (全ユーザーのための) 単一のログファイルが作成されます。また、Siebel CTI Connect のイベントがログに記録されます。
- **Service:ServiceLogFile** : ログファイルの名前を指定します (デフォルト値は ctc\_{@Username}.log)。エージェントセッションごとに、個別のログファイルが ctc\_username.log の形式で作成されます。エージェントセッションごとに Siebel CTI Connect イベントのログも記録されます。
- **LogDebug** : ログファイルに詳細を含めるかどうかを指定します。FALSE に設定すると、パフォーマンスが向上します。
- **MaxLogKB** : ログファイルの最大サイズを指定します。
- **ReleaseLogHandle** : ログファイルのハンドルを定期的に関断することを指定します。デフォルト設定である TRUE でより優れたパフォーマンスが得られます。

## セッション接続の可用性の向上

エージェントがブラウザの障害または接続のドロップを経験した後に Siebel アプリケーションにログインしたときに、セッションコミュニケーションが依然として使用できない場合があります。

セッションコミュニケーションの可用性は、パフォーマンスの問題として捉えることができます。セッションコミュニケーションの可用性が喪失すると、エージェントの生産性への影響だけでなく、その他の機能をサポートできるはずのサーバーリソースも浪費されます。

次のメカニズムを使ってセッションコミュニケーションの可用性を向上できます。

- **Push Keep Alive ドライバ**：Push Keep Alive コミュニケーションドライバを使用すると、エージェントに空のメッセージ（ハートビートメッセージ）が定期的にプッシュされます。これは、コミュニケーションプッシュチャンネルを継続するために役立ちます。この機能は、タイムアウトの強制によりコミュニケーションセッション接続のドロップが時おり発生する環境で効果的です。

たとえば、多くの顧客が Web サーバーの負荷を分散するための何らかのネットワーク装置を展開しています。デフォルトでは、このようなネットワーク装置によりブラウザへの接続がタイムアウトされ、その結果エージェントのコミュニケーションが中断する場合があります。Push Keep Alive ドライバは定期的にトラフィックを生産するので、非アクティブ状態の接続がタイムアウトすることがありません。

Push Keep Alive ドライバを使用するには、ドライバプロファイルを作成した上で、PushKeepAliveTimer ドライバパラメータを使ってハートビート間隔（180 秒など）を指定します。次に、このプロファイルをコミュニケーション設定に追加します。

**ChannelCleanupTimer パラメータ（コミュニケーション設定）**：コミュニケーション設定の

ChannelCleanupTimer パラメータは、セッションタイムアウトに関連する再接続の遅延を減少させます。このパラメータにより、たとえば接続のドロップやブラウザの障害により接続が機能しなくなったことをシステムが識別できるようになります。

**備考**：Push Keep Alive ドライバを使用している場合は、ChannelCleanupTimer パラメータも使用する必要があります。

- **CommMaxMsgQ パラメータと CommReqTimeout パラメータ（AOM）**：エージェントの接続を効果的に管理するために、一般的なアプリケーションタイムアウトの設定に加えて、AOM パラメータ CommMaxMsgQ および CommReqTimeout を設定することも役立ちます。
- **バックアップ Communications Session Manager (CommSessionMgr) コンポーネント**：コミュニケーション設定パラメータを使用して、バックアップ CommSessionMgr コンポーネントを指定できます。プライマリ CommSessionMgr コンポーネントに障害が発生して再起動しない場合に、エージェントが介入しなくてもバックアップ CommSessionMgr コンポーネントは別の Siebel Server マシンで実行され、アクセスできます。

これらの機能の使用については、『Siebel Communications Server 管理ガイド』を参照してください。

## 画面ポップアップのパフォーマンスの向上

コンタクトセンターエージェントが経験する画面ポップアップの応答時間は、パフォーマンスが受け入れ可能であるかどうかを示す重要な指標です。画面ポップアップは、コミュニケーションイベントに応じて表示されるビュー、および特定レコード（オプション）です。この種のイベントは一般的に、CTI ミドルウェアから受信します。たとえば、着信した電話の呼び出しや、エージェントによる電話の応対などです。

画面ポップアップの動作は、特定の電話に添付されたデータに基づいてその電話に適用される電話処理ロジックにより決定します。個別エージェントの動作も [ユーザー設定] 画面の [コミュニケーション] セクションのユーザー設定による影響を受けます。

画面ポップアップのパフォーマンスは、イベントハンドラやイベント応答などのコミュニケーション設定要素の相対的な複雑さ、および起動されるスクリプトまたはビジネスサービスに左右されます。クエリーの指定、データベースのパフォーマンス、およびネットワークの容量と遅延もまた、画面ポップアップのパフォーマンスに影響します。詳しくは、76 ページの「[コミュニケーション設定のパフォーマンスの向上](#)」を参照してください。

Siebel Web クライアントの応答時間については、第 5 章「[パフォーマンスのための Siebel Web クライアントのチューニング](#)」を参照してください。

## Siebel CTI Connect の画面ポップアップに関するパフォーマンスの向上

Siebel CTI Connect を使用している場合、別の方法で画面ポップアップのパフォーマンスを向上させることもできます。一般的な注意事項については、79 ページの「[画面ポップアップのパフォーマンスの向上](#)」を参照してください。

エージェントが電話に対応すると表示される画面ポップアップでは一般に、エージェントが電話に対応してから CTI ミドルウェアがエージェントを発信者に接続するまでに若干の遅延が存在します。Siebel CTI Connect では、この遅延を減少させることができます。

Siebel CTI Connect の Siebel CTI Connect コミュニケーションドライバには、EventAnswerCall デバイスイベントが含まれます。このイベントは、AnswerCall デバイスコマンドが Siebel CTI Connect ドライバによって起動されたとき、一般的にはエージェントがコミュニケーションツールバーを使って電話に対応したときに常に発生します。EventAnswerCall イベントは Siebel CTI Connect が該当の TpAnswered イベントを送信する前に発生するので、画面ポップアップを生成するための余剰時間が得られます。

この機能を使用するには、EventAnswerCall デバイスイベントに基づいてイベントハンドラ定義を作成します。このイベントハンドラでは、画面ポップアップを生成して適切なイベントログを開始するイベント応答を起動します。

詳細および例については、『Siebel Communications Server 管理ガイド』を参照してください。

## 活動の作成によるパフォーマンスへの影響の確認

デフォルトでは、コミュニケーション作業項目ごとに S\_EVT\_ACT テーブルおよび関連テーブル内に活動レコードが作成されます。

展開の計画時には、活動レコードの作成の有無および活動レコードの作成方法について考慮するとともに、適切なデータベーステーブルのインデックスおよびレイアウトの確認と活動レコードの生成によるパフォーマンスへの影響の確認を行う必要があります。

# Siebel Email Response のインフラストラクチャ

Siebel Email Response は Communications Server コンポーネントを使って、コンタクトセンターエージェントが受信電子メールメッセージを読み取って返信することを可能にします。

セッションコミュニケーションのパフォーマンスの問題を防止または解決するには、Siebel Email Response コミュニケーションをサポートするインフラストラクチャを理解することが大切です。

この項、および次の項では、Siebel Email Response のパフォーマンスについて取り上げます。

- 81 ページの「Siebel Email Response のパフォーマンスに影響する要因」
- 82 ページの「Siebel Email Response のトポロジに関する注意事項」
- 83 ページの「Siebel Email Response のチューニングを行うための最善の方法」

## 重要なサーバーコンポーネント

Siebel Server 環境内で Siebel Email Response は主に次のコンポーネントによりサポートされます。

- **Communications Inbound Receiver (CommInboundRcvr)** : 着信ワークアイテムを受信して、Communications Inbound Processor で処理するためのキューに登録します。
  - Communications Inbound Receiver は、受信した非リアルタイムワークアイテム（導入したほとんどの Siebel eMail Response の電子メールメッセージなど）を Communications Inbound Processor で処理するためにキューに登録します。
  - Communications Inbound Receiver は、受信したリアルタイムワークアイテム（Siebel CTI の電話、導入した一部の Siebel eMail Response の電子メールメッセージなど）を処理します。この場合、Communications Inbound Processor は使用されません。
- **Communications Inbound Processor (CommInboundProcessor)** : Communications Inbound Receiver によってキューに登録された着信ワークアイテムを処理します。
- **Communications Outbound Manager (CommOutboundMgr)** : 電子メールまたは他のタイプのメッセージを送信します。
- **Siebel File System Manager (FSMSrvr)** : Siebel File System への書き込みおよび Siebel File System からの読み取りを行います。このコンポーネントは、受信メッセージの処理前の保存、送受信電子メールメッセージへの添付ファイルの保存を行います。

## 他の Siebel コンポーネントまたはモジュール

Siebel Email Response に加えて、次の Siebel コンポーネントまたはモジュールが使用される場合があります。

- **Siebel Smart Answer**：このモジュールは、電子メールおよび検索リクエストのコンテンツを分析し、自動返信を返すか、またはユーザーに承認を求める 1 つ以上の返信を提示します。

Siebel Smart Answer は、Banter のサードパーティ製品に基づいています。『Siebel Smart Answer Administration Guide』、および『Siebel Business Applications Third-Party Bookshelf』にある Banter のマニュアルを参照してください。

詳しくは、[85 ページの「Siebel Smart Answer のパフォーマンス」](#)を参照してください。

- **Siebel 割当マネージャ**：このモジュールは、エージェントへの電子メールメッセージのルーティングで使用される場合があります。

## サードパーティの電子メールサーバー

Siebel Email Response は、サードパーティの電子メールサーバーと併用できます。展開に影響するあらゆる要件について、電子メールサーバーのマニュアルを確認してください。サポートされる電子メールサーバーについては、Siebel SupportWeb 上の「System Requirements and Supported Platforms & Miscellaneous Documentation」を参照してください。

# Siebel Email Response のパフォーマンスに影響する要因

ここでは、Siebel Email Response の展開について、パフォーマンスを向上させる要因や、パフォーマンスに影響する要因を説明します。

- **1 時間あたりに処理された電子メールメッセージの受信数**：コミュニケーションインフラストラクチャによって 1 時間あたりに（またはその他の時間あたりに）処理された電子メールメッセージの受信数。

送信メッセージ処理のための要件は比較的小さく、受信メッセージ量に大きく左右されます。しかし、CommOutboundMgr コンポーネントまたは電子メールシステムの他の使用分も考慮する必要があります。たとえば、CommOutboundMgr を通じて電子メールを送信するために Send Email コマンドを設定する場合があります。

- **顧客データの量**：テンプレートまたはカテゴリ、資料アイテムなどを含む顧客データの合計量。関連要因に、テンプレート形式（HTML またはプレーンテキスト）があります。

Siebel Smart Answer を展開する場合、知識ベースのサイズも考慮する必要があります。

その他の要因には、受信する電子メールメッセージおよび送信する返信のサイズと複雑さなどが挙げられます。

また、返信にオリジナルメッセージを含めるかどうか（[返信にオリジナルメッセージを含む] 設定）、HTML またはプレーンテキストをエージェントのデフォルトのメッセージフォーマットにするかどうか（[eMail Response デフォルトのメッセージ形式] 設定）など、[ユーザー設定] 画面の [送信コミュニケーション] セクションのユーザー設定が関連します。

**備考：** このマニュアルでは、Siebel Email Response について送受信の電子メール処理に焦点を当てます。マルチチャネル環境内では、セッションコミュニケーションのパフォーマンスの問題も考慮します。Siebel Smart Answer、特にその自動応答機能を使用することにより、受信電子メールを処理するためのエージェント数、AOM や CommSessionMgr などセッションに関連するコンピューティングリソースの対応する要求量が削減されます。

## Siebel Email Response のトポロジに関する注意事項

受信電子メールメッセージの処理は、送信電子メッセージの処理と比較して、サーバーリソース、特に CPU 使用レベルについてより厳しい要求を持ちます。

単一の返信グループに関連する受信メッセージは、単一のマシン上で処理する必要があります。

受信メッセージ量が相応に大きく、CommInboundRcvr、CommInboundProcessor、および他のコンポーネントを実行するために使用可能なサーバーマシンが複数存在する場合、他の Communications Server コンポーネントとは異なるマシン上で CommInboundRcvr および CommInboundProcessor を実行することを検討する必要があります。これらのコンポーネントのトポロジオプションは、リアルタイム処理と非リアルタイム処理では異なります。

CommInboundRcvr および CommInboundProcessor については、『Siebel Communications Server 管理ガイド』と『Siebel eMail Response 管理ガイド』を参照してください。

CommOutboundMgr および Siebel Smart Answer (Smart Answer Manager) は、必要に応じて異なるマシン上で実行できます。

単一返信グループ内の複数の電子メールアカウントに対応するメッセージの処理を組み合わせることで、受信メッセージをより効率的に処理できます。ただし、メッセージ量の増大が予測される場合は、各返信グループにより処理される電子メールアカウントの数を制限することにより、複数のサーバー間で処理を分散し、それによって処理のボトルネックを回避する柔軟性が得られます。

# Siebel Email Response のチューニングを行うための最善の方法

ハードウェアリソースを最適に使用し、システムを適切に設定することにより、パフォーマンス面の目標を達成するために役立ちます。リソースと要件について慎重に考慮し、システムパフォーマンスを継続的にテストおよび監視する必要があります。

詳しくは、『Siebel eMail Response 管理ガイド』、『Siebel Communications Server 管理ガイド』、『Siebel Smart Answer Administration Guide』、関連するサードパーティのマニュアル、その他の資料を参照してください。

## CommInboundRcvr スレッドの設定

各 CommInboundRcvr タスクは、受信メールを処理するために複数のスレッドを実行します。スレッド数を決定するには、パラメータ MinThreads および MaxThreads を設定します。ある特定のサーバマシ上で CPU 能力が余剰する場合、該当する各 CommInboundRcvr タスクのために実行するスレッドを増やすことができます。

## 電子メールディレクトリの管理

デフォルトで CommInboundRcvr は、該当する返信グループおよびワークフロープロセスでメッセージの処理が可能になるまで、受信電子メールメッセージのコンテンツを Siebel Server インストールディレクトリのサブディレクトリに一時的に書き込みます。

Internet SMTP/POP3 Server コミュニケーションドライバのパラメータを使って、受信電子メール、処理済み電子メール、送信済み電子メール、その他の処理状態にある電子メールメッセージの代替ディレクトリの場所を指定できます。また、たとえば処理済みの電子メールメッセージを保存または削除するかどうかを指定するためにドライバのパラメータを設定できます。

- システムの設定時には、一時的な電子メール処理ディレクトリ用のリソースの要件を考慮する必要があります。
- 受信電子メールまたはキューに入れられた電子メールのディレクトリからメッセージを削除しないでください。処理済みディレクトリまたは送信済みディレクトリに書き込まれた電子メールメッセージは、それぞれのニーズに合わせて、後で削除または保存できます。
- CommInboundRcvr 処理による一時的な電子メール処理ディレクトリへの書き込み頻度が高いため、ディスクのデフラグを定期的に行う必要があります。

電子メール処理ディレクトリについては、『Siebel Communications Server 管理ガイド』および『Siebel eMail Response 管理ガイド』を参照してください。

## 活動の作成によるパフォーマンスへの影響の確認

電子メール作業項目ごとに S\_EVT\_ACT テーブルおよび関連テーブル内に活動レコードが作成されます。

送受信メッセージのこのような活動レコードへの添付ファイルは、Siebel File System に保存されます。

展開の計画時には、活動レコードの作成方法および管理方法について考慮するとともに、適切なデータベーステーブルのインデックスおよびレイアウトの確認と、活動レコードの生成によるパフォーマンスへの影響の確認を行う必要があります。

さらに、活動の添付ファイルを保存するための Siebel File System のリソース要件を考慮する必要があります。

FSMSrvr サーバーコンポーネントは一般に、CommInboundRcvr および CommOutboundMgr を実行する同じ Siebel Server マシン上で実行することが必要です。

**備考：** Siebel Email Response 処理による Siebel File System への書き込み頻度が高いため、ディスクのデフラグを定期的に行う必要があります。

受信電子メールに対応して保存される活動の添付ファイルについては、『Siebel Communications Server 管理ガイド』および『Siebel eMail Response 管理ガイド』を参照してください。

## Siebel Email Response のログの設定

パフォーマンスの監視またはチューニングの一環として、ログのデータを分析する場合があります。詳しくは、[第 14 章「Siebel ARM による Siebel アプリケーションのパフォーマンスの監視」](#)を参照してください。

より高レベルのログでは、システムエラーまたはパフォーマンスの問題を解決するために役立つより詳細なデータが得られます。これは、システムのテスト時に適しています。しかし本番システムでは、パフォーマンスを向上させるためにログのレベルを下げる必要があります。

インターネット SMTP/POP3 サーバーコミュニケーションドライバの該当するパラメータの 1 つに LogDebug があります。詳しくは、『Siebel Communications Server 管理ガイド』を参照してください。

Siebel Email Response の適切なイベントログレベルには、タスクの実行、ワークフローステップの実行、ワークフロープロセスの実行、およびワークフローのパフォーマンスなどに対応するイベントログレベルがあります。

## Siebel Smart Answer のパフォーマンス

Siebel Smart Answer は、電子メールおよび検索リクエストのコンテンツを分析し、自動返信を返すか、またはユーザーに承認を求める 1 つ以上の返信を提示します。Smart Answer は、受信メッセージのコンテンツを読み取り、メッセージの性質（カテゴリ）を決定する内部 AI（人工知能）エンジンを備えています。

次に、パフォーマンスに影響する考慮すべき重要な要因を示します。

- **受信メッセージの複雑さ**：受信メッセージが複雑かまたは大きい場合、Smart Answer が処理するテキスト量が増大します。これは、Smart Answer のパフォーマンスに影響します。したがって、受信メッセージのフォーマットを管理できる場合は、メッセージを小さくするか、より簡素にして、Smart Answer のパフォーマンスを向上させることを考慮します。
- **知識ベース (KB) ファイル内のカテゴリ数**：カテゴリ数が増加すると、それに伴って Smart Answer がカテゴリを決定するために参照するデータ量が増大します。KB ファイル内のカテゴリ数を適度に抑えることをお勧めします。
- **Smart Answer がスタンドアロンモードまたはマスター/スレーブモードで実行するかどうか**：Smart Answer は、複数のサーバー間で Smart Answer の複数のインスタンスを同時に実行できるマルチサーバーモードをサポートしています。ただし、読み取った電子メールから「学習」し、KB にフィードバックを提供するマスターノードとして単一のノードが指定されます。しかしスレーブモードの Smart Answer は、フィードバックを KB に提供せずに、電子メールメッセージだけを処理します。
- **Smart Answer インスタンスの数**：デフォルトでは、MaxMTServers は 1 に設定されます。これは、大半の展開で十分な値です。
- **CommInboundRcvr または CommInboundProcessor に対する Smart Answer の配置**：Smart Answer および CommInboundRcvr または CommInboundProcessor はともに、受信電子メールメッセージのテキストを処理しますが、両者とも大量のサーバーリソースを消費します。したがって、受信電子メールの量、カテゴリの数、または受信メッセージの複雑さが増大する場合は、それに応じて CommInboundRcvr、CommInboundProcessor、および Smart Answer を別個の物理サーバー上で実行することを検討します。



# 7

## パフォーマンスのための Siebel ワークフローのチューニング

この章では、最適なパフォーマンスおよびスケーラビリティを達成し維持するために、ワークフローのプロセスとポリシーをチューニングする場合のガイドラインを示します。ここで説明する内容は、次のとおりです。

- 87 ページの「Siebel ワークフローについて」
- 88 ページの「ワークフローポリシーの監視」
- 90 ページの「パフォーマンスのためのワークフローポリシーのチューニング」
- 93 ページの「ワークフロープロセスのチューニング」
- 96 ページの「パフォーマンスのための Workflow Process Manager のチューニング」

Siebel ワークフローについては、『Siebel Bookshelf』の次のマニュアルを参照してください。

- 『Siebel Business Process Framework : Workflow ガイド』
- 『Configuring Siebel Business Applications』
- 『Siebel システム管理ガイド』

## Siebel ワークフローについて

Siebel ワークフローは、ビジネスプロセスを自動化する対話式ソフトウェアツールです。

ワークフロープロセスは、Siebel Tools のグラフィカルユーザーインターフェイスである Business Process Designer を使って設計および管理します。Business Process Designer を使った個別のワークフロープロセスの設計、計画、作成、およびテストについては、『Siebel Business Process Framework : Workflow ガイド』を参照してください。

ワークフローポリシーとワークフロープロセスは、ビジネスプロセスの自動化時に設計および作成する Siebel ワークフローの 2 つのコンポーネントです。これらのコンポーネントの定義は次のとおりです。

- **ワークフロープロセス**：ビジネスプロセスの表現。ワークフロープロセスには、ビジネスプロセスが開始および終了する時点を示し、そのビジネスプロセス内の個々の活動についての情報を保持する、1 つ以上のステップが含まれています。
- **ワークフローポリシー**：ビジネスルールの系統的な表現。ワークフローポリシーには、1 つ以上のポリシー条件と 1 つ以上のポリシーアクションが含まれています。ワークフローポリシーのポリシー条件がすべて真の場合、ポリシーアクションが発生します。ワークフローポリシーは 1 つのワークフローポリシーグループに含まれ、1 つのワークフローポリシーオブジェクトに関連付けられています。ワークフローポリシーには、その動作を制御するその他のプロパティも含まれています。

## ワークフローポリシーの監視

ワークフローポリシーを定期的に監視し、すべてのイベントが正しく処理されていること、Siebel Server がそのリソースを最適に利用していることを確認する必要があります。ログファイルを定期的に削除するようにすれば、ファイルが大きくなりすぎることはありません。ワークフローポリシーでは、ログ記録に General Events イベントを使用します。説明メッセージを表示するには、ログレベルを「3」に設定します。デバッグ情報を表示する場合は、「4」に設定します。

ワークフローポリシーは、次のようなビュー、ログファイル、およびテーブルを使用して監視できます。

- **【ポリシー頻度分析】ビュー**：詳しくは、88 ページの「[ポリシー頻度分析ビューの使用](#)」を参照してください。
- **ワークフローエージェント追跡ログ**：詳しくは、89 ページの「[ワークフローエージェント追跡ログの使用](#)」を参照してください。
- **【ワークフローポリシー】テーブル**：詳しくは、89 ページの「[ワークフローポリシーテーブルの監視](#)」を参照してください。

## ポリシー頻度分析ビューの使用

[ポリシー頻度分析] ビューには、実行されたすべてのポリシーのリストが表示されます。[ポリシー頻度分析] ビューを使用すると、ポリシーの実行頻度を時間を追って解析できます。

このビューには、ワークフローモニターエージェントプロセスで実行が検出された、すべてのポリシーのログが表示されます。ポリシー作成者は、ワークフローエージェントプロセスのアクティビティを監視して、現在のポリシーに不足がないかを判断できます。新しいポリシーを作成する必要性やポリシーを改善する必要性も判断できます。

[ポリシー頻度分析] ビューを使用すると、ポリシーログデータをグラフィック形式で表示できます。ログ情報は、ワークフローポリシーの Siebel Server コンポーネントによって生成されます。[ポリシー頻度分析] ビューには、Siebel クライアントの [管理-ビジネスプロセス] 画面で [ポリシー頻度分析] ビューに移動してアクセスできます。

[ポリシー頻度分析] ビューには、次のフィールドが含まれています。

- **ポリシー**：実行されたポリシーの名前
- **ワークフローオブジェクト**：割り当てられたワークフローポリシーオブジェクトの名前
- **オブジェクト識別子**：ポリシー実行対象となったワークフローポリシーオブジェクトの ID
- **オブジェクト値**：ポリシーを実行した行の識別情報
- **イベント**：ポリシー実行イベントの日時

## ワークフローエージェント追跡ログの使用

ワークフローエージェント追跡ログには、次の項目が含まれます。

- **ワークフローモニターエージェントのタスクログ**：ワークフローモニターエージェントは、その処理に関する詳細情報を追跡ファイルで提供する。
- **ワークフローアクションエージェントタスクログ**：ワークフローアクションエージェントは、その処理に関する詳細情報を追跡ファイルで提供する。

ワークフローアクションエージェントタスクの追跡を設定する必要があるのは、ワークフローモニターエージェントの Use Action Agent パラメータを「TRUE」に設定した場合のみです。その場合、ワークフローアクションエージェントを手動で起動する必要があります。また、電子メール統合を使用する場合にも手動で起動する必要があります。

Use Action Agent はデフォルトで「FALSE」になっています。ワークフローアクションエージェントは、ワークフローモニターエージェントによって自動的に起動されます。

- **Email Manager と Page Manager の追跡ログ**：

- 電子メールアクティビティの詳細レポートが必要な場合は、Trace Flag を「1」に設定して、Email Manager と Page Manager のコンポーネントを実行する。
- ワークフローアクションエージェントによって記録された電子メールリクエストとポケベルリクエストの状況に関する情報が必要な場合は、S\_APSRVR\_REQ を照会する。

## ワークフローポリシーテーブルの監視

ワークフローポリシーでは、リクエストの処理および追跡に、次の 3 つのデータベーステーブルを使用します。

- S\_ESCL\_REQ
- S\_ESCL\_STATE
- S\_ESCL\_ACTN\_REQ

これらのテーブルを監視して、ポリシーが正しく処理されていることを確認します。

ワークフローポリシー条件に対してトリガーが起動されると、エスカレーションリクエストテーブル S\_ESCL\_REQ にレコードが挿入されます。このテーブルのレコードによって、ワークフローポリシーを起動してアクションを実行できるデータベース内の行が指定されます。ワークフローモニターエージェントは要求を処理した後に、このテーブルから行を除去します。

S\_ESCL\_STATE 時間ベーステーブルには、実行されて（すべての条件が真）、持続時間要素が失効するのを待機しているすべての行が含まれます。

S\_ESCL\_ACTN\_REQ テーブルには、アクションの実行を待機しているすべての行が含まれます。これらの行にはポリシー違反があり、持続時間要素も失効しています。

S\_ESCL\_REQ、S\_ESCL\_STATE、または S\_ESCL\_ACTN\_REQ のいずれかのテーブルのサイズが非常に大きくなった場合は、監視されているポリシー数が多すぎる可能性があります。この場合、新しいワークフローポリシープロセスを作成して、負荷を共有してパフォーマンスを改善する必要があります。

行が監視されていて、時間間隔をおいた後もテーブルから除去されない場合は、データベーストリガーを除去せずにポリシーが非活動化された可能性があります。この場合、トリガーは、ワークフローポリシープロセスの作用を受けないデータを送信し続けています。これらのテーブルは、Generate Triggers を再起動しないと非常に大きくなる可能性があります。

有効なワークフローポリシーが失効したか、または削除された場合、未処理のレコードが S\_ESCL\_REQ、S\_ESCL\_STATE、または S\_ESCL\_ACTN\_REQ テーブルに残っていないことを確認します。

ストレージ、アクセス、およびキャッシュに関連するパラメータを調整することにより、S\_ESCL\_REQ テーブル、S\_ESCL\_ACTN\_REQ テーブル、および S\_ESCL\_STATE テーブルを管理します。これらのパラメータの適切な調整については、データベースのマニュアルを参照してください。また、これらのテーブルの重要性についてデータベース管理者（DBA）が認識していることを確認します。

## パフォーマンスのためのワークフローポリシーのチューニング

類似したポリシーをグループ化し、その仕事量を処理できる Siebel Server にそのグループを割り当てることにより、ポリシーのタイミング条件を満たしながらリソースを最適化するようにワークフローポリシーをチューニングできます。パフォーマンスのチューニングは、相関関係にある次のような方法で行います。詳しくは、次のトピックを参照してください。

- 90 ページの「Siebel Server の負荷を管理するワークフローポリシーグループの作成」
- 91 ページの「複数のワークフローモニターエージェントとワークフローアクションエージェント」
- 91 ページの「複数の Siebel Server 上でのワークフローエージェントの実行」
- 91 ページの「最適なワークフローポリシーグループのスリープ間隔の設定」
- 92 ページの「ワークフローモニターエージェントとワークフローアクションエージェントに最適なアクション間隔を設定する」

## Siebel Server の負荷を管理するワークフローポリシーグループの作成

ワークフローポリシーグループを使用すると、類似したポーリング間隔を持つポリシーをグループ化できます。これによって、負荷を分散し、効率のよい処理が可能になります。たとえば、ポリシートリガーイベントの数分内に対応しなければならない非常に重要なポリシーと、1 日以内に対応すればよいポリシーがある場合は、それらを異なるワークフローポリシーグループに割り当てることができます。

ポリシーを選択してグループ化することの利点は、ワークフローエージェントのポーリングリソースを少数のポリシーに集中できることです。これで、監視とアクションの実行を効果的に行うことができます。

## 複数のワークフローモニターエージェントとワークフローアクションエージェント

各ワークフローエージェントの組み合わせで、割り当てられているワークフローポリシーグループ内のポリシーを監視します。大量のデータを扱うコールセンターの場合、または非常に短いポーリング間隔を必要とするポリシーが多い場合は、複数のワークフローエージェントプロセスを作成して並列的に実行してもかまいません。単一のワークフローエージェントプロセスで多くのイベントを監視および処理していると、対応時間が長くなって、ポリシーで要求される時間間隔コミットに一致しなくなることがあります。

複数のワークフローモニターエージェントとワークフローアクションエージェントの並行実行：

- 1つのコンポーネントのポーリングリソースを少数のワークフローポリシーに集中させる。
- ワークフローポリシーイベントがトリガーされてからコンポーネントがイベントを認識するまでの時間を短縮することにより、高いスループットが得られる。

## 複数の Siebel Server 上でのワークフローエージェントの実行

異なる Siebel Server 上でワークフローエージェントプロセスを実行すると、各 Siebel Server の負荷を軽減できます。そして、各グループのポーリング間隔を調整し、重要性の低いポリシーのポーリングによって重要なポリシーの高効率処理が妨げられないようにすることができます。

ワークフローポリシープロセスを Siebel Server 間で分散する：

- ワークフローの CPU 条件を処理するための十分なリソースを持つ Siebel Server 上に高保守ポリシーをグループ化できます。
- 他の Siebel プロセスとリソースを共有する Siebel Server 上で低保守ポリシーを実行できます。

## 最適なワークフローポリシーグループのスリープ間隔の設定

類似したポーリング間隔のグループを作成すると、ポーリングレートが一致するワークフローエージェントプロセスにワークフローポリシーグループを割り当てることができます。Sleep Time パラメータを使って、各ワークフローポリシーグループに異なるポーリング間隔を割り当てることが可能です。

ワークフローポリシーのサーバー管理については、『Siebel Business Process Framework：Workflow ガイド』を参照してください。

ワークフローエージェントがすべてのリクエストを処理したら、ワークフローエージェントプロセスは、この引数によって指定される間隔だけスリープ状態になった後、処理を再開します。スリープ間隔は、できるかぎり大きな値に設定します。ただし、ビジネス要件に合う値でなければなりません。

**備考：**スリープ間隔の値を小さくしすぎると、インフラストラクチャ全体に過度の負荷がかかる場合があります。スリープ間隔がビジネスプロセスのコンテキストで可能な限り大きい値であることを確認します。

各ワークフローポリシーグループの要件に合わせて、ワークフローエージェントプロセスごとにスリープ間隔を調整します。

たとえば、ワークフローポリシーグループ A には、重大度 1 のサービスリクエストに対して 10 分以内に応答しなければならない取引先が含まれます。ワークフローポリシーグループ B には、14 日以内に顧客のフォローアップコールが必要なポリシーが含まれます。

ワークフローポリシーグループ A では時間が非常に重要です。したがって、スリープ間隔を 60 秒に設定して、割り当て済みのワークフローポリシーインスタンスのポーリングが頻繁に行われるようにします。ワークフローポリシーグループ B では、時間が最重要要素になりません。したがって、スリープ間隔を 48 時間に設定しても、ワークフローポリシーのインスタンスはそのコミットにまだ一致させることができます。

また、Sleep Time パラメータに最適な値を設定する必要があるケースとして、同一のレコードを更新するユーザーが複数存在することがあります。たとえば、サービスリクエストを監視するポリシーがあり、オープンサービスリクエストを検索して変更するユーザーが複数存在する場合、Sleep Time パラメータの値はユーザーがテキストフィールドを更新できるように、十分な長さに設定する必要があります。

スリープ間隔が短いと、「選択されたデータは他のユーザーが取り出していたので、他のユーザーにより変更されました。そのまま続けてください。」というエラーメッセージが表示されることがあります。この場合、レコードの新しいフィールド値が表示され、自分が行った変更は失われます。

**備考：**一定の期間にワークフローポリシーの速度が著しく低下する場合は、Siebel Server 上で他のプロセスが CPU リソースをめぐって競合していないかを確認する必要があります。その Siebel Server では一定の期間にアクティビティレベルが高くなり、その間ワークフローポリシープロセスの監視 / 動作能力が妨げられることがわかる場合があります。Siebel Server 上のワークフローポリシープロセスを調整して、ポーリング時間と使用可能リソースが矛盾しないようにします。

## ワークフローモニターエージェントとワークフローアクションエージェントに最適なアクション間隔を設定する

各ワークフローモニターエージェントまたはワークフローアクションエージェントのコンポーネントに対して、アクション間隔パラメータを設定できます。これにより、ベーステーブルの行に対して特定ポリシーのアクションが再実行されるタイミングが決まります。この設定によって、行の一致状態とその解消が続く場合に、アクションの実行回数を制限できます。

ワークフローアクションエージェントではなく、ワークフローモニターエージェントのアクション間隔パラメータを設定するのは、ワークフローモニターエージェントの Use Action Agent パラメータを「TRUE」に設定した場合のみです。Use Action Agent はデフォルトで「FALSE」になっています。

たとえば、サービスリクエストの重大度が「非常に高い」に設定されていて、このサービスリクエストがポリシーをトリガーする場合、現在の間隔中に重大度を変更された後で再び「非常に高い」に設定されたとき、そのポリシーアクションを再実行したくないこともあります。

## ワークフロープロセスのチューニング

ワークフロープロセスの実行時のパフォーマンスを向上させるには、次の項で説明するガイドラインに従います。

- 93 ページの「Search Specification パラメータの使用の最小化」
- 94 ページの「親ビジネスコンポーネントと子ビジネスコンポーネントに基づく条件の監視」
- 94 ページの「ワークフローのパフォーマンスのために Siebel Business Applications を設定する」
- 94 ページの「ワークフロープロセスのメモリのオーバーヘッドの監視」

**備考：**ここで掲載するパフォーマンスのチューニング情報は、ワークフロープロセスのパフォーマンスをチューニングおよび最適化するための一般的なガイドラインです。Siebel アプリケーションの各実装は固有であるため、ワークフロープロセスのそれぞれの使用方法もまた固有です。

### Search Specification パラメータの使用の最小化

サーバーコンポーネントの Search Specification パラメータ(エイリアス SearchSpec)は Siebel ワークフローの機能ですが、頻繁に起動されるワークフロープロセスでこのパラメータを使用するのはできるだけ控えることをお勧めします。

SearchSpec の使用（特に頻繁に呼び出されるプロセスに対する使用）を最小限にすると、ワークフローエンジンが SearchSpec 文字列を構築する必要がなくなるので、実行時のこのエンジンのパフォーマンスを向上できます。

ただし、SearchSpec の使用を完全に避けることはしないでください。このパラメータを使用しない場合、状況によっては現在の行に対するアクション、さらにはすべての行に対するアクションが発生することがあります。個々のガイドラインでは、次の点に注意してください。

- Siebel の処理では、SearchSpec の使用を最小にします。
- バッチプロセスリクエストでは、ビジネスオブジェクトに対して SearchSpec を使用して、処理する行数を制限します。

### SearchSpec 内のフィールドのインデックス作成

SearchSpec の使用が不可欠と判断した場合、使用するすべてのフィールドに適切にインデックスが付けられていることを確認するようにお勧めします。フィールドに適切なインデックスを作成することは、Siebel ワークフローとその基盤となるデータベースでクエリーを効率的に作成するために役立ちます。

## 親ビジネスコンポーネントと子ビジネスコンポーネントに基づく条件の監視

親ビジネスコンポーネントと子ビジネスコンポーネントの組合せを使用する、決定ステップやその他のステップでの条件の評価時には、式や条件を厳密にベンチマークすることをお勧めします。この作業では、SQL のスプールが必要になる場合もあります。詳しくは、175 ページの「生成された SQL のパフォーマンスの問題に関する分析」を参照してください。

**備考：** SQL のクエリープランを見ると、長いクエリーのパフォーマンスが十分ではないことがわかる場合があります。このような場合、条件を複数の決定ステップに分割し、各条件を個別に評価する方が優れています。

## ワークフローのパフォーマンスのために Siebel Business Applications を設定する

場合によっては、異なるオブジェクト間で比較を行うことが必要になります。

たとえば、サービスリクエストをそれに関連する取引先の業種に応じた候補に割り当てる場合を考えます。このような例では、取引先に対して適切な業種を取得するクエリー、または取引先に関連付けられたすべての業種に対して業種をチェックするクエリーを実行する必要があります。

この例のワークフロープロセスが頻繁に評価される場合は、ワークフローのパフォーマンスを向上させるために、適切な設定を通じてサービスリクエスト上で取引先の業種を開示することを考慮します。

## ワークフロープロセスのメモリのオーバーヘッドの監視

オーバーヘッド、パフォーマンス、およびスケーラビリティの特性は、ワークフローを AOM でローカルに実行しているか、または Workflow Process Manager (WfProcMgr) で実行しているか、さらに WfProcMgr をどこで実行しているかによって異なります。パフォーマンスとスケーラビリティの特性は、ワークフロープロセスリクエストに非同期モードを使用しているかどうかにも依存します。

詳しくは、『Siebel Deployment Planning Guide』および『Siebel Business Process Framework: Workflow ガイド』を参照してください。

### ワークフローを AOM でローカルに実行

ワークフローインスタンス（つまり、ワークフロー定義の 1 回の実行）は AOM の内部で実行できます。この場合、ワークフローはログインユーザーが使用する現在のスレッド内でローカルに実行します。これは、N 人のユーザーが接続されていて、すべてのユーザーがワークフロー定義を実行する必要がある場合、ワークフロー定義がそれぞれのユーザースレッドで実行されることを意味します。

このモードでは、ワークフローによりユーザーセッションメモリ（モデルとも呼ばれます）に対する固定のオーバーヘッド（100 ~ 200 KB）に加えて、ワークフロー内のタスクに含まれる他のオブジェクト（ビジネスコンポーネントなど）による占有メモリが使用されます。

通常、このオプションによりパフォーマンスは最高になりますが、スケーラビリティが重要な要素でない場合にのみ適しています。

## Workflow Process Manager でのワークフローの実行

ワークフロー自身は、個別のコンポーネント内で実行されます。このコンポーネントは、ワークフローをスケジュールするための固定リソースセット (MaxMTServers パラメータと MaxTasks パラメータ) を使用します。Workflow Process Manager (コンポーネントエイリアス WfProcMgr) は、複数のワークフローを実行するマルチスレッドプロセスであり、スレッドとモデルのプールを使用するため、よりスケーラブルです。

一般に、使用されるワークフローのモードは、アプリケーションで実現する機能に依存します。特に実行結果をすぐに必要としない場合は、ワークフロータスクを WfProcMgr でスケジュールすることをお勧めします。

WfProcMgr は、ワークフローが呼び出された、AOM と同じ Siebel Server で実行することも、専用の Siebel Server マシンで実行することもできます。ワークフローをローカルで実行する場合と比較して、WfProcMgr でワークフローを実行するとパフォーマンスが低下する場合がありますが、スケーラビリティは向上します。WfProcMgr を専用 Siebel Server で実行すると、一般にスケーラビリティは最も高くなりますが、WfProcMgr と AOM を同じ場所で実行するほうがパフォーマンスは良くなります。

## ワークフロープロセスリクエストの非同期モードについて

前述のすべてのワークフロープロセスの展開オプションに対して、ワークフロープロセスリクエストは同期モードまたは非同期モードを使用して処理できます。非同期モードを使用する場合、次のような利点と欠点があります。

### 利点：

- すべてのユーザースレッドがロードされません。
- 次の場合に、よりスケーラブルです。
  - 最大  $N$  の同時接続ユーザーが存在します。
  - 最大  $X$  の同時実行ワークフローが存在します。
  - $X$  が  $N$  より小さい場合、 $X$  のタスクを持つ WfProcMgr がはるかに大きなユーザープール ( $N$ ) を処理できます。

### 欠点：

- エラー時には、自動通知が行われないので、ログファイルを調査する必要があります。
- SRBroker がタイムアウト機能または再試行機能を持つことがあります。
- 遅延がわずかに大きくなります。応答ごとに 1 回のリクエストで最小限の追加コストがかかります。

# パフォーマンスのための Workflow Process Manager のチューニング

この項では、Workflow Process Manager のチューニングと最適化の一般的なアプローチについて説明します。

ここでは、前述したように Siebel アプリケーションのすべての実装が固有であり、したがってワークフロープロセスのすべての使用方法もまた固有であることに注意する必要があります。最適なスループットを実現するには、ワークフロープロセスのテストと、継続的な監視、そしてチューニングを行うことが大切です。

次の項で説明するガイドラインに従ってください。

■ 96 ページの「ビジネスサービスのキャッシュ」

■ 96 ページの「セッションのキャッシュ」

**備考：** この項で説明する内容は、一般的な背景情報であるとお考えください。特定サイトでチューニングに影響する多数の変動する要素について詳細に説明することはありません。ここでの説明は、Siebel テクニカルサービスによるチューニングに関する特定の推奨事項に優先するものではありません。

## ビジネスサービスのキャッシュ

Workflow Process Manager から呼び出されたビジネスサービスのキャッシュプロパティは「TRUE」に設定されています。この機能によって、ワークフローエンジンはビジネスサービスの読み込みと解析を再実行する必要がなくなるので、ビジネスサービスを呼び出すワークフローのパフォーマンスが向上します。

**備考：** Cache プロパティが「FALSE」に設定された定義済み Siebel ビジネスサービスでは、Cache プロパティを「TRUE」に再設定しないでください。

## セッションのキャッシュ

Workflow Process Manager のパラメータ OM (Model Cache Maximum : エイリアス ModelCacheMax) は、モデルオブジェクト (キャッシュされたセッション) のキャッシュサイズを決定します。キャッシュされたセッションでは、データベース接続およびロケール、ユーザー設定、およびアクセス制御に関するセッションデータが維持されます。

**備考：** セッションのキャッシュは、Workflow Process Manager などの非対話的なオブジェクトマネージャベースのサーバーコンポーネントのみに適用されます。AOM や EAI Object Manager コンポーネントには適用されません。

この機能では、要求されるたびに新しいセッションを作成する代わりに、既存のセッションを維持して再利用します。この機能を使用すると、Workflow Process Manager のログインパフォーマンスが向上します。

キャッシュ内の各モデルは、そのモデルのライフサイクルの間、2つのデータベース接続 (挿入、更新、削除オペレーション用の接続と読み取り専用オペレーション用の接続) を作成します。

デフォルト値は 10 です。0 に設定すると、このパラメータは無効になります。最大値は 100 です。通常 ModelCacheMax は、Workflow Process Manager コンポーネントがサポートする必要のある同時セッション数とほぼ同じ値に設定する必要があります。

**備考：** コンポーネントセッションが複数のユーザー ID を使用する場合、セッションのキャッシュのコストに対する効果は低下します。効果は、同じユーザー ID を使用するコンポーネントセッションに対して最大になります。

『Siebel システム管理ガイド』も参照してください。



# 8

## パフォーマンスのための Siebel Configurator のチューニング

この章では、Siebel Configurator のサーバーベースの展開について、そのパフォーマンスとスループットに影響する問題を解説するとともに、このモジュールで最適なパフォーマンスとスケーラビリティを実現および維持するためのチューニングのガイドラインを示します。ここで説明する内容は、次のとおりです。

- 100 ページの「Siebel Configurator のインフラストラクチャ」
- 100 ページの「Siebel Configurator のパフォーマンスに影響する要因」
- 101 ページの「Siebel Configurator のトポロジに関する注意事項」
- 103 ページの「Siebel Configurator のチューニングを行うための最善の方法」
- 106 ページの「Siebel Configurator のキャッシュについて」

Siebel Configurator は、製品の設定機能と解決策の算出機能を備えています。また、サーバーベースまたはブラウザベースのモジュールとして展開できます。

**備考：**この章では、Siebel Configurator のサーバーベースの展開についてのみ説明します。詳しくは、『Siebel 製品管理ガイド』を参照してください。

Siebel Configurator は、Siebel Order Management モジュールの 1 つです。Siebel Interactive Selling モジュールは相互に連携することにより、オンライン販売を始めとする取引の実施をサポートします。

Siebel Configurator について詳しくは、『Siebel Bookshelf』の次のマニュアルを参照してください。

- 『Siebel 製品管理ガイド』
- 『Siebel システム管理ガイド』

関連する Siebel Order Management モジュールについては、次のマニュアルを参照してください。

- 『価格設定管理ガイド』
- 『Siebel Order Management ガイド』
- 『Siebel eSales 管理ガイド』
- 『Siebel Advisor Administration Guide』（『Siebel Advisor 管理ガイド』）

## Siebel Configurator のインフラストラクチャ

Siebel Configurator では、設定セッションの管理のためにいくつかのインフラストラクチャ要素が使用されます。Siebel Server 環境内で Siebel Configurator は次のコンポーネントによりサポートされます。

- **Application Object Manager (AOM)** : Siebel Configurator の機能は、Siebel Call Center の Call Center Object Manager (米語環境のエイリアス SCCObjMgr\_enu) などの AOM 内で実行できます。
- **Siebel Product Configuration Object Manager (日本語版のエイリアス eProdCfgObjMgr\_jpn)** : 一部の Siebel Configurator 展開に適したオプションのコンポーネントです。AOM コンポーネントから提出されるユーザーセッションの設定リクエストを処理します。一般的にこのコンポーネントは、AOM を実行するマシン以外の別個の Siebel Server マシン上で動作します。個別の Siebel Server マシンで、このコンポーネントの複数のインスタンスを実行できるので、さまざまなインスタンスにリクエストを分散できます。詳しくは、[101 ページの「Siebel Configurator のトポロジに関する注意事項」](#)を参照してください。

**備考:** Siebel Product Configuration Object Manager エイリアスの 3 文字の拡張子 (eProdCfgObjMgr\_jpn の例では jpn) は、このアプリケーションオブジェクトマネージャに関連付けられている Locale Code パラメータ (エイリアス LocaleCode) の値に対応します。Locale Code パラメータについて詳しくは、『Siebel Global Deployment Guide』を参照してください。

Instance Broker (Complex Object Instance Service ビジネスサービス)、Object Broker (Cfg Object Broker ビジネスサービス) など、Siebel Configurator の内部アーキテクチャを構成する要素については、『Siebel 製品管理ガイド』を参照してください。

## Siebel Configurator のパフォーマンスに影響する要因

Siebel Configurator のサーバーベースの展開を計画する場合、または既存の展開のパフォーマンスをトラブルシューティングする場合、パフォーマンスを決定付けたり、パフォーマンスに影響を与えたりするいくつかの主要な要因について考慮する必要があります。

以降の項では、最適なパフォーマンスとスケーラビリティを実現および維持するのに役立つ情報とガイドラインを示します。

考慮すべきパフォーマンスのコンテキストには、次の応答時間が含まれます。

- **カスタマイズ可能製品のロード** : ユーザーが見積りまたはオーダーの [カスタマイズ] をクリックしてから、カスタマイズ可能製品のユーザーインターフェイスがロードされユーザーに表示されるまでに経過する時間です。
- **ユーザーの選択に対する応答** : ユーザーが選択を行ってから、Siebel Configurator がカスタマイズ可能製品の更新やコンフリクトを知らせるメッセージなど、応答を返すまでに経過した時間です。

次の要因、特にカスタマイズ可能製品のサイズと複雑さは、前述のパフォーマンスコンテキストの両方に関連します。

次に、Siebel Configurator のサーバーベースの展開時に重要となる、パフォーマンスに影響する要因をいくつか紹介します。

- **同時設定ユーザーの数**：カスタマイズ可能製品モデルにアクセスする同時ユーザーの数です。この数値は、AOM 上の同時ユーザーの総数に占める割合です。  
特に、1 時間あたりの設定セッションの総数と、これらのセッションの平均長に注意する必要があります。
- **製品モデルのサイズと複雑さ**：個々のカスタマイズ可能製品モデルの全体のサイズと複雑さです。特に複数の階層レベル、多数の制約、および複雑なユーザーインターフェイスが定義されている場合に重要です。  
パフォーマンスに影響する主な潜在的要因の 1 つとして、Quote、Quote Item、Quote Item Attribute、Order、Order Item、Order Item Attribute などの適切なビジネスコンポーネント上でのイベント更新に添付されているカスタムスクリプトが挙げられます。
- **製品モデルの数**：ユーザーによってアクセスされたカスタマイズ可能製品モデルの数。各ユーザーが複数のカスタマイズ可能製品モデルに一度にアクセスしないことを前提としています。しかし、同時ユーザーのグループが複数のモデルにアクセスする場合は、それぞれのモデルを個別にキャッシュに格納する必要があります。

## Siebel Configurator のトポロジに関する注意事項

ここでは、Siebel Configurator のサーバーベースの展開に関するトポロジを定義する場合の注意事項について説明します。Siebel Configurator の展開では、トポロジについて主に次の 2 つのアプローチがあります。

- AOM コンポーネント内で Siebel Configurator を実行する。  
詳しくは、[101 ページの「AOM コンポーネント内での Siebel Configurator の実行」](#)を参照してください。
- 1 台以上の専用 Siebel Server 上で Siebel Configurator を実行する（このようなサーバーは、AOM を実行中のマシンに対してリモートであるために、リモートサーバーと呼ばれることもあります。一般的に、ここでは「専用サーバー」という用語を使用します）。  
詳しくは、[102 ページの「専用サーバー上での Siebel Configurator の実行」](#)を参照してください。

Siebel Configurator の最適な展開アプローチ、およびこのモジュールに必要なサーバーマシンの最適数は、[100 ページの「Siebel Configurator のパフォーマンスに影響する要因」](#)に示す要因に依存します。

## AOM コンポーネント内での Siebel Configurator の実行

Siebel Call Center などの AOM コンポーネント内で Siebel Configurator を実行できます。

少数の同時ユーザーが設定セッションを要求する場合、または少数のカスタマイズ可能製品モデルが存在する場合に、この展開オプションによって、妥当なパフォーマンスが得られるとともに、ハードウェアリソースを最も有効に使用できます。

このオプションによって、適切な各 AOM 上で Siebel Configurator キャッシュを管理するためのすべてのパラメータを設定します。詳しくは、[106 ページの「Siebel Configurator のキャッシュについて」](#)を参照してください。

## 専用サーバー上での Siebel Configurator の実行

AOM以外のサーバーコンポーネントを使って、1台以上の専用Siebel Serverマシン上でSiebel Configuratorを実行できます。このコンポーネントは、Siebel Product Configuration Object Manager（日本語版のエイリアス eProdCfgObjMgr\_jpn）です。

この一般的なトポロジオプションには、次のようなバリエーションが考えられます。

- 1つのAOMコンポーネントについて1つのeProdCfgObjMgrコンポーネントを実行する
- 1つのAOMコンポーネントについて複数のeProdCfgObjMgrコンポーネントを実行する
- 複数のAOMコンポーネントについて1つのeProdCfgObjMgrコンポーネントを実行する

多数の同時ユーザーが設定セッションを要求する場合、または多数のカスタマイズ可能製品モデルが存在する場合に、この展開オプションで1台以上の専用サーバーを使用することによって、最善のパフォーマンスが得られるとともに、ハードウェアリソースを最も有効に使用できます。

このオプションでは、適切なAOM上および個別の適切な専用Siebel Configuratorサーバー上でSiebel Configurator キャッシュを管理するためのいくつかのパラメータを設定します。詳しくは、[106ページの「Siebel Configuratorのキャッシュについて」](#)を参照してください。

## 専用 Siebel Configurator の展開用の AOM の設定

Siebel Product Configuration Object Manager（日本語版のエイリアス eProdCfgObjMgr\_jpn）コンポーネントを実行するための専用サーバーマシンを1台以上指定する場合、ユーザーがそのサーバーマシンに設定リクエストを送るための設定セッションの開始元になる、任意のAOMコンポーネントを設定する必要があります。

AOMは、最も同時ユーザーの少ない専用Siebel Configuratorサーバーに各設定セッションリクエストを転送します。

表3は、専用Siebel Configuratorの展開を管理するためのサーバーパラメータのリストです。これらのパラメータは、Server Managerを使って各AOM上で設定します（専用Siebel Configuratorサーバーマシン上では設定しません）。

表3 専用 Siebel Configurator サーバーの展開のためのサーバーパラメータ

パラメータ名	表示名	データタイプ	デフォルト値	説明
eProdCfgRemote	Product Configurator-Use remote service	Boolean	FALSE	1台以上の専用サーバー上でeProdCfgObjMgrコンポーネントを実行している場合は、このパラメータをTRUEに設定します。  AOMだけを使用するSiebel Configuratorの展開では、このパラメータを「FALSE」に設定します。

表 3 専用 Siebel Configurator サーバーの展開のためのサーバーパラメータ

パラメータ名	表示名	データタイプ	デフォルト値	説明
eProdCfgServer	Product Configurator-Remote Server Name	テキスト		Siebel Configurator サーバーに対する製品のマッピングを明示的に有効にしていない場合、このパラメータは、eProdCfgObjMgr を実行している専用マシンの名前に設定します。それ以外の場合は、このパラメータの値を「NULL」に設定します。
eProdCfgTimeOut	Product Configurator-Time out of connection	Integer	20	AOM が eProdCfgObjMgr を実行中の専用 Siebel サーバーに接続を試みる時間をミリ秒単位で設定します。 タイムアウトに達すると、ユーザーにエラーが返されます。

## Siebel Configurator のチューニングを行うための最善の方法

ハードウェアリソースを最適に使用し、システムを適切に設定することにより、パフォーマンス面の目標を達成するために役立ちます。リソースと要件を慎重に検討し、システムパフォーマンスを継続的にテストおよび監視する必要があります。

詳しくは、『Siebel 製品管理ガイド』、『Siebel システム管理ガイド』、その他の資料を参照してください。

次に、パフォーマンスおよびスケーラビリティに関する目標を達成するために考えられる活動をいくつか示します。

- システムトポロジの調整。詳しくは、[101 ページの「Siebel Configurator のトポロジに関する注意事項」](#)を参照してください。
- Siebel Configurator の Siebel Server サーバーコンポーネントの設定。
- カスタマイズ可能製品モデルの設計と展開。詳しくは、[105 ページの「カスタマイズ可能製品のモデルとクラスの定義」](#)を参照してください。

ここでは、Siebel Web Client を使用する展開を対象としています。次のトピックが含まれています。

- [104 ページの「Siebel Configurator のチューニング」](#)
- [105 ページの「Siebel Configurator ファイルシステムの場所の指定」](#)
- [105 ページの「カスタマイズ可能製品のモデルとクラスの定義」](#)

## Siebel Configurator のチューニング

Siebel Configurator サーバーの展開での適切なチューニングに対応する Siebel Server コンポーネントの設定方法は、使用する展開方法に応じて部分的に異なります。詳しくは、[101 ページの「Siebel Configurator のトポロジに関する注意事項」](#)を参照してください。

- AOM 上で Siebel Configurator を展開する場合、AOM チューニングの計算とともに Siebel Configurator チューニングの計算を行う必要があります。
- 専用 Siebel Server マシン上で Product Configurator Object Manager (eProdCfgObjMgr) サーバーコンポーネントを使って Siebel Configurator を展開する場合、Siebel Configurator チューニングの計算と AOM チューニングの計算の関係は間接的なものに限られ、主に設定に関連する同時ユーザーとリクエスト負荷によって Siebel Configurator チューニングの計算が決定します。

特に、専用 Siebel Configurator サーバーの場合で注意する必要があるのは、一般的に AOM の場合よりも MaxTasks パラメータをかなり低く設定しなければならないことです。eProdCfgObjMgr のデフォルトの設定では、MaxMTServers と MaxTasks の比率は 20:1 です。

また、リクエストの負荷に応じて、一般的には Siebel Configurator を実行していない AOM よりも、Siebel Configurator を実行している AOM の MaxTasks をより低く設定する必要があります。

次の基本手順に従って、これらのパラメータの設定方法を決定できます。

- Siebel アプリケーションのユーザーのうち、Siebel Configurator のユーザーでもある人の割合を決定します。たとえば、100 ユーザーあたり 60 ユーザーが見積りを使用する場合を考えます。
- これらのユーザーが Siebel Configurator の使用に費やす時間の割合を計算します。たとえば、前述の 60 ユーザーのうち 30 ユーザーだけが Siebel Configurator を同時に使用する場合を例に取ります。
- MaxTasks/MaxMTServers のデフォルトの比率である 20:1 をそのまま使用します。

専用 Siebel Server マシンで eProdCfgObjMgr を使って Siebel Configurator を展開し、データベース接続（ログインとログアウト）が低速である場合は、次の処理を実行することをお勧めします。

- データベース接続プールを有効にする

接続プールを有効にするには、MaxSharedDbConns パラメータと MinSharedDbConns パラメータを、MaxTasks - 1 未満かつ 1 以上の正の整数値に設定します。

これによって、すべてのユーザー接続を共有せずにプールできるので、eProdCfgObjMgr セッションごとに新しいデータベース接続を作成して削除することを回避できます。

- サードパーティのユーザー認証を使用する

データベース認証の代わりに、LDAP などのサードパーティのユーザー認証を使用すると、認証のために追加のデータベース接続を作成する必要がなくなります。認証オプションについて詳しくは、『Siebel セキュリティガイド』を参照してください。

データベース接続プールについては、[37 ページの「AOM のデータベース接続プールの設定」](#)を参照してください。

## Siebel Configurator ファイルシステムの場所の指定

Siebel Configurator は、ファイルシステムのディレクトリを使用して、設定に関するすべてのオブジェクト定義をキャッシュします。この場所は、「Product Configurator - FS location (alias eProdCfgCacheFS)」サーバーパラメータによって決まります。このパラメータには、書き込みが許可されているサーバーディレクトリパスを示す値を指定します。例：¥¥MyServer¥SibFS¥SiebConfig

**備考：** この値には、ネットワークからアクセスできるディレクトリを指定する必要があります。

最上位レベルのディレクトリは指定しないでください。たとえば、SibFS が最上位レベルのディレクトリの場合は、SiebConfig などのサブディレクトリを指定します。

eProdCfgCacheFS に値を指定しない場合、Siebel Configurator は Siebel File System の使用を試みます。Siebel File System が File System Manager (エイリアス FSMSrvr) コンポーネントを使用する場合、Siebel Configurator はファイルシステムにオブジェクト定義をキャッシュしません。Siebel File System については、『Siebel システム管理ガイド』を参照してください。

## カスタマイズ可能製品のモデルとクラスの定義

ここでは、パフォーマンスを最適化するためにカスタマイズ可能製品およびそのクラスを作成する場合に役立つガイドラインをいくつか紹介します。

- 良好なパフォーマンスを維持するには、絶対に必要でない限り、カスタマイズ可能製品またはそのクラスを不用意に大きくしたり、複雑にしたりしないようにします。
- 複雑さは、カスタマイズ可能製品モデルに組み込まれた階層レベルと制約、およびそのクラスの構造の階層レベルと制約を表す数値の関数です。
- クラスの関係の定義では、可能な限り特定のクラスを使用します。たとえば、クラスを指定しないでクラス関係を定義しないようにします。また、親クラスとサブクラスが定義されている場合は、親クラスではなくサブクラスを使用します。
- カスタマイズ可能製品モデルに関連付けるユーザーインターフェースの要素の複雑さを可能な限り軽減します。
- 一般的に、カスタマイズ可能製品の対話的または自動的な価格更新を使用することをお勧めします。パフォーマンスに悪影響がある場合は、手動による価格更新に切り替えることを考慮します。
- ルールの作成時に Set Preference テンプレートを使用すると、柔軟な制約を作成できます。この制約は、Siebel Configurator エンジンが解決策を生成する際のガイドとなりますが、コンフリクトやパフォーマンス上の問題を避ける必要がある場合は無視できます。
- デフォルトでは、たとえばカスタマイズ可能製品を見積りに追加するときに、デフォルトの製品と選択内容が含まれます。また、このデフォルトのインスタンスを作成するために Siebel Configurator が起動されることがあります。カスタマイズ可能製品のデフォルトの選択内容のサイズが大きく、複雑になり、ユーザーが即座に製品をカスタマイズする必要がある場合は、このデフォルトインスタンス作成機能をオフにすることにより、機能の喪失を生じることなくパフォーマンスが向上します。

この問題について詳しくは、『Siebel 製品管理ガイド』を参照してください。

## Siebel Configurator のキャッシュについて

Siebel Configurator では、設定セッションユーザーの応答時間最適化のために、カスタマイズ可能製品の情報に対するさまざまなキャッシュがサポートされています。次のキャッシュオプションがあります。

### ■ メモリ内キャッシュ

Siebel Configurator では、カスタマイズ可能製品、製品クラス、および属性定義オブジェクトの複数のバージョンがメモリにキャッシュされます。このキャッシュがサイズの上限に到達すると、LRU (Least Recently Used) 方式で古いオブジェクトのバージョンから破棄されます。詳しくは、[107 ページの「Siebel Configurator のデフォルトのキャッシュ動作」](#)を参照してください。

### ■ Siebel Configurator ファイルシステムのキャッシュ

このディレクトリには、メモリに読み込まれたカスタマイズ可能製品、製品クラス、および属性定義オブジェクトの複数のバージョンがキャッシュされます。これはデフォルトの動作です。詳しくは、[107 ページの「Siebel Configurator のデフォルトのキャッシュ動作」](#)を参照してください。

### ■ 前述のキャッシュオプションに加えて、カスタマイズ可能製品、製品クラス、および属性定義オブジェクトの複数のバージョンをキャッシュするサーバーまたはコンポーネントを指定することもできます。

指定したキャッシュは定期的に更新できます。このオプションを使用すると、特定のカスタマイズ可能製品のリクエスト応答時間を短縮できます。詳しくは、[108 ページの「Siebel Configurator のキャッシュ管理」](#)を参照してください。

**備考:** Siebel Configuratorサーバーマシンのメモリリソースがキャッシュ処理の要件に十分に対応できることが必要です。

前述のトピックに加えて、ここでは次のトピックについて説明します。

- [109 ページの「Siebel Configurator のキャッシュの設定パラメータ」](#)
- [112 ページの「キャッシュパラメータのサイズ決定」](#)
- [112 ページの「Siebel Configurator キャッシュの管理」](#)

## Siebel Configurator のデフォルトのキャッシュ動作

Siebel Configurator のキャッシュのデフォルトの動作は次のとおりです。

- ユーザーが設定セッションを開始すると、Siebel Configurator はキャッシュ内のオブジェクトに影響する新しいキャッシュ更新リクエストを探します。キャッシュ内のオブジェクトに影響する新しいキャッシュ更新リクエストがある場合、Siebel Configurator は該当するオブジェクトを更新または削除します。
- Siebel Configurator は、リクエストされたカスタマイズ可能製品がメモリにキャッシュされているかどうかを確認します。
- カスタマイズ可能製品がまだメモリにキャッシュされていない場合、Siebel Configurator は Configurator ファイルシステムを探します。

**備考：** Configurator ファイルシステムの場所は、「Product Configurator - FS location parameter (alias eProdCfgCacheFS)」パラメータの値によって決まります。eProdCfgCacheFS に値を指定しない場合、Siebel Configurator は Siebel File System を探します。Configurator ファイルシステムについて詳しくは、105 ページの「Siebel Configurator ファイルシステムの場所の指定」を参照してください。

- カスタマイズ可能製品が Configurator ファイルシステムにない場合、Siebel データベースからロードされます。カスタマイズ可能製品は、メモリキャッシュと Configurator ファイルシステムに追加されます。
- これ以降に設定セッションが開始されると、カスタマイズ可能製品はメモリキャッシュまたは Configurator ファイルシステムからロードされます。
- システムは、Configurator ファイルシステムからカスタマイズ可能製品をロードする前に、Siebel データベースをチェックして、製品内の各項目が最新バージョンであることを確認します。
- データベース内でキャッシュされた製品が変更されている場合、最新バージョンの項目がデータベースからロードされます。これにより、カスタマイズ可能製品とそのコンテンツの最新バージョンをロードできます。
- 製品管理者がカスタマイズ可能製品の新しいバージョンをリリースすると、Siebel データベースに変更内容が書き込まれ、変更されたカスタマイズ可能製品に対するキャッシュ更新リクエストが送信されます。カスタマイズ可能製品に対する次の設定セッションが要求されるまで、メモリキャッシュおよび Configurator ファイルシステムは、その変更内容によって更新されません。

**備考：** カスタマイズ可能製品のルールでは、開始日と終了日は使用しないことをお勧めします。日付を受け取ってもキャッシュ内のカスタマイズ可能製品は更新されません。

## Siebel Configurator のキャッシュ管理

ユーザーが設定セッションを開始すると、Siebel Configurator は要求されたカスタマイズ可能製品をメモリにロードします。頻繁に要求されるカスタマイズ可能製品の応答時間を短縮するためのキャッシュ（サーバーまたはコンポーネント）を指定できます。そのためには、指定したキャッシュに対して「明示的な製品マッピングのみ」を選択する必要があります。指定したキャッシュは、ユーザー要求を受け取る前にメモリにマップされたカスタマイズ可能製品をロードします。

指定したキャッシュのカスタマイズ可能製品を定期的に更新する間隔も指定できます。これによって、ユーザー要求によってデータベースからデータを取得し、変更されたカスタマイズ可能製品をロードする必要性が軽減されます。間隔を指定するには、eProdCfgObjMgr コンポーネントの次のパラメータに値を設定します。

- サーバーセッションループスリープ時間（エイリアス ServerSessionLoopSleepTime）
- Product Configurator - Cache Engine Objects (alias eProdCfgCacheEngineObjects)

これらのパラメータについて詳しくは、109 ページの表 4 を参照してください。

特定のキャッシュにマップされていないその他のカスタマイズ可能製品へのリクエストは、「明示的な製品マッピングのみ」設定が無効になっているキャッシュによって処理されます。

次の手順では、「明示的な製品マッピングのみ」設定が有効になっているキャッシュに製品をマップすることによって製品のキャッシュを設定する方法について説明します。

### 製品のキャッシュを設定するには

- 1 [管理-商品] > [キャッシュ管理] に移動します。

[キャッシュ管理] ビューが表示されます。

- 2 [キャッシュ] アプレットで、キャッシュを選択します。

**備考：**一度に有効にできるキャッシュは 1 つだけです。

- 3 [キャッシュタイプ] フィールドで、次の一覧の説明に従って値を選択します。

- サーバー

設定リクエストをキャッシュに関連付けられている Siebel Server にルーティングする場合は、キャッシュタイプとして「サーバー」を選択します。

- コンポーネント

設定リクエストをキャッシュに関連付けられているコンポーネントにルーティングする場合は、キャッシュタイプとして「コンポーネント」を選択します。これらのコンポーネントは、コンポーネントがアクティブになっている場所によって、複数の Siebel Server にまたがることもあります。

**備考：**キャッシュタイプとして「コンポーネント」を選択する場合、AOM と eProdCfgObjMgr コンポーネントの両方に対して、「内部負荷分散を有効にする（エイリアス EnableVirtualHosts）」コンポーネントパラメータに同じ値を設定する必要があります。たとえば、AOM コンポーネントで EnableVirtualHosts を「TRUE」に設定した場合は、eProdCfgObjMgr コンポーネントでも「TRUE」に設定する必要があります。

- 4 [コンポーネント] アプレットで、Siebel Server 名またはコンポーネント名を指定して、[ステップ 2](#) で選択したキャッシュに関連付けます。  
指定する値は、[ステップ 2](#) で選択したキャッシュタイプの値に依存します。たとえば、キャッシュタイプに [サーバー] を選択した場合は、Siebel Server の名前を入力します。キャッシュタイプに [コンポーネント] を選択した場合は、コンポーネントの名前を入力します。
- 5 サーバーキャッシュまたはコンポーネントキャッシュを、それぞれのキャッシュにマップされている製品のみを使用する場合は、[明示的な製品マッピングのみ] を選択します。
- 6 [製品] アプレットで、[ステップ 4](#) で選択したコンポーネントに関連付ける製品を選択します。
- 7 [キャッシュ] アプレットで、[確認] をクリックします。  
アプリケーションは、選択した Siebel Server 名またはコンポーネント名が指定したキャッシュタイプに対して有効かどうかを確認します。
- 8 作成した設定が正しいことが確認された場合は、[リリース] をクリックし、選択したキャッシュ（キャッシュにマップした製品のキャッシュインスタンス）を有効にします。

## Siebel Configurator のキャッシュの設定パラメータ

Siebel Configurator のキャッシュは、デフォルトで有効になっています（eProdCfgSnapshotFlg が「TRUE」に設定）。その他のパラメータは、[112 ページの「キャッシュパラメータのサイズ決定」](#)で説明する要領で、次のガイドラインに従って調整する必要があります。

[表 4](#) に、Siebel Configurator のキャッシュを設定するサーバーパラメータのリストを示します。これらのパラメータは、Siebel Configurator の AOM での展開では、AOM コンポーネントに設定します。専用 Siebel Configurator サーバーの展開では、AOM と eProdCfgObjMgr コンポーネントの両方でこのパラメータを設定します。

サーバーパラメータの設定方法については、『Siebel システム管理ガイド』を参照してください。

表 4 Siebel Configurator キャッシュの動作を設定するためのサーバーパラメータ

パラメータエイリアス	パラメータ名	データタイプ	デフォルト値	説明
eProdCfgCacheFS	Product Configurator - FS location	String		Configurator ファイルシステムの場所を指定します。eProdCfgCacheFS に値を指定しない場合、Configurator は Siebel File System を探します。Configurator ファイルシステムについて詳しくは、 <a href="#">105 ページの「Siebel Configurator ファイルシステムの場所の指定」</a> を参照してください。

表 4 Siebel Configurator キャッシュの動作を設定するためのサーバーパラメータ

パラメータエイリアス	パラメータ名	データタイプ	デフォルト値	説明
eProdCfgAttrSnapshotFlg	Product Configurator - Collect and Use the snapshots of the ISS_ATTR_DEF Ob	Boolean	TRUE	属性定義のキャッシュを有効にする場合は「TRUE」に設定します。これによって、属性定義がメモリにキャッシュされます。このパラメータは、変更しないことを強くお勧めします。
eProdCfgNumOfCachedAttrs	Product Configurator - Number of Attribute Definitions Cached in Memory	Integer	100	設定中にメモリに保持する属性定義の数を設定します。
eProdCfgClassSnapshotFlg	Product Configurator - Collect and Use the snapshots of ISS_CLASS_DEF Ob	Boolean	TRUE	製品クラス定義のキャッシュを有効にする場合は「TRUE」に設定します。このパラメータは、変更しないことを強くお勧めします。
eProdCfgNumOfCachedClasses	Product Configurator - Number of Class Definitions Cached in Memory	Integer	100	設定中にメモリに保持するクラス定義の数を設定します。
eProdCfgProdSnapshotFlg	Product Configurator - Collect and Use the snapshots of ISS_PROD_DEF Ob	Boolean	TRUE	製品定義のキャッシュを有効にする場合は「TRUE」に設定します。これによって、製品定義がメモリにキャッシュされます。このパラメータは、変更しないことを強くお勧めします。
eProdCfgNumOfCachedProducts	Product Configurator - Number of Product Definitions Cached in Memory	Integer	1000	設定中にメモリに保持する製品定義の数を設定します。

表 4 Siebel Configurator キャッシュの動作を設定するためのサーバーパラメータ

パラメータエイリアス	パラメータ名	データタイプ	デフォルト値	説明
eProdCfgSnapshotFlg	Product Configurator-Collect and use snapshots of the Cfg objects	Boolean	TRUE	Siebel Configurator キャッシュをオンにする場合は「TRUE」に設定します。このパラメータは、変更しないことを強くお勧めします。
eProdCfgNumbOfCachedCatalogs	Product Configurator-Number of cached catalogs	Integer	10	メモリ内にキャッシュできるモデル管理カタログの最大数を設定します。 <b>備考：</b> このパラメータは、以前のリリースのパラメータ eProdCfgNumbOfCachedFactories による機能を提供します。
eProdCfgNumbOfCachedWorkers	Product Configurator-Number of workers cached in memory	Integer	50	メモリ内にキャッシュできるワーカーの最大数を設定します。この数は、すべてのモデル管理カタログに適用されます。
eProdCfgCacheEngineObjects	Product Configurator - Cache Engine Objects	Boolean	TRUE	コンテンツキャッシュとプレキャッシュを有効にする場合は「TRUE」に設定します。
ServerSessionLoopSleepTime	ServerSessionLoopSleepTime	Integer	300	明示的な製品マッピング設定を使用して、Configurator サーバーキャッシュまたはコンポーネントキャッシュにマップされているキャッシュ済みの製品を更新する間隔(秒)を指定します。 <b>備考：</b> eProdCfgCacheEngineObjects を「TRUE」に設定する必要があります。

## キャッシュパラメータのサイズ決定

Siebel Configurator のキャッシュパラメータの設定方法を定める場合、一般的にはカスタマイズ可能製品で必要になるメモリの増加分を測定することが役立ちます。

オブジェクトのキャッシュの要件よりも、モデル管理およびワーカーのキャッシュの要件のほうが大きく影響します。オブジェクトのキャッシュの要件は小さく、複数のユーザーに適用されます。モデル管理のキャッシュは複数のユーザーに適用されます（同じカスタマイズ可能製品を使用します）。また、ワーカーのキャッシュも複数のユーザーに適用されます。

見積りまたはオーダーに含まれているカスタマイズ可能製品の [カスタマイズ] をクリックした前後で siebel.exe プロセスによって使用されているメモリを確認し、さらに（概算の最大メモリ使用量に到達させるために）カスタマイズ可能製品を設定した後に再びこのメモリを確認することによって、Siebel Developer Web クライアント（専用データベース接続を使用するモバイル Web クライアント）上で、このようなメモリの増加分の測定を試みることができます。

たとえば、X がロード前のメモリサイズ、Y がロード後のメモリサイズ、Z が追加製品設定後のメモリサイズである場合を考えます。

認識されたメモリの増加分について、さらに次の分類を検討します。

- カスタマイズ可能製品のモデル管理のサイズは、製品のインスタンスを作成するのに必要なメモリの増加分の約 75% です（つまり  $Y - X$  の 75%）。
- カスタマイズ可能製品のワーカーのサイズは実行時に変化し、一般的にはユーザーが選択を行うたびに増加します。このサイズは、Z から X を減算した結果から、モデル管理のサイズを減算することによって近似できます。

## Siebel Configurator キャッシュの管理

Siebel 管理者または製品管理者は、各種の方法を使用して Siebel Configurator キャッシュをリフレッシュまたは更新できます。次の項では、カスタマイズ可能製品、製品クラス、および属性定義の変更によって Siebel Configurator キャッシュをリフレッシュまたは更新する手順について説明します。詳しくは、次の項を参照してください。

- [113 ページの「Siebel Configurator キャッシュ全体のリフレッシュ」](#)
- [113 ページの「製品の変更による Siebel Configurator キャッシュのリフレッシュ」](#)
- [114 ページの「製品クラスの変更による Siebel Configurator キャッシュの更新」](#)
- [114 ページの「製品クラスの変更による Siebel Configurator キャッシュのリフレッシュ」](#)
- [115 ページの「属性定義の変更による Siebel Configurator キャッシュの更新」](#)
- [115 ページの「属性定義の変更による Siebel Configurator キャッシュのリフレッシュ」](#)

## Siebel Configurator キャッシュ全体のリフレッシュ

ここでは、カスタマイズ可能製品、製品クラス、および属性定義の変更によって Siebel Configurator キャッシュを一度の操作でリフレッシュする方法について説明します。開発環境から本番環境にデータを移行する場合などは、このタスクを Siebel Configurator の展開の一部として実行することを検討します。

この項の他のトピックでは、カスタマイズ可能製品、製品クラス、および属性定義の変更によって Siebel Configurator キャッシュを個別の操作で更新する方法について説明します。詳しくは、[112 ページの「Siebel Configurator キャッシュの管理」](#)を参照してください。

### Siebel Configurator キャッシュ全体をリフレッシュするには

- 1 [管理-商品] > [キャッシュ管理] に移動します。
- 2 キャッシュをリフレッシュするレコードを選択します。
- 3 [キャッシュ] リストのメニューボタンをクリックし、[製品キャッシュのリフレッシュ] を選択します。

## 製品の変更による Siebel Configurator キャッシュのリフレッシュ

製品レコードの編集集中に [製品キャッシュのリフレッシュ] を選択すると、製品管理者は、選択した製品の変更内容によって Siebel Configurator キャッシュをリフレッシュできます。ユーザーが次回にカスタマイズ可能製品を要求すると、ユーザーは製品の変更を反映して新規に作成されたインスタンスバージョンを受け取ります。また、キャッシュはこのバージョンによりリフレッシュされます。たとえば、製品の説明を変更してからキャッシュをリフレッシュします。

Siebel Configurator キャッシュのリフレッシュオプションについては、[112 ページの「Siebel Configurator キャッシュの管理」](#)を参照してください。

### 製品の変更でキャッシュをリフレッシュするには

- 1 [管理-商品] 画面に移動します。
- 2 変更されたカスタマイズ可能製品または更新されるカスタマイズ可能製品のレコードを選択します。
- 3 [製品] リストのメニューボタンをクリックし、[製品キャッシュのリフレッシュ] を選択します。

## 製品クラスの変更による Siebel Configurator キャッシュの更新

製品クラスのレコードの編集集中に [キャッシュを更新] を選択すると、製品管理者は Siebel Configurator キャッシュから、選択した製品クラスのすべてのバージョンについて、キャッシュされた製品クラスのバージョンアクセスキーを削除できます。これによって、アプリケーションは次回バージョンアクセスキーが必要になった場合に、データベースを確認するようになります。製品名などのバージョン化されていないキャッシュされたプロパティを変更する場合、製品管理者は [キャッシュを更新] を選択する必要があります。製品管理者が [キャッシュを更新] を選択しない場合、すでにキャッシュされているバージョンのある AOM は古いバージョンを使用します。

Siebel Configurator キャッシュの管理オプションについては、[112 ページの「Siebel Configurator キャッシュの管理」](#) を参照してください。

### クラスの変更によってキャッシュを更新するには

- 1 [管理-商品] > [商品クラス] ビューに移動します。  
[製品クラス] リストアプレットが表示されます。
- 2 製品クラスを選択し、必要に応じてその属性の定義を変更します。
- 3 [製品クラス] リストのメニューから [キャッシュを更新] を選択します。

## 製品クラスの変更による Siebel Configurator キャッシュのリフレッシュ

製品クラスのレコードの編集集中に [キャッシュのリフレッシュ] を選択すると、製品管理者は Siebel Configurator キャッシュのカスタマイズ可能製品を製品クラスのレコードの変更内容によって更新できます。ユーザーが次回にカスタマイズ可能製品を要求すると、ユーザーは製品の変更を反映して新規に作成されたインスタンスバージョンを受け取ります。また、キャッシュはこのバージョンによりリフレッシュされます。この新しいインスタンスには、製品クラスの変更が反映されます。

Siebel Configurator キャッシュのリフレッシュオプションについては、[112 ページの「Siebel Configurator キャッシュの管理」](#) を参照してください。

### クラスの変更でキャッシュをリフレッシュするには

- 1 [管理-商品] > [商品クラス] ビューに移動します。  
[製品クラス] リストアプレットが表示されます。
- 2 製品クラスを選択し、必要に応じてその属性の定義を変更します。
- 3 [製品クラス] リストのメニューから [キャッシュのリフレッシュ] を選択します。

## 属性定義の変更による Siebel Configurator キャッシュの更新

属性定義レコードの編集集中に [キャッシュを更新] を選択すると、製品管理者は Siebel Configurator キャッシュから、選択した属性定義のすべてのバージョンについて、属性定義レコードのバージョンアクセスキーを削除できます。これによって、アプリケーションは次回バージョンアクセスキーが必要になった場合に、データベースを確認するようになります。属性定義名などのバージョン化されていないキャッシュされたプロパティを変更する場合、製品管理者は [キャッシュを更新] を選択する必要があります。製品管理者が [キャッシュを更新] を選択しない場合、すでにキャッシュされているバージョンのある AOM は古いバージョンを使用します。

Siebel Configurator キャッシュの管理オプションについては、[112 ページの「Siebel Configurator キャッシュの管理」](#) を参照してください。

### 属性定義の変更によってキャッシュを更新するには

- 1 [管理 - 商品] > [属性定義] ビューに移動します。  
[属性定義] リストアプレットが表示されます。
- 2 属性定義を選択し、必要に応じて変更します。
- 3 [属性定義] リストのメニューから [キャッシュを更新] を選択します。

## 属性定義の変更による Siebel Configurator キャッシュのリフレッシュ

属性定義レコードの編集集中に [キャッシュのリフレッシュ] を選択すると、製品管理者は Siebel Configurator キャッシュのカスタマイズ可能製品を属性定義レコードの変更内容によって更新できます。

ユーザーが次回にカスタマイズ可能製品を要求すると、ユーザーは属性定義の変更を反映して新規に作成されたインスタンスバージョンを受け取ります。また、キャッシュはこのバージョンによりリフレッシュされます。この新しいインスタンスには、属性定義の変更が反映されます。

Siebel Configurator キャッシュのリフレッシュオプションについては、[112 ページの「Siebel Configurator キャッシュの管理」](#) を参照してください。

詳しくは、[112 ページの「Siebel Configurator キャッシュの管理」](#) を参照してください。

### 属性定義の変更によってキャッシュをリフレッシュするには

- 1 [管理 - 商品] > [属性定義] ビューに移動します。  
[属性定義] リストアプレットが表示されます。
- 2 属性定義を選択し、必要に応じて変更します。
- 3 [属性定義] リストのメニューから [キャッシュのリフレッシュ] を選択します。



# 9

## パフォーマンスのための Siebel EAI のチューニング

ここでは、最適なパフォーマンスを得るために必要な Siebel Enterprise Application Integration (Siebel EAI) のチューニングについて説明します。ここで説明する内容は、次のとおりです。

- 117 ページの「[Siebel Enterprise Application Integration について](#)」
- 118 ページの「[Siebel EAI のチューニングを行うための最善の方法](#)」

Siebel EAI について詳しくは、『Siebel Bookshelf』の次のマニュアルを参照してください。

- 『Overview: Siebel Enterprise Application Integration』
- 『Integration Platform Technologies: Siebel Enterprise Application Integration』
- 『Transports and Interfaces: Siebel Enterprise Application Integration』
- 『Business Processes and Rules: Siebel Enterprise Application Integration』
- 『XML Reference: Siebel Enterprise Application Integration』

## Siebel Enterprise Application Integration について

Siebel EAI には、Siebel Business Applications と外部アプリケーションおよび社内テクノロジーを統合するためのコンポーネントがあります。Siebel EAI は、XML、HTTP、Java、およびサードパーティ製のさまざまなミドルウェア製品およびアプリケーション統合ソリューションを含むテクノロジー、標準、またはアプリケーションと連動します。

Siebel EAI は、Siebel アプリケーションと他のアプリケーションを統合するための双方向リアルタイムソリューションおよびバッチソリューションを提供します。Siebel EAI は、相互間の対話および Siebel アプリケーション内の他コンポーネントとの対話を行うインターフェイスセットとして機能するように設計されています。

## Siebel EAI のチューニングを行うための最善の方法

ここでは、Siebel EAI を使用して許容可能なパフォーマンスを維持するための最善の方法について説明します。

まず、一般的なガイドラインについて説明します。その後、Siebel EAI と一緒に使用される IBM WebSphere MQ (以前の MQSeries) Transport アダプタ、HTTP Inbound Transport アダプタ、EAI Siebel アダプタ、仮想ビジネスコンポーネント、Workflow Process Manager などの Siebel EAI 機能に固有な推奨事項について説明します。

次の一般的なガイドラインに従うことにより、データ統合の全体的なパフォーマンスと Siebel Business Applications のスループットが向上します。

- システム間のやり取りを最小化します。たとえば、統合で 3 つのデータが必要な場合に、1 つのデータのリクエストを送信し、応答を待機し、次のリクエストを送信するといったやり取りは避けてください。複数のデータが必要な場合でも、1 回のリクエストでデータを収集します。
- サーバーコンポーネント間の呼び出しを回避するために、可能な限り単一セッション内で処理を続けます。
- セッション内で、ワークフロー、スクリプト、EAI Siebel アダプタなどのコンポーネント間で呼び出しのネストを最小にします。たとえば、ワークフロープロセスを使って、ビジネスサービスの呼び出しをシーケンス化し、スクリプトコードの手順の独立性を維持します。ワークフローサブプロセスは、よく呼び出される一連のサービスをパッケージ化するために使用できます。
- 可能であれば、スクリプトの代替手段を使用します。スクリプトを使用する場合は、必要最小限かつ経済的に使用し、説明されているガイドラインに従います。詳しくは、[179 ページの「Siebel スクリプトに関する最善の方法」](#)を参照してください。
- ビジネスコンポーネント、ビジネスサービス、キャッシュ、および統合処理をサポートするその他のアプリケーション機能を、最適なパフォーマンスが得られるように設定します。詳しくは、この章のその他の項および [第 12 章「パフォーマンスのための顧客設定のチューニング」](#)を参照してください。
- 統合処理をサポートするすべてのサーバーに対して容量計画を実行します。サイズの検討については、Siebel エキスパートサービスにお問い合わせください。
- Siebel アプリケーションが内部的に使用する同じコードページとエンコード (UCS-2) で外部からの着信データを表します。これによって、ワークフロープロセス内で Transcode ビジネスサービスを使用する必要性がなくなり、パフォーマンスが向上します。

次のいくつかのトピックでは、個々の技術と、パフォーマンスを向上させる手段について、分野ごとに説明します。詳しくは、次の資料を参照してください。

- [119 ページの「IBM WebSphere MQ Transport パフォーマンスの向上」](#)
- [121 ページの「HTTP Inbound Transport のパフォーマンスの向上」](#)
- [121 ページの「EAI Siebel アダプタのパフォーマンス」](#)
- [123 ページの「仮想ビジネスコンポーネントのパフォーマンス」](#)
- [124 ページの「Workflow Process Manager のパフォーマンスの向上」](#)
- [125 ページの「Siebel EAI に関するその他の最善の方法」](#)

## IBM WebSphere MQ Transport パフォーマンスの向上

IBM WebSphere MQ キューのパフォーマンスは、キューマネージャマシンのディスクのパフォーマンスと、ディスク上のキューファイルのレイアウトに大きく依存します。実際のアプリケーションで想定される最も高いパフォーマンスを得るために、スタンドアロンユーティリティによりキューをテストする必要があります。

より高いスループットを実現するには、次のオプションを考慮します。

- **複数の MQ Receiver タスクを実行します。** 同じマシン上または複数のマシン間で、複数の MQ Receiver タスクを並列に実行します。MQ Receiver タスクの最適数は、トランザクションのタイプに依存します。

**備考：** このマニュアルでは、使用する実際の Siebel Server コンポーネントが MQSeries Server Receiver (エイリアス MqSeriesSrvRcvr) または MQSeries AMI Receiver (エイリアス MqSeriesAMIRcvr) である場合に、「MQ Receiver」という用語を使用します。

MQ Receiver タスクのデフォルト数は 1 です。トランザクションの性質と使用可能なサーバーの容量に応じて、この数を 10 以上に設定することもできます。

一般的には、MQ Receiver を追加すると、キュー解除レートが低く MQ 競合が検出されない CPU 負荷の高いトランザクションで非常に効果的です。

複数の MQ Receiver が同じ MQ キューマネージャまたはキューに接続する場合など、MQ Receiver タスクを追加した後でも依然として競合が検出されることがあります。詳しくは、次の項目を参照してください。

- **複数の MQ キューマネージャを実行します。** MQ Receiver タスクを追加してもその効果が小さくなってきた場合は、MQ キューマネージャを追加実行すると効果が得られることがあります。これは、ディスク上の物理フォルダに保存された MQ リソースの競合を回避するのに役立ちます。
- **永続的なキューが不要な場合はそれをオフにします。** CPU 負荷の高くないトランザクションまたは永続的なキューに関連するパフォーマンスの問題は、MQ の競合に関係する 경우가多く、MQ Receiver を追加してもあまり効果は得られません。永続的なキューを必要としない場合は、永続的なキューをオフにします。

永続的なキューは、WebSphere MQ の標準キューよりも低速です。しかし、永続的なキューを使用しない場合、キューマネージャがダウンすると、メッセージが喪失します。

- **Maximum Number of Channels パラメータを設定します。** WebSphere MQ キューマネージャの Maximum Number of Channels パラメータを実行中の同時クライアントの最大数以上に設定します。

さらに、次に説明するように、WebSphere MQ Transport の送受信時の転送パフォーマンスを向上させるための方法がいくつかあります。

### 受信メッセージ

受信 WebSphere MQ メッセージでは、複数の MQ Receiver を並行して実行することにより、スループットが向上します。詳しくは、前述した説明を参照してください。

## 送信メッセージ (Send, SendReceive)

WebSphere MQ Transport ビジネスサービスをキャッシュすることにより、メッセージごとにキューに接続する必要がなくなるため、送信のパフォーマンスが向上します。あらゆる状況でキャッシュを使用できるわけではないので、デフォルトではキャッシュは無効です。キャッシュを有効にする場合、次のヒントを参考にしてください。

- クライアントセッションのみをキャッシュに格納します。Workflow Process Manager (WfProcMgr) コンポーネント内で転送を呼び出す場合は、キャッシュを使用しないでください。このコンポーネントのスレッドモデルは、WebSphere MQ API と共に使用できません。
- ビジネスサービスのキャッシュを有効にするには、Siebel Tools 内で Cache プロパティを TRUE に設定し、SRF ファイルを再コンパイルします。
- Workflow Process Manager 内およびクライアントセッション内で WebSphere MQ Transport を呼び出す必要がある場合は、毎回サービスの別個のコピー（キャッシュされているサービスとキャッシュされていないサービス）を作成します。
- キューごとにキャッシュを行います。また、同時に 1 つの接続のみをオープンしたまま保持します。単一セッションで複数のキューとやり取りする場合は、送信キューごとに転送のコピーを作成することを考慮します。

**備考：** パフォーマンスおよびサイズのガイドラインについては、IBM WebSphere MQ マニュアルを参照します。

## パフォーマンスのイベント

EAItransportPerf イベントをレベル 5 に設定することにより、WebSphere MQ Transport のパフォーマンスに関する詳細な追跡情報が得られます。

Workflow Process Manager (WfProcMgr)、EAI Object Manager (EAIObjMgr)、MQ Receiver など、Siebel EAI 機能で何らかの役割を果たす複数の Siebel Server コンポーネントで、このイベントレベルを設定できます。たとえば、svrvmgr で MQ Receiver のイベントレベルを設定できます。

```
change evtloglvl EAItransportPerf=5 for comp MqSeriesSrvRcvr
```

## HTTP Inbound Transport のパフォーマンスの向上

HTTP Inbound Transport は、セッションモードおよびセッションレスモードの2つのモードをサポートしています。

- セッションモードでは、ログオフが呼び出されるまでセッションは有効なままです。
- セッションレスモードでは、リクエストごとにログインおよびログオフが自動的に行われます。

可能な場合は常に、セッションモードを使用すべきです。これは、アプリケーションにログインするのにかかる時間が通常の平均的リクエストを処理するのにかかる時間よりかなり長いからです。

また、SessPerSisnConn コンポーネントパラメータを使用することにより、Web サーバーと EAI Object Manager 間の同じ物理 SISNAPI 接続を共有するセッションの数を制御できます。

このパラメータを 1 に設定すると、Siebel セッションごとに専用の物理接続が提供されます。デフォルト値は 20 であり、この場合は最大 20 セッションが同じ SISNAPI 接続を共有できます。EAI Object Manager の場合は、SessPerSisnConn を 1 に設定することをお勧めします。SessPerSisnConn を 1 に設定するとセッション数が過剰になる場合は、SessPerSisnConn の値を増やすことを検討するか、ハードウェアリソースを追加してください。

エンタープライズレベルまたはサーバーレベルで、svrMgr を使ってこのパラメータを変更できます。たとえば、EAI Object Manager のパラメータをエンタープライズレベルで設定するには、次のコマンドを入力します。

```
change param SessPerSisnConn=1 for compdef eaiobjmgr_enu
```

SessPerSisnConn の設定方法については、[47 ページの「AOM の SISNAPI 接続プールの設定」](#)を参照してください。

## EAI Siebel アダプタのパフォーマンス

ここでは、EAI Siebel Adapter のパフォーマンスおよびスループットを向上させる方法を説明します。

### スクリプトのレビュー

EAI Siebel Adapter によって使用されるビジネスコンポーネント上でのイベントのスクリプト記述を避けます。EAI Siebel Adapter の呼び出し内でスクリプトタスクを実行するのではなく、EAI Siebel Adapter 呼び出しの前後でスクリプトタスクを実行します。

一般的なスクリプトのガイドラインについては、[179 ページの「Siebel スクリプトに関する最善の方法」](#)も参照してください。

### ログの無効化

パフォーマンスが重要になるプロセスが正しく機能している場合は、そのプロセスのログを無効にすることにより、パフォーマンスが 10% 高速化します。BypassHandler サーバーパラメータを TRUE に設定することにより、EAI Object Manager (または MQ Receiver など他の適切なサーバーコンポーネント) のログを無効化できます。

## インテグレーションオブジェクトのサイズの最小化

インテグレーションオブジェクトとその基盤となるビジネスコンポーネントのサイズが EAI Siebel Adapter のパフォーマンスに影響することがあります。この影響を最小限に抑えるには、次の方法があります。

- ビジネスオブジェクトおよびビジネスコンポーネントをコピーし、Siebel EAI コンテキストで必要のない要素（スクリプト、結合、マルチバリューフィールド、ユーザープロパティなど）を削除するように変更することを検討します。これらの比較的合理化されたオブジェクト定義に基づいてインテグレーションオブジェクトを作成します。クエリーの実行時に、インテグレーションオブジェクトのユーザーキーがインデックスを効率的に使用することを確認します。
- インテグレーションオブジェクト内の不要なインテグレーションコンポーネントと、インテグレーションコンポーネントフィールドを無効化します。ビジネスニーズに従って、メッセージ処理に必要なコンポーネントとフィールドのみをアクティブ化します。
- 基になっている各ビジネスコンポーネントに不要なフィールドを無効化します。不要なフィールドに対して、Force Active が「TRUE」に設定されている場合は、「FALSE」に設定します。Force Active を FALSE に設定すると、EAI Siebel Adapter が該当フィールドを処理しません。該当フィールドを無効化しない場合は、インテグレーションオブジェクトに実際にフィールドが含まれていないときでも、EAI Siebel Adapter がそのフィールドを処理します。

詳しくは、[188 ページの「アクティブなフィールドの数の制限」](#)を参照してください。

## EAI Siebel Adapter によって生成される SQL の分析

EAI Siebel Adapter へのリクエストにより、最終的には Siebel データベースに対して実行される SQL が生成されます。EAI Siebel Adapter 内で実行するコンポーネントのイベント SQL のレベルを 4 に設定することにより、実行中の SQL 文の追跡情報と各文のタイミングをミリ秒単位で取得できます。

イベント EAI SiebAdptPerf を 4 または 5 に設定することにより、各 EAI Siebel Adapter 処理のタイミングを取得できます。これによって、EAI Siebel Adapter の呼び出しとその関連 SQL を対応付けることができます。

この情報を取得した後、ログを調べ、平均よりも著しく長い時間がかかっている SQL 文を見つけます。このような文のパフォーマンスを向上させるには、ビジネスコンポーネントを見つけるか（おそらく不要な Join およびフィールドの削除が必要です）、または物理データベーススキーマを探します（おそらくインデックスの追加が必要です）。

**備考：** EAI Siebel Adapter はかなりの追加オーバーヘッドを必要とするため、EAI SiebAdptPerf イベントに関連した個々のログ値を追加することによっていくつかの操作にまたがる全体的なタイミング（TotalTimeForProcess イベントと同等）を決定することはできません。EAI SiebAdptPerf が高い値に設定されていると、オーバーヘッドはさらに大きくなります。本番システムでは、最適なパフォーマンスを得るため、このイベントをより低い値に設定してください。

## EAI Siebel Adapter の並列実行

スループットを向上させる一般的な方法の 1 つに、EAI Siebel Adapter の複数のインスタンスを並列に実行することがあります。

MQ Receiver の場合は、このために複数の MQ Receiver タスクを実行します。詳しくは、[119 ページの「IBM WebSphere MQ Transport パフォーマンスの向上」](#)を参照してください。

EAI Object Manager の場合は、このために MaxTasks、MaxMTServers、および MinMTServers の各パラメータを設定することにより、EAI Object Manager コンポーネントのより多数のマルチスレッドプロセス上で、より多数のスレッド（タスク）を実行します。また、複数の同時 HTTP セッションを開始します。EAI Siebel Adapter の各インスタンス間で行われる対話はわずかです。

Siebel Database Server の能力が十分に大きい場合、Siebel Server の CPU またはメモリのいずれかが限界値に達するまで、EAI Siebel Adapter のスケーラビリティの直線性が維持されます。

**注意：**2つのセッションが同じレコードの更新または挿入を同時に試みると、一方のセッションが成功し、もう一方のセッションでエラーが生成されます。したがって、EAI Siebel Adapter を並列に実行する場合は、複数のセッションで同じレコードの同時更新を防止する必要があります。データをパーティション化するか、EAI Siebel Adapter 処理を再試行することにより、エラーが発生した場合にそれを防止できます。

## ビジネスオブジェクトのキャッシュ

EAI Siebel Adapter は、デフォルトでビジネスオブジェクトをキャッシュします。デフォルトのキャッシュサイズは、5 オブジェクトです。キャッシュを使用することによって、実行ごとにビジネスオブジェクトを再作成する必要がないので、アダプタ上での以降の実行が大幅に高速化されます。

必要な場合にキャッシュのサイズを変更するには、EAI Siebel Adapter 上で BusObjCacheSize パラメータを使用します。ただし、大半の用途でキャッシュサイズは 5 オブジェクトで十分です。この数を大きくしすぎると、メモリの使用量が不必要に増加します。

## 仮想ビジネスコンポーネントのパフォーマンス

このタイプの統合では、遅延に対して特に注意する必要があります。これは、ユーザーが Virtual Business Component (VBC) の応答により画面上で統合 GUI コンポーネントが表示されるまで待機する必要があるためです。

統合に複数のリクエストが含まれる場合に仮想ビジネスコンポーネントのパフォーマンスを向上させるには、特定システムへのリクエストを単一バッチ内にまとめます。

## Workflow Process Manager のパフォーマンスの向上

ここでは、Workflow Process Manager コンポーネントのパフォーマンスの問題について説明します。

Siebel ワークフローのパフォーマンスについては、第 7 章「パフォーマンスのための Siebel ワークフローのチューニング」および『Siebel Business Process Framework : Workflow ガイド』を参照してください。

Workflow Process Manager は、タスクベースのサーバーコンポーネントです。したがって、リクエストごとに新しいスレッドが作成されます。ただし、ワークフロープロセスを起動する可能性がある Object Manager コンポーネント (EAI Object Manager や AOM など) のセッションは、以降のリクエストのためにキャッシュされ、再使用されます。システムのサイズを決定する場合、同時にアクティブになることが想定されるワークフロータスクの最大数を確認する必要があります。これにより、Siebel アプリケーションで作成される Object Manager セッションの最大数が決定します。一般には、可能であれば小さいワークフロープロセスを作成することをお勧めします。大きいワークフロープロセスの作成が避けられない場合は、そのワークフロープロセスをいくつかのサブプロセスに分割してください。

各タスクでの CPU およびメモリの正確な使用量は、ワークフロープロセスで実行されるアクションに依存します。本番環境内での CPU およびメモリの使用量を見積もるには、単一のタスクを実行し、そのリソースの使用量を測定します。そして、最大同時セッション数に基づいて見積りを行います。このような測定を行う場合は、セッションのキャッシュを考慮します。

多数のセッションを必要とする場合、複数の Siebel Server マシン上で Workflow Process Manager を実行することが必要になることがあります。そのときは、いくつかの Siebel Server にまたがる負荷分散要求に対して Siebel Server の負荷分散機能を使用することができます。1 サーバーで多数のタスク (たとえば 100 個以上) を実行することを計画している場合は、複数のマルチスレッドプロセスを実行することも良い方法です。

異なるタイプのワークフローを複数実行する場合は、各タイプのワークフローを個別のプロセス内で実行します。これによって、プロセスのタイプごとの全体的な CPU 使用量およびメモリ使用量が簡単に監視できます。

マルチスレッドプロセスの数と、1 プロセスあたりのタスク数は、パラメータ MaxMTServers (Maximum MT Servers)、MinMTServers (Minimum MT Servers)、および MaxTasks (Maximum Tasks) によって制御されます。

**備考:** これらのパラメータは、Siebel Server ごとに設定します。たとえば MaxMTServers は、各 Siebel Server マシン上で実行するマルチスレッドプロセスの数を表します。詳しくは、『Siebel システム管理ガイド』を参照してください。

### パフォーマンスのイベント

ワークフローを実行中のコンポーネントのイベント WfPerf を設定することにより、ワークフローのパフォーマンスに関する追跡情報を取得できます。イベント WfPerf のレベルを 4 に設定すると、全体的なプロセスの実行タイミングを取得できます。イベント WfPerf のレベルを 5 に設定すると、各手順のタイミングも同様に取得できます。

Siebel EAI 機能の一部としてワークフロープロセスを起動する際の Siebel Server コンポーネントでも、このイベントレベルを設定できます。たとえば、srvrMgr を使って MQ Receiver にこのイベントレベルを設定するには、次のコマンドを入力します。

```
change evtloglvl wfPerf=5 for comp MqSeriesSrvRcvr
```

これらのイベントは、ワークフローのパフォーマンスを測定するだけでなく、ワークフロー内で実行するビジネスサービスのパフォーマンスを測定するときにも役立つ場合があります。

## Siebel EAI に関するその他の最善の方法

Siebel EAI のパフォーマンスの最適化について、それぞれの展開に適用できるかどうかを次の点により確認してください。

- **マシン上のディスクを確認します。** 使用中のキューマネージャで予備テストを行うことにより、(複数のドライバを使って)1秒あたりにサポートできる送信回数と対応する MQ Receiver の数を確認します。IBM WebSphere MQ などのキューベンダーには、このようなテストを実行し、キュー自身の拡大可能なサイズを特定するためのテストプログラムが用意されています。特に、マシン上のディスク速度が重要です。

- **メッセージを最適化します。** メッセージ内では、必要なカラムのみを参照します。

- **小さなビジネスコンポーネントを作成します。** メッセージでは、実際のビジネスコンポーネントの一部分のみを使用できます。

使用中のビジネスコンポーネントのコピーを作成します。このコピーでは、最適化されたインテグレーションオブジェクトによって使用されるすべてのアクティブなフィールド、またはそれ以外のメッセージを正しく処理するためのフィールド(表示/非表示フィールドや状況フィールドなど)をアクティブなままにします。他のすべてのフィールドを非アクティブ化します。また、メッセージの処理で不要な Join の定義とマルチバリュールINK (MVL) を無効にします。

元のビジネスコンポーネントが大きくかつ複雑であり、統合の目的で不要な要素が含まれている場合があります。小さなビジネスコンポーネントと、最適化されたインテグレーションオブジェクトの作成時に生成されたビジネスオブジェクトおよびリンクを使用します。

ビジネスコンポーネントに、Force Active が TRUE に設定されたフィールドが存在する場合があります。Siebel Tools を使って、ビジネスコンポーネント内のフィールドの Force Active プロパティを確認します。このフィールドが必要ない場合は、Force Active を「FALSE」に設定します。

- **ユーザープロパティ All Mode Sort を FALSE に設定します。** 最適化されたビジネスコンポーネントのユーザープロパティ All Mode Sort に FALSE を設定します(まだ設定されていない場合)。この設定は、Siebel EAI とともに使用するために作成した小さなビジネスコンポーネントの場合にのみ行います。これは、All Mode Sort ユーザープロパティが行の取得順を変更し、このようなクライアントが適切または標準ではない場合があるためです。All Mode Sort について詳しくは、『Siebel Developer's Reference』を参照してください。

- **データベースクエリーを最適化します。** MQ Receiver プロセスによって生成されるクエリーをレビューし、それが最適化されていることを確認します。

- **ログのオフ** 不要なサーバー側ログをオフにします。



# 10 パフォーマンスのための Siebel EIM のチューニング

この章は、EIM のパフォーマンスの向上に推奨される最善の方法について説明しており、次の項で構成されています。

- 128 ページの「Siebel EIM について」
- 129 ページの「EIM アーキテクチャ計画の要件」
- 131 ページの「EIM の使用計画」
- 134 ページの「EIM を最適化するときの一般的なガイドライン」
- 138 ページの「EIM のパフォーマンスのトラブルシューティング」
- 150 ページの「EIM を最適化するときのデータベースのガイドライン」
- 160 ページの「EIM を最適化するためのデータ管理ガイドライン」
- 161 ページの「EIM を最適化するための実行パラメータガイドライン」
- 161 ページの「EIM タスク時の Siebel Server の監視」

## Siebel EIM について

Siebel Enterprise Integration Manager (EIM) は、Siebel データベースとその他の企業データソース間においてデータを転送する Siebel EAI コンポーネントグループのサーバーコンポーネントです。この情報交換は、EIM テーブルと呼ばれる中間テーブルにより行われます（以前のリリースでは、EIM テーブルはインターフェイステーブルと呼ばれていました）。EIM テーブルは、Siebel アプリケーションデータベースとその他のデータソース間におけるステージング領域として機能します。

EIM は、Siebel データベースに大量のデータをロードするための主要な手段です。データの一括インポート、更新、マージ、および削除を実行するには、EIM を使用する必要があります。

Siebel アプリケーションデータベースには、Siebel アプリケーションで使用されるアプリケーションテーブル（ベーステーブル）があります。他の企業データソース（外部データベース）から Siebel アプリケーションテーブルに渡されるデータは、EIM テーブルを経由する必要があります。そのため、Siebel データベースと外部データベース間で行われるデータ交換は、次の 2 つのパートに分けられます。

- 1 EIM テーブルへのデータのロード
- 2 Siebel EIM を実行し、データを EIM テーブルから Siebel ベーステーブルにインポート

**備考：** このデータ交換プロセスの最初のパートでは、EIM テーブルと呼ばれる中間テーブルを使用します。Siebel EIM を使用するのには、プロセスの 2 番目のパートだけです。

Siebel ユーザーインターフェイスからデータが入力されると、ビジネスコンポーネントのオブジェクトタイプで設定したプロパティが参照されます。ただし、EIM から Siebel ベーステーブルにデータが入力されると、テーブルのオブジェクトタイプで設定したプロパティが参照されます。

**備考：** Oracle では、ネイティブ SQL を使用してデータを直接 Siebel ベーステーブル（データを受信するためのテーブル）にロードすることをサポートしていないため、一括のインポート、エクスポート、マージ、および削除を実行するには、EIM を使用する必要があります。また、EIM では、空の文字列は NULL に変換されることに注意してください。

## EIM アーキテクチャ計画の要件

Siebel アプリケーションを使用する作業の前には、必ず実装の規模や複雑さについて考慮する必要があります。最初に Siebel の実装を行うときは、本番アプリケーションの動作に直接影響する問題の優先順位が低い場合があります。ただし、実装の最初の段階で行った決定は、Siebel アプリケーションのパフォーマンスやスケーラビリティだけでなく、全体のメンテナンスにも大きく影響します。

Siebel エキスパートサービスの Siebel 認定プリンシパルコンサルタントまたはアーキテクチャスペシャリストが、組織に最適な論理および物理アーキテクチャの設計にかかわることを強くお勧めします。これには、容量計画やシステムのサイズ設定、物理データベースのレイアウト、およびその他の主要なアーキテクチャ項目も含まれます。

詳しくは、次のトピックを参照してください。

- 129 ページの「データベースのサイズ設定のガイドライン」
- 130 ページの「データベースレイアウトのガイドライン（論理および物理）」

### データベースのサイズ設定のガイドライン

データベースについて決定するうえで最も重要な要素の 1 つは、全体のサイズです。計画段階では、リレーショナルデータベース管理システム (RDBMS) で要求されるシステムストレージ、ロールバックセグメントおよびコンテナ、一時ストレージスペース、ログファイル、およびその他のシステムファイル用のスペースと、Siebel アプリケーションのデータおよびインデックス用のスペースを割り当てる必要があります。システムに割り当てたスペースが少なすぎると、パフォーマンスに影響が現れ、極端な場合には、システム自体が停止することがあります。割り当てが大きすぎると、スペースが無駄になる場合があります。

データベースに必要なスペースは、サポートされるユーザーの総数とタイプによって異なります。これらの条件については、RDBMS ベンダーの技術マニュアルを参照することをお勧めします。

Siebel のデータとインデックスに必要な領域は、実装する機能と、その機能をサポートするデータの量および特性によって異なります。

データベースのサイズを正確に算出するプロセスは、多くの変数を使用する複雑なプロセスです。このとき、次のガイドラインに従います。

- Siebel Business Applications のユーザーの総数とタイプ（営業員が 500 人、販売管理者が 75 人など）を決定します。
- 実装する機能とそれをサポートするために必要なエンティティを決定します。一般に、サイズの大きいエンティティは次のとおりです。
  - 取引先
  - 活動
  - 担当者
  - 売上予想
  - 商談
  - サービスリクエスト

- ユーザーあたりのエンティティの平均数（営業員あたりの取引先が 100 など）を見積もり、全体のユーザーベースでのエンティティあたりのレコード総数の概算を出します。
- 特定データベースの標準のサイズ変更手順および『Siebel Data Model Reference』を使用して、エンティティあたりの平均レコードサイズを計算し、レコード総数に掛けます。これらのエンティティは、複数の物理テーブルにわたっているのが一般的ですが、そのすべてを行サイズの計算に含める必要があります。これで、最大エンティティの概算データサイズがわかります。
- 他の Siebel アプリケーションデータのストレージ用スペースを追加する必要があります。ここで追加する容量の大まかな目安は、これらの主要なエンティティに必要な容量の半分です。
- 一般的に、インデックスはデータとほぼ同じ容量を必要とします。
- 全体のサイズ計算では、誤差を見込む必要があります。
- 増大率を全体のサイズ計算に必ず含めてください。

## データベースレイアウトのガイドライン（論理および物理）

Siebel Business Applications の全体的なパフォーマンスは、データベースサーバーの入力／出力パフォーマンス (I/O) に大きく依存します。最適な I/O パフォーマンスを実現するには、データベースのテーブルとインデックスを、I/O の負荷が均等に分散されるように、利用可能な複数のディスクデバイスにわたって配置することが重要です。

データベースオブジェクトの分散方法は、RDBMS ごとにさまざま、ストレージスペースがどのように割り当てられるかによって決まります。ほとんどのデータベースでは、特定のディスクに特定のオブジェクトを作成できます。次のリストに、これらのオブジェクトといくつかのガイドラインを示します。

独立したディスクの冗長アレイまたは RAID は、大量の I/O スループットおよび容量を提供できます。これは、オペレーティングシステムおよび RDBMS からは、単一の大容量ディスク（または管理の点で必要であれば複数ディスク）のように見えます。RAID を使用すると、高いパフォーマンスが保証されるとともに、物理ディスクの上に抽象層を提供することによって、データベースのレイアウトプロセスを大幅に簡素化できます。実装した RDBMS や選択したディスクの配置を問わず、次のタイプのデータベースオブジェクトを適切に分散していることを確認します。

- データベースログファイルまたはアーカイブファイル。
- データベースによって使用される一時ワークスペース。
- テーブルおよびインデックス。ほとんどの実装では、次のリストに含まれているテーブルとそれに対応するインデックスは、使用頻度が高くなる傾向があるため、別々のデバイスに置く必要があります。一般的には、次に示すインデックスは、それが作成されるテーブルとは異なる物理デバイスに置きます。
 

■ S_ACCNT_POSTN	■ S_PARTY_REL
■ S_OPTY	■ S_PARTY
■ S_ADDR_ORG	■ S_SRV_REQ
■ S_OPTY_POSTN	■ S_EVT_ACT
■ S_CONTACT	■ S_OPTY
■ S_POSTN_CONON	■ S_ORG_EXT
■ S_DOCK_TXN_LOG	

**備考：** EIM を頻繁に使用する計画の場合は、主要な EIM テーブル（固有のビジネス要件に依存）およびそれに対応するインデックスを、Siebel ベーステーブルおよびインデックスとは異なるデバイスに置きます。これは、EIM の処理中に、これらのすべてが同時にアクセスされるためです。

## EIM の使用計画

ここでは、Siebel の実装全体の規模にかかわらず、EIM を効果的に効率よく実装するためのいくつかの一般的なガイドラインについて説明します。EIM を使用できるようにするだけでなく、Siebel の実装を成功に導くためにも、EIM を実装するために戦略的なアプローチを取ることが非常に重要です。

詳しくは、次のトピックやサブトピックを参照してください。

- 131 ページの「チームの定義」
- 132 ページの「Siebel アプリケーションへのデータのマッピング」
- 133 ページの「EIM プロセスのテスト」

## チームの定義

顧客の経験に基づいて、組織に必要な EIM プロセスを管理および維持するためのチームを割り当てることをお勧めします。次のスキルを持つメンバーを含めることを考慮してください。

- 小規模から中規模の Siebel アプリケーションの実装を行う場合
  - 組織で使用する RDBMS だけではなく、Siebel データモデルについても十分な知識を持っているデータベース管理者。この担当者は、EIM テーブルにロードする実際のデータを決定し、データベースの物理レイアウトにより最適なパフォーマンスが得られるようにします。また、データを Siebel ベーステーブルにマッピングするタスクも実行します。このタスクの実行については、『Siebel Enterprise Integration Manager Administration Guide』を参照してください。
  - 組織で使用するシステムに関する経験が豊富なシステム管理者。この担当者は、組織に固有のスクリプトを開発して EIM テーブルへのデータのロードを自動化し、データを Siebel ベーステーブルで処理するために EIM を実行します。

**備考：** 組織には両方のスキルを持つ人が存在する場合があるため、これらのタスクを 1 人の担当者に割り当てることもできます。その場合は、その人をバックアップする別の担当者を決めて、主担当者が作業できないときはこのバックアップ担当者が Siebel の実装作業に必要なタスクを実行できるようにします。

- 大規模な Siebel の実装を行う場合
  - 組織で使用する RDBMS だけではなく、Siebel データモデルについても十分な知識を持っているデータベース管理者。この担当者は、EIM テーブルにロードする実際のデータを決定し、データベースの物理レイアウトにより最適なパフォーマンスが得られるようにします。また、このチームメンバーは、データを Siebel ベーステーブルにマッピングする重要なタスクも実行します。このタスクの実行については、『Siebel Enterprise Integration Manager Administration Guide』を参照してください。
  - 組織で使用するシステム（データベースサーバーとアプリケーションサーバーの両方）に関する経験が豊富なシステム管理者。この担当者は、組織に固有のスクリプトを開発して EIM テーブルへのデータのロードを自動化し、データを Siebel ベーステーブルで処理するために EIM を実行します。
  - Siebel データモデルと、Siebel の実装における Siebel データモデルの意図された使用方法を十分に理解しているビジネスアナリスト。このメンバーは、EIM チームのビジネスメンバーとテクニカルメンバーの連絡担当者となります。

## Siebel アプリケーションへのデータのマッピング

EIM では、EIM テーブルマッピングを使用して、EIM テーブルのカラムを Siebel ベーステーブルにマッピングします。Siebel の定義済み EIM マッピングは固定されており、再マップできません。

**備考：** EIM では、EIM テーブルマッピングのみを使用して、テーブルの関係が決定されます。EIM では、Siebel レポジトリの設定ロジックを使用して、テーブルの関係が決定されることはありません。

Siebel Tools を使用して、次の情報を表示できます。

- Siebel ベーステーブルへの EIM テーブルのマッピング
- Siebel ベーステーブルカラムへのカラムのマッピング
- EIM テーブルへの Siebel ベーステーブルのマッピング

いくつかのベーステーブルは、対応する EIM テーブルにマッピングできないことがあります。このような場合は、Siebel Visual Basic (VB) を使用してデータをこれらのベーステーブルにロードし、マッピングが不足していることを Siebel 技術サービスに連絡します。Siebel VB の使用方法については、『Siebel VB Language Reference』を参照してください。

ライセンス供与された Database Extensibility を所有しており、エクステンションを作成した場合は、[カラムマッピング] 画面を使用して新しいフィールドにマッピングを指定できます。Database Extensibility と EIM では、拡張テーブルのカラムと EIM テーブルのカラムが同じベーステーブルを共有している場合にのみ、これらのカラム間のマッピングがサポートされます。EIM テーブルエクステンションをベーステーブルエクステンションにマッピングするには、拡張フィールドが参照するベーステーブルのカラムを指定する必要があります。Database Extensibility については、『Configuring Siebel Business Applications』を参照してください。

### Siebel アプリケーションにデータをマッピングするには

- 1 Siebel を実装するために入力する必要のある Siebel ベーステーブルのカラムと、これらのベーステーブルにロードされる外部データを決定します。
- 2 ソースからターゲットへのインポートに使用される EIM テーブルとカラムを決定します。
- 3 この外部データを分析して、格納する必要のある属性と、このデータの他のエンティティに対する関係を決定します。

これを容易にするため、Siebel エキスパートサービスに EIM データのマッピングとデザインの確認を依頼できます。これにより、EIM マッピングが正しいこと、および意図した目標を達成できることを確認できます。

## EIM プロセスのテスト

すべての EIM プロセスを完全にテストすることは、見落とされがちです。テストは、データをマッピングしてからデフォルトの EIM 設定ファイルを使用して EIM プロセスを実行するだけではありません。完全にテストするには、類似したデータを持つ同じ EIM ジョブを大量に実行する必要があります。これにより、見落としていた部分を知るだけでなく、EIM バッチに最適なサイズ設定ができるようになり、本番環境で発生する可能性のあるシナリオを明確にすることができます。

EIM を使用する前に、データベース管理者は、EIM で処理するデータを EIM テーブルに入力する必要があります。そうしておけば、EIM を起動してこのデータを処理し、EIM でテーブルを介して処理を複数回実行して、指定した処理を完了させることができます。

EIM は、実行する EIM プロセス（インポート、マージ、削除、またはエクスポート）と、適切なパラメータを指定する特別な設定ファイルを読み取ります。EIM の設定ファイル（デフォルトファイルは default.ifb）は、拡張子が .IFB の ASCII テキストファイルで、Siebel Server ディレクトリの admin サブディレクトリにあります。EIM プロセスを実行する前に、EIM の設定ファイルの内容を編集して、EIM で実行するプロセスを定義する必要があります。

EIM ログファイルには、Error（エラー）フラグ、SQL フラグ、および Trace（追跡）フラグの 3 つのフラグの値に応じて、複数レベルの情報を含めることができます。これらのフラグについては、『Siebel Enterprise Integration Manager Administration Guide』を参照してください。次のリストに、いくつかの推奨設定を示します。

- 最初に、Error、SQL、Trace の各フラグをそれぞれ 1 に設定することをお勧めします。このように設定することで、エラーと使用していない外部キーが表示されます。Trace フラグを 1 に設定すると、EIM は必要に応じて Siebel データベーステーブルの主要な子関係を更新し、オプションのさまざまな SQL 文を実行した後に、経過時間の概要が（各バッチ後に）出力されます。
- Error フラグを 1、SQL フラグを 8、および Trace フラグを 3 に設定します。このように設定すると、各文の所要時間の情報を含む SQL 文を含むログファイルが生成されます。これは、SQL のパフォーマンスを最適化するのに役立ちます。
- Error フラグを 0、SQL フラグを 0、および Trace フラグを 1 に設定します。このように設定すると、EIM の各ステップの所要時間を示すログファイルが生成されます。このログファイルは、最適なバッチサイズを確認したり、特定ステップでのパフォーマンス低下を監視するのに便利です。

## EIM を最適化するときの一般的なガイドライン

EIM パフォーマンスを向上させるため、次のガイドラインを推奨します。

- 関連するテーブルのすべてのインデックスがあることを確認します。ただし、大量のロードを行う場合は、ターゲットテーブルの多くのインデックスを削除して、プロセスが完了した後にこれらのインデックスを再構築して、処理速度を向上させる必要があることに注意してください。
- ONLY BASE TABLES/COLUMNS 設定パラメータを使用して、処理するテーブルとカラムを制限し、EIM の処理を最小限に抑えます。
- EIM の実行中には、Docking: Transaction Logging システム属性を無効にすることを考慮してください。トランザクションのログをオフにするとパフォーマンスが向上します。ただし、この利点を利用する場合、モバイルユーザーは後で再抽出する必要があるということも合わせて考慮する必要があります。
- バッチサイズを変更して、特定ビジネスコンポーネントの最適なバッチサイズを確認することは、一般的にパフォーマンスの問題の解決に役立ちます。バッチサイズは、データ量および実行する EIM プロセスのタイプにより異なります。

**備考：** 処理可能な行数の制限は、データベースサーバーの機能に直接関係しますが、100,000 行以上のバッチを実行することはできるだけ避けてください。

- DELETE EXACT パラメータを使用する EIM の削除プロセスでは、20,000 行以下のバッチサイズを使用してください。
- バッチの範囲 (BATCH = x-y) を使用してください。これにより、小さなバッチサイズで実行して各バッチの起動時のオーバーヘッドを回避することができます。EIM プロセスで実行できるバッチの最大数は、1,000 です。
- EIM テーブルの定期的な保守を実行します。EIM テーブルで頻繁に挿入または削除オペレーションを行うと、断片化が発生する可能性があります。EIM テーブルの断片化の検出および修正については、データベース管理者にお問い合わせください。
- プロセスが完了したら、EIM テーブルからバッチを削除します。EIM テーブルにバッチを残しておくと、スペースが無駄になり、パフォーマンスに悪影響を及ぼすことがあります。
- 独立した EIM ジョブを並列して実行します。
- .IFB ファイルの USING SYNONYMS パラメータを「FALSE」に設定して、取引先略称を確認する必要がないことを指定します。
- 他の戦略が成功しない場合は、SQLPROFILE パラメータを使用して実行速度の遅いステップとクエリーを特定します。詳しくは、[142 ページの「SQLPROFILE パラメータの使用」](#)を参照してください。

## EIM プロセスを実装する際の推奨手順

EIM プロセスを実装する際は、次の手順を推奨します。

- 1 ビジネス要件を満たせるように、.IFB ファイルをカスタマイズしてテストします。
- 2 .IFB パラメータを調整します。
- 3 EIM プロセスを分離します。
- 4 データベースパラメータを設定して、ハードウェア、設定、および断片化をなくすか最小限に抑えることを含む、基本的な要件が満たされていることを確認します。

EIM プロセスの最適化を開始する前に、最適化の結果に影響する可能性のあるネットワークの問題やサーバーパフォーマンスの問題がないことを確認してください。Siebel エキスパートサービスでは、少なくとも 100 MB のネットワークセグメントおよびネットワークインターフェイスカード (NIC) を使用して Siebel サーバーおよび Siebel データベースサーバーに接続することをお勧めします。また、Siebel エキスパートサービスでは、スループットを最大化するために、ハブではなく、ネットワークスイッチまたは類似したテクノロジーを使用することをお勧めします。

### .IFB ファイルの最適化

ビジネス要件を満たすための .IFB ファイルのコーディングとテストが終了したら、次に .IFB ファイルを最適化します。.IFB ファイルの各セクションで選択したパラメータは、各 EIM タスクの焦点を決定します。次に、.IFB ファイルの各セクションの推奨事項について説明します。

- **ONLY BASE TABLES** または **IGNORE BASE TABLES** : これらのパラメータは、EIM プロセスのベーステーブルを指定し、選択したベーステーブルを制限します。1 つの EIM テーブル (インターフェイステーブルとも呼ばれる) は、複数のユーザーまたは複数のベーステーブルにマッピングされます。たとえば、テーブル EIM\_ACCOUNT は、S\_PARTY、S\_ORG\_EXT、S\_ADDR\_ORG、およびその他のテーブルにマッピングされます。デフォルトの設定では、EIM テーブルごとにすべてのベーステーブルが処理されます。

**備考:** Siebel エキスパートサービスでは、.IFB ファイルのセクションごとに必ずこれらのパラメータを含めること、および特定の EIM タスクに関係するテーブルとカラムだけをリストすることを強くお勧めします。

- **ONLY BASE COLUMNS** または **IGNORE BASE COLUMNS** : これらのパラメータは、EIM プロセスのベースカラムを指定し、選択したベースカラムを制限します。デフォルトでは、ベーステーブルごとにすべてのベースカラムが処理されます。ベーステーブルのカラムをすべて使用する可能性は低いので、これらのパラメータにより EIM がテーブルの必要なカラムだけを処理するようにします。外部キー (FK) として定義されているこれらのカラムを除外し、Siebel の設定で使用しない場合は、パフォーマンスがさらに向上します。これは、EIM では中間処理 (SQL 文を使用) を実行して、これらの FK の値を解決する必要がなくなるためです。EIM Task パラメータの Error フラグを 1 に設定して、EIM で解決されない FK を確認します (このパラメータで、解決されない FK を除外している場合もあります)。

**備考:** マージプロセスまたはエクスポートプロセスには、IGNORE BASE COLUMNS パラメータを使用しないでください。このパラメータは、インポートプロセスと削除プロセスにのみ使用してください。

### .IFB ファイルの最適化の確認

.IFB ファイルが最適化されているかどうかを確認する 1 つの方法は、EIM テーブルで処理されているレコードの状況を確認することです。レコードの状況を確認すると、不必要に処理されているテーブルまたはカラムがあるかどうかわかります。EIM テーブルのレコードの状況を確認するには、次のクエリーを使用できます。

```
select count(*), IF_ROW_STAT from EIM Table
where IF_ROW_BATCH_NUM = ?
group by IF_ROW_STAT;
```

多くの行の状況が PARTIALLY IMPORTED になっている場合は、不要なベーステーブルとベースカラムを除去して、さらにチューニングを行うことができます。たとえば、EIM\_ACCOUNT テーブルの 5000 件の取引先をインポート (IMPORT) する 2 つのテストを実行したとします。最初のテストではすべてのベーステーブルが含まれましたが、.IFB ファイルに次の行を含めることで、2 番目のテストでは 4 つの必要なテーブルのみをテストの対象としました。

```
ONLY BASE TABLES = S_ORG_EXT, S_ADDR_ORG, S_ACCNT_POSTN, S_ORG_TYPE
```

最初のインポートのテスト (プライマリの更新ステップを除く) には 89 分かかりましたが、2 番目のテスト (プライマリの更新ステップを除く) はわずか 2 分で完了しました。

### オペレーションごとの EIM プロセスの分離

可能な場合は、EIM バッチを、挿入のみのトランザクションと更新のみのトランザクションに分割します。たとえば、毎週行う処理の一部として、50,000 レコードをロードしていると仮定します。10,000 レコードは新しいデータで、40,000 レコードは既存データの更新です。

デフォルトで EIM は、ベーステーブルの追加するレコードや更新するレコードを判断できますが、これらのレコードを特定するために、追加処理を (SQL 文を使用して) 実行する必要があります。50,000 レコードをトランザクションのタイプごとに別のバッチ番号に分割できる場合は、この追加処理を避けることができます。

さらに、更新アクティビティの一部として処理されるカラムは、挿入アクティビティの一部として処理されるカラムより少なくなる場合があります (結果として、パフォーマンスがさらに向上します)。これを説明するため、次のセクションを使用して、前の例の .IFB をコーディングすることができます。

#### ■ 複数のトランザクションが混在している .IFB の場合

```
[Weekly Accounts]
```

```
TYPE = IMPORT
```

```
BATCH = 1-10
```

```
TABLE = EIM_ACCOUNT
```

```
ONLY BASE TABLES = S_ORG_EXT
```

```
IGNORE BASE COLUMNS = S_ORG_EXT.?
```

- 挿入トランザクションと更新トランザクションが分離されている .IFB の場合

[Weekly Accounts - New]

TYPE = IMPORT

BATCH = 1-2

TABLE = EIM\_ACCOUNT

ONLY BASE TABLES = S\_ORG\_EXT

IGNORE BASE COLUMNS = S\_ORG\_EXT.?

INSERT ROWS = TRUE

UPDATE ROWS = FALSE

[Weekly Accounts - Existing]

TYPE = IMPORT

BATCH = 3-10

TABLE = EIM\_ACCOUNT

ONLY BASE TABLES = S\_ORG\_EXT

ONLY BASE COLUMNS = S\_ORG\_EXT.NAME, S\_ORG\_EXT.LOC, S\_ORG\_EXT.?

INSERT ROWS = FALSE

UPDATE ROWS = TRUE

## EIM のパフォーマンスのトラブルシューティング

EIM のパフォーマンスをトラブルシューティングする前に、Siebel Server マシンまたはネットワークにパフォーマンスのボトルネックがないことを確認してください。

### EIM のための SQL の最適化

この処理中は、いくつかの類似したバッチを実行できるようにしておく必要があります。テストに十分なデータがない場合は、実行と実行の間にデータベースをバックアップして復元する必要があります。このようにすることで、同じバッチの処理を続行することができます。

まず、Error フラグを 1、SQL フラグを 8、Trace フラグを 3 にそれぞれ設定して、EIM ジョブを実行する必要があります。このように設定すると、SQL 文を含むログファイルが生成され、各文の所要時間が表示されます。時間のかかりすぎている SQL 文を特定します（1 つのバッチで 5,000 行が実行される場合、1 分以上かかっている文を探します）。これらの文が焦点を当てる文です。この時点で、十分な経験のあるデータベース管理者に相談してください。EIM の SQL を最適化するプロセスでは、次の操作を行います。

- データベースベンダーのユーティリティまたはサードパーティ製ユーティリティを使用して、実行時間の長い SQL 文を分析します。
- データのアクセスパスのレビュー結果に基づいて、SQL 文でインデックスが適切に使用されていることを確認します。インデックスがまったく使用されていないか、最も効果的なインデックスが選択されていない場合があります。この場合は、詳細な分析が必要になります。
- この分析結果に基づいて、体系的なアプローチを使用して、これらの実行時間の長い文を調整します。変更は一度に 1 つずつ行う必要があります。変更を行ったら、これを最初のベンチマークと比較して、変更の結果を測定します。たとえば、実行時間の長い文のパフォーマンスを向上させるために特定のインデックスを削除すると、他の SQL 文のパフォーマンスが低下する場合があります。

インデックスを削除するかどうかは、個々の実行時間の長い SQL 文ではなく、プロセス全体に対する影響に基づいて決定する必要があります。このような理由から、変更による影響の測定は、1 つの変更ごとに行うことが重要になります。

- 実行時間の長い SQL 文を繰り返し実行して最適化した後は、EIM テーブルで一度に処理されるレコード数を増やしたり、EIM タスクを並列実行したりするなど、他のチューニング方法に注目することができます。

## USE INDEX HINTS パラメータと USE ESSENTIAL INDEX HINTS パラメータの使用

.IFB ファイルパラメータの USE INDEX HINTS と USE ESSENTIAL INDEX HINTS について、両方の設定 (TRUE および FALSE) を使用してテストを実行します。USE INDEX HINTS のデフォルト値は FALSE です。USE ESSENTIAL INDEX HINTS のデフォルト値は TRUE です。

**備考:** 設定ファイルに複数のプロセスセクションがある場合は、そのプロセスセクションごとに USE INDEX HINTS を指定する必要があります。

これらのパラメータが FALSE に設定されていると、EIM では処理中にヒントが生成されません。値を TRUE に設定して、最適よりも少ないインデックスを使用するようにデータベースオプティマイザを指定するヒントが生成される場合は、FALSE に設定することでパフォーマンスの向上を実現できる場合があります。EIM 処理は TRUE と FALSE の両方の設定を使用してテストし、EIM ジョブごとにどちらの設定がよりよいパフォーマンスを実現できるかを確認する必要があります。

**備考:** USE INDEX HINTS パラメータは、Oracle データベースプラットフォームに対してのみ適用できます。USE ESSENTIAL INDEX HINTS パラメータは、Microsoft SQL Server および Oracle データベースプラットフォームに対してのみ適用できます。

これらの 2 つのパラメータは別のクエリーにも使用できるため、両方のパラメータを有効にして Oracle データベースプラットフォーム上のすべてのインデックスヒントを取得する必要があります。

詳しくは、次のトピックを参照してください。

- 139 ページの「例：USE INDEX HINTS パラメータと USE ESSENTIAL INDEX HINTS パラメータの使用」
- 141 ページの「USE INDEX HINTS と USE ESSENTIAL INDEX HINTS：データベースへのインデックスの引き渡しの EIM 基準」

### 例：USE INDEX HINTS パラメータと USE ESSENTIAL INDEX HINTS パラメータの使用

次の例は、インデックスヒントのある SQL 文とインデックスヒントのない SQL 文の実行結果を示しています。この例は、Microsoft SQL Server プラットフォームで実行されています。

SQL User Name	CPU	Reads	Writes	Duration	Connection ID	SPID
SADMIN	549625	<b>38844200</b>	141321	<b>626235</b>	516980	9

```

UPDATE dbo.S_ASSET5_FN_IF
SET T_APPLDCVRG__RID =
(SELECT MIN(BT.ROW_ID)
FROM dbo.S_APPLD_CVRG BT (INDEX = S_APPLD_CVRG_U2)
WHERE (BT.COVERAGE_CD = IT.CVRG_COVERAGE_CD AND

```

```
BT.TYPE = IT.CVRG_TYPE AND
BT.ASSET_ID = IT.T_APPLDCVRG_ASSETI AND
(BT.ASSET_CON_ID = IT.T_APPLDCVRG_ASSETC OR
(BT.ASSET_CON_ID IS NULL AND IT.T_APPLDCVRG_ASSETC IS NULL)) AND
(BT.INSITEM_ID = IT.T_APPLDCVRG_INSITE OR
(BT.INSITEM_ID IS NULL AND IT.T_APPLDCVRG_INSITE IS NULL))))
FROM dbo.S_ASSET5_FN_IF IT
WHERE (CVRG_COVERAGE_CD IS NOT NULL AND
CVRG_TYPE IS NOT NULL AND
T_APPLDCVRG_ASSETI IS NOT NULL AND
IF_ROW_BATCH_NUM = 10710001 AND
IF_ROW_STAT_NUM = 0 AND
T_APPLDCVRG__STA = 0)
SET STATISTICS PROFILE ON
GO
SET STATISTICS IO ON
GO
select
(SELECT MIN(BT.ROW_ID)
FROM dbo.S_APPLD_CVRG BT (INDEX = S_APPLD_CVRG_U2)
WHERE (BT.COVERAGE_CD = IT.CVRG_COVERAGE_CD AND
BT.TYPE = IT.CVRG_TYPE AND
BT.ASSET_ID = IT.T_APPLDCVRG_ASSETI AND
(BT.ASSET_CON_ID = IT.T_APPLDCVRG_ASSETC OR
(BT.ASSET_CON_ID IS NULL AND IT.T_APPLDCVRG_ASSETC IS NULL)) AND
(BT.INSITEM_ID = IT.T_APPLDCVRG_INSITE OR
(BT.INSITEM_ID IS NULL AND IT.T_APPLDCVRG_INSITE IS NULL))))
FROM dbo.S_ASSET5_FN_IF IT
```

```
WHERE (CVRG_COVERAGE_CD IS NOT NULL AND
CVRG_TYPE IS NOT NULL AND
T_APPLDCVRG_ASSETI IS NOT NULL AND
IF_ROW_BATCH_NUM = 10710001 AND
IF_ROW_STAT_NUM = 0 AND
T_APPLDCVRG__STA = 0)
```

#### インデックスヒントありの場合

Table 'S\_APPLD\_CVRG'.Scan count 1, **logical reads 394774**, physical reads 0, read-ahead reads 280810.

Table 'S\_ASSET5\_FN\_IF'.Scan count 1, logical reads 366, physical reads 0, read-ahead reads 0.

#### インデックスヒントなしの場合

Table 'S\_APPLD\_CVRG'.Scan count 1268, **logical reads 10203**, physical reads 697, read-ahead reads 0.

Table 'S\_ASSET5\_FN\_IF'.Scan count 1, logical reads 366, physical reads 0, read-ahead reads 0.

## USE INDEX HINTS と USE ESSENTIAL INDEX HINTS : データベースへのインデックスの引き渡しの EIM 基準

ここでは、USE INDEX HINTS パラメータと USE ESSENTIAL INDEX HINTS パラメータを使用するときに、EIM でデータベースに渡すヒント句に含めるインデックスが決定される方法について説明します。インデックスヒントとしてデータベースに渡すインデックスを決定する際に、EIM では次の手順が実行されます。

- 1 クエリーを生成する前に、EIM によりインデックスが必要と判断されたカラムのリストが作成されます。
  - 2 次に、レポジトリのすべてのインデックスが確認され、カラムが最もよく一致するインデックスが検索されます。
- EIM では、インデックスを選択する際に、次の選択基準が使用されます。

- 一意のインデックスは、一意ではないインデックスよりも優先されます。
- 必須カラムは、任意カラムよりも優先されます。

新しいインデックスが作成され、これがレポジトリ内で宣言されている場合は、EIM でそのインデックスを選択し、ヒントでデータベースに渡す可能性があります。

**備考：** Oracle データベースでは、EIM は、Oracle のルールベースオプティマイザー（RBO）モードを使用します。INDEX ヒントを指定すると、RBO モードでは、ヒントで指定されたインデックスを使用する必要があることが認識されます。この場合、RBO モードではテーブル全体のスキャンは行われず、他のインデックスが使用されることもありません。Oracle データベースのオプティマイザーモードについては、『Siebel データベースアップグレードガイド』を参照してください。

## SQLPROFILE パラメータの使用

このパラメータを含めることで、最も時間のかかる SQL 文を特定するタスクが大幅に簡素化されます。次の文を .IFB ファイルのヘッダーセクションに挿入することで、最も時間のかかる SQL 文が、次のファイルに出力されます。

```
SQLPROFILE = c:\temp\%eimsql.sql
```

次に、eimsql.sql ファイルの例を示します。

ファイルの先頭 - 最も時間のかかるクエリーの一覧

```
EIM: Integration Manager v6.0.1.2 [2943] ENU SQL profile dump (pid 430).
*****
Top 34 SQL statements (of 170) by total time:
Batch Step Pass Total Rows Per Row What
-----
106 10 401 1334.48 5000 0.27 update implicit primaries to child
106 9 114 242.56 5000 0.05 copy
```

(クエリーの一覧が続きます)

ステップおよびパスごとの統計

```
*****
Statements per step by total time:
Step Stmt Total Min Max Avg %
-----
10 15 2627.27 0.00 1334.48 175.15 83.73
9 11 329.52 0.00 242.56 29.96 10.50
```

(統計の一覧が続きます)

SQL 文:

```
*****
batch 106, step 10, pass 401:"update implicit primaries to child":
(total time 22:14m (1334s), 5000 rows affected, time/row 0.27s)
UPDATE siebel.S_CONTACT BT
```

```

SET PR_BL_PER_ADDR_ID =
(SELECT VALUE(MIN(ROW_ID), 'No Match Row Id')
FROM siebel.S_ADDR_PER CT
WHERE (CT.PER_ID = BT.ROW_ID)),
LAST_UPD = ?,
LAST_UPD_BY = ?,
MODIFICATION_NUM = MODIFICATION_NUM + 1
WHERE (ROW_ID IN (
SELECT T_ADDR_PER_PER_ID C1
FROM siebel.EIM_CONTACT
WHERE(
T_ADDR_PER_PER_ID IS NOT NULL AND
IF_ROW_BATCH_NUM = 106 AND
T_ADDR_PER__STA = 0 AND
T_ADDR_PER__EXS = 'N' AND
T_ADDR_PER__UNQ = 'Y' AND
T_ADDR_PER__RID IS NOT NULL)
GROUP BY T_ADDR_PER_PER_ID)
AND
(PR_BL_PER_ADDR_ID IS NULL OR PR_BL_PER_ADDR_ID = 'No Match Row Id'))
*****

```

(SQL 文の一覧が続きます)

## EIM テーブルの追加インデックス

データのアクセスパスを確認すると、実行時間の長い SQL のパフォーマンスを向上させるために追加インデックスが必要かどうかを決定するのに役立ちます。特に、テーブルスキャンや大きなインデックス範囲スキャンを探してください。次の例では、ネストされた選択 (select) の内部ループを評価した後、すべての T2 カラムにインデックスを追加することが推奨されました。

内部ループ：

```
(SELECT MIN(ROW_ID)
FROM siebel.EIM_ACCOUNT T2
WHERE (T2.T_ADDR_ORG__EXS = 'Y' AND
T2.T_ADDR_ORG__RID = T1.T_ADDR_ORG__RID AND
T2.IF_ROW_BATCH_NUM = 105 AND
T2.IF_ROW_STAT_NUM = 0 AND
T2.T_ADDR_ORG__STA = 0))
```

インデックスの最後に ROW\_ID を指定し、WHERE 句で使用される T2 カラムで構成されるインデックスが作成されました。これにより、インデックス専用アクセスを行うため、データベースオプティマイザーはこのインデックスを選択します。このクエリーでは最小値 (ROW\_ID) が必要になるため、インデックスの最初の有効なページにも最小値が含まれるようになります。

**備考：** ROW\_ID 列を先頭インデックスカラムとして指定するのも、よい方法です。ROW\_ID は一意であるため、インデックスはより選択的になると考えられます。

## S\_ORG\_EXT のパフォーマンスを向上させるためのインデックスの追加

S\_ORG\_EXT テーブルには多くのカラムにインデックスがありますが、すべてのカラムにはありません。S\_ORG\_EXT に多数のレコード (数百万件の取引先) がある場合は、次の 1 つまたは複数のカラムにインデックスを追加すると、削除およびマージ処理におけるパフォーマンスが向上されることがあります。

- PR\_BL\_OU\_ID
- PR\_PAY\_OU\_ID
- PR\_PRTNR\_TYPE\_ID
- PR\_SHIP\_OU\_ID

インデックスを追加する前に、この件について、適切なサポート担当者にお問い合わせください。

## EIM テーブルに関する適切な統計の作成

.IFB ファイルの UPDATE STATISTICS パラメータは、DB2 データベースプラットフォームでのみ使用できます。このパラメータによって、EIM テーブルの統計を動的に更新するかどうかは制御されます。デフォルトの設定は TRUE です。このパラメータを使用して、EIM テーブルに関する統計セットを作成できます。この統計セットは、保存して後続の実行に再適用できます。最適な統計セットを決定したら、.IFB ファイルの UPDATE STATISTICS パラメータを無効にして (UPDATE STATISTICS = FALSE)、EIM の実行時間を短縮することができます。

最適な統計セットを決定するには、いくつかのテストバッチを実行したり、さまざまなオプションを選択して RUNSTATS コマンドを実行したりして、最適な結果となる設定を確認する必要があります。

各テストの前後に、mimic mode で db2look ユーティリティを実行して、データベースのシステムカタログの統計を保存する必要があります。たとえば、SIEBELDB データベースの EIM\_CONTACT1 を使用して EIM の実行をテストする場合は、次のコマンドにより、EIM\_CONTACT1\_mim.sql ファイルに UPDATE STATISTICS コマンドが生成されます。

```
db2look -m -a -d SIEBELDB -t EIM_CONTACT1 -o  
EIM_CONTACT1_mim.sql
```

EIM\_CONTACT1\_mim.sql ファイルには、保存した統計でデータベースのシステムカタログテーブルを更新する SQL UPDATE 文が含まれています。

「DB2 バージョン 8 のオプション」に説明のある RUNSTATS コマンドを挿入した後に、テスト用 EIM バッチを実行できます。最もよく機能する統計セットを見つけたら、この特定の mim.sql ファイルをデータベースに適用できます。

**備考：** 次の実行を行う前に、必ず db2look で統計を保存するようにしてください。

### DB2 バージョン 8 のオプション

DB2 V8 のコマンドの構文には、次のように、多くのオプションが用意されています。

- shrlevel change
- allow write access
- allow read access

allow read access および shrlevel change 句を使用すると、同時実行能力を最大限に高めることができます。

## 初期実行でのインデックスの削除

通常、EIM の初期ロードは、データベースを集中的に使用するプロセスとなります。ベーステーブルに挿入される各行では、影響を受けるすべてのインデックスのデータページとインデックスページを変更する必要があります。ただし、これらのほとんどのインデックスは、EIM の実行中に使用されることはありません。インデックスの管理は、多くのデータベース管理者にとって非常に時間のかかる処理なので、できるだけ避ける必要があります。

このため、ここでの目標は、EIM に不要で EIM の実行中に削除できるインデックスを特定することです。これらのインデックスは、それぞれのデータベースプラットフォームで使用できる並列実行方式を使用して、後でバッチモードで作成できます。この方法を使用すると、大幅に時間を節約することができます。

**備考：** 通常の操作では、並列実行方式の使用はお勧めできません。

- **ターゲットテーブルのインデックス処理方法：**ターゲットベーステーブル(S\_ORG\_EXT など)の場合、Primary Index (Px、たとえば P1 など) や、Unique Index (Ux、たとえば U1 など) だけを使用して、EIM のインポート中に残りのインデックスを削除する必要があります。過去の経験から、サンプルの EIM 実行の分析を詳細に行った結果、Fx インデックスと Mx インデックスを削除できることが示されています。
- **ターゲット以外のテーブルのインデックス処理方法：**子テーブル (S\_ADDR\_ORG など) の場合は、使用する必要があるのは Primary Index (Px)、Unique Index (Ux)、および Foreign Key Index (親テーブルのプライマリ外部キーを設定するために必要) のみです。過去の経験から、サンプルの EIM 実行の分析を詳細に行った結果、Fx インデックスと Mx インデックスを削除できることが示されています。

**備考：** インデックスを削除 (または追加) して予想した結果になることを確認するには、必ずテストを実行する必要があります。

## バッチサイズの制御

実行時間の長い SQL 文を調整したら、さらにテストを行って、処理する各エンティティに対する最適なバッチサイズを決定できます。適切なバッチサイズは、使用できるバッファキャッシュの量により影響を受けるため、異なります。最適なバッチの範囲は、500 ~ 15,000 行の範囲であることが確認されています。異なるバッチサイズで何回かテストを行って、1 秒当たりの EIM トランザクションで最適なレートが得られるバッチサイズを決定します。EIM の実行中に Trace フラグを 1 に設定すると、各ステップの所要時間や EIM プロセスで処理される行数を確認できるため、このタスクに役立ちます。

**備考：** また、EIM の並列実行にパフォーマンスの低下が見られるときは、このスループットレートを監視する必要があります。

## 単一バッチに推奨される行数

初期ロードの場合、大きなバッチでは 30,000 行使用できます。継続的なロードの場合、大きなバッチで 20,000 行使用できます。大きなバッチは、100,000 行以下にする必要があります。

また、Microsoft SQL Server および Oracle の環境では、EIM テーブルのレコード数を、処理されているレコード数に制限する必要があります。たとえば、実装の最適なバッチサイズがバッチ当たり 19,000 行であり、8 つの EIM 並列プロセスを実行する場合は、EIM テーブルに 152,000 行が必要になります。1 つの EIM テーブルに 250,000 を超える行を含めるとパフォーマンスが低下するため、いかなる場合でも、250,000 行を超えないようにしてください。

前述の例の制限は、DB2 環境には適用されません。インデックスを使用して EIM テーブルにアクセスする限り、EIM テーブルの行数は、DB2 環境では問題になりません。

**備考：** 単一のバッチでロードできる行数は、テーブルがロードされる物理マシンの設定により異なります。リソースに対する要求を軽減しパフォーマンスを向上させるため、一般的に、バッチサイズをさまざまに変更して、処理する各エンティティの最適なサイズを決定する必要があります。場合によっては、バッチサイズを小さくすることで、パフォーマンスが向上します。ただし、S\_ASSET などの単純なテーブルの場合は、S\_CONTACT などの、より複雑なテーブルよりもバッチサイズを大きくした方が、ロードのパフォーマンスが向上します。

## EIM テーブルのレコード数の制御

EIM 処理中に許容可能なスループットレートを維持しながら、EIM テーブルに一度に配置できるレコード数を決定する必要があります。EIM テーブルのレコード数を増やすことで見られる影響の 1 つは、EIM ジョブのパフォーマンス低下です。これは多くの場合、オブジェクトの断片化、テーブル全体のスキャン、または大きなインデックス範囲のスキャンが原因です。

**備考：** DB2 環境では、テーブルスキャンや非一致のインデックススキャンは簡単に修正できるため、EIM テーブルのサイズは、パフォーマンスに影響を与える重要な要素ではありません。そのため、EIM テーブルに多数のレコードがあっても、DB2 環境ではパフォーマンスが低下することはほとんどありません。

オブジェクトの断片化を解決し、実行時間の長い SQL 文を調整した後は、EIM の処理中に EIM テーブルに配置できるレコード数を増やすことができる可能性があります。数百万件のレコードをロードする場合は、この方法により、新しいデータセットを使用して EIM テーブルをステージングする必要がある回数が減るため、大幅に時間を節約することができます。

大きなデータ（数百万件のレコード）のロードを実行する場合は、最初に EIM テーブルで少ないレコードを使用してロードテストを行うことをお勧めします。たとえば、実行時間の長い SQL を特定して調整する場合は、約 50,000 レコードからテストを開始します。調整作業が終わったら、徐々にレコード数を増やしながら他のテストを実行します。たとえば、ロードするのに最適なレコード数を決定できるまで、レコード数を 100,000、200,000 というように徐々に増やしていくことができます。

## USING SYNONYMS パラメータの使用

USING SYNONYMS パラメータは、インポート処理中に取引先略称のクエリーを制御します。また、このパラメータは、S\_ORG\_SYN テーブルにも関連付けられています。FALSE に設定すると、略称を検索するクエリーは使用されないため、このパラメータの処理時間が短縮されます。デフォルトの設定は TRUE です。このパラメータは、取引先略称が不要なときのみ FALSE に設定してください。

## NUM\_IFTABLE\_LOAD\_CUTOFF 拡張パラメータの使用

この拡張パラメータを正の値に設定すると、EIM がレポジトリ情報をロードする時間が短縮されます。これは、このパラメータを正の値に設定すると、必要な EIM テーブルの情報だけがロードされるためです。このパラメータについて詳しくは、『Siebel Enterprise Integration Manager Administration Guide』を参照してください。

**備考：** このパラメータは、特にマージ処理で重要になります。また、その他のタイプの処理でも使用できます。

次に、サーバーコマンドラインモードから Microsoft Windows で実行する場合の、このパラメータの使用例を示します。

```
run task for comp eim server siebserver with config=account2.ifb,  
ExtendedParams="NUM_IFTABLE_LOAD_CUTOFF=1", traceflags=1
```

## Docking: Transaction Logging の無効化

通常、Docking: Transaction Logging 設定は、最初にデータをロードするときのみ使用されます。システム属性値 Docking: Transaction Logging は、Siebel アプリケーションの [システム設定] ビューから設定されます。この設定は、Siebel モバイル Web クライアントにデータをルーティングする目的で、Siebel アプリケーションでトランザクションのログを記録するかどうかを指定します。

このパラメータのデフォルト値は TRUE です。Siebel モバイル Web クライアントがない場合は、このシステム設定を FALSE にできます。Siebel モバイル Web クライアントがある場合は、Siebel モバイル Web クライアントにトランザクションをルーティングするため、このパラメータを TRUE に設定する必要があります。ただし、最初にデータをロードするときに、このパラメータを FALSE に設定して、Siebel ドッキングテーブルに対するトランザクションアクティビティを減らすことができます。初期ロードが完了したら、このパラメータを TRUE に戻します。

**備考：** 増分データロードでは、モバイルクライアントがある場合は、Docking: Transaction Logging の設定は TRUE のままにしておく必要があります。増分データロードでこの設定を変更する場合は、すべてのモバイルクライアントを再抽出する必要があります。

## トリガーの無効化

[サーバー管理] 画面からデータベーストリガーを削除してこれを無効にすると、スループットレートを向上させることができます。これを行うには、REMOVE パラメータと EXEC パラメータの両方を TRUE に設定して、Generate Triggers サーバータスクを実行します。ワークフローポリシーや割当マネージャなどのコンポーネントは、新しいデータおよび更新されたデータには機能しないことに注意してください。また、EIM のロードが完了したら、トリガーを再適用してください。

## EIM タスクの並列実行

EIM タスクの並列実行は、EIM のスループットレートを向上させるために適用する最後の手段です。つまり、この方法は、実行時間の長いすべての SQL 文を調整し、最適なバッチサイズを決定して、EIM テーブルで一度に処理する最適なレコード数を決定し、データベースを適切にチューニングし終えるまで、試さないでください。タスクを並列実行する前に、Maximum Tasks パラメータの値を確認してください。このパラメータは、サービスに対して一度に実行できるタスクの最大数を指定します。このパラメータについては、『Siebel システム管理ガイド』を参照してください。

**備考：** EIM タスクを IBM DB2 プラットフォームで並列実行する場合は、.IFB ファイルで UPDATE STATISTICS を FALSE に設定する必要があります。この設定を行わないと、EIM タスクと RUNSTATS の実行完了に長い時間がかかることがあります。また、EIM タスクを並列実行する場合は、.IFB ファイルで UPDATE STATISTICS が TRUE に設定されていると、デッドロックとタイムアウトが発生します。

## EIM を最適化するときのデータベースのガイドライン

次のトピックでは、Siebel アプリケーションでサポートされているデータベースプラットフォーム（Microsoft SQL Server、Oracle、および IBM DB2 UDB）のチューニングのヒントについて説明します。詳しくは、次のトピックを参照してください。

- 150 ページの「Microsoft SQL Server」
- 153 ページの「Oracle データベース」
- 155 ページの「IBM DB2 UDB」
- 158 ページの「IBM DB2 UDB for z/OS」
- 158 ページの「EIM のための IBM DB2 ローディングプロセス」
- 159 ページの「IBM DB2 ローディングプロセスに関する一般的な推奨事項」

### Microsoft SQL Server

次の項では、Microsoft SQL Server データベースプラットフォームでの、EIM のチューニングのヒントについて説明します。

#### テーブルの断片化の解消

テーブルとインデックスの断片化は、多くの挿入、更新、および削除操作を行ったテーブルで発生します。テーブルが変更されるため、ページがいっぱいになり、クラスタ化されたインデックスにページが分割されます。ページが分割されると、新しいページにより連続していないディスク領域が使用され、パフォーマンスが低下します。これは、連続するページは順次入出力（I/O）の形式になっており、非順次 I/O よりも高速に処理されるためです。

EIM を実行する前に、テーブルのクラスタ化インデックスで DBCC DBREINDEX コマンドを実行して、断片化を解消しておくことが重要です。これは、特に EIM の処理中に使用されるインデックスに該当します。この処理では、FILL FACTOR のデータ量（FILLFACTOR オプションを使用して設定）を指定した各データページがバックされ、連続するデータページの情報が並び替えられます。また、SORTED\_DATA オプションを使用せずに、インデックスを削除して再作成することもできます。ただし、次の例に示すように、インデックスを削除して再作成するよりも高速に処理できるため、DBCC DBREINDEX コマンドを使用することをお勧めします。

```
DBCC SHOWCONTIG scanning '**S_GROUPIF' table...

Table: '**S_GROUPIF' (731969784); index ID:1, database ID: 7

TABLE level scan performed.

Pages Scanned.....: 739
Extents Scanned.....: 93
Extent Switches.....: 92
Avg.Pages per Extent.....: 7.9
```

```
Scan Density [Best Count:Actual Count].....: 100.00% [93:93]
Logical Scan Fragmentation .....: 0.00%
Extent Scan Fragmentation .....: 1.08%
Avg.Bytes Free per Page.....: 74.8
Avg.Page Density (full).....: 99.08%

DBCC execution completed. If DBCC printed error messages, contact the system
administrator.
```

インデックスページが過剰に分割されたためインデックスを再構築する必要があるかどうかを判断するには、DBCC SHOWCONTIG で表示される Scan Density の値を確認します。Scan Density の値は、100% かそれに近い必要があります。値が 100% よりも大幅に下回るときは、インデックスを再構築してください。

## EIM テーブルの削除

EIM テーブルからデータを削除するときは、TRUNCATE TABLE 文を使用します。この方法では、高速にテーブルのすべての行を削除し、ログには記録されません。DELETE は、物理的に一度に 1 行を削除し、削除した各行をトランザクションログに記録します。TRUNCATE TABLE は、データページ全体の割り当て解除だけをログに記録し、そのテーブルのデータとインデックスにより消費されているすべての領域を即時に開放します。すべてのインデックスの割り当てページも、開放されます。

## bcp を使用した EIM テーブルのデータの並列ロード

Microsoft SQL Server では、bcp ユーティリティまたは BULK INSERT 文を使用して、複数のクライアントから 1 つの EIM テーブルに並行して、データを一括コピーすることができます。次の条件に当てはまる場合は、bcp ユーティリティまたは BULK INSERT 文を使用してください。

- SQL Server が、複数のプロセッサを搭載したコンピュータ上で実行されている
- EIM テーブルに一括コピーするデータが、複数のデータファイルに分割できる

次の推奨事項により、データロード操作のパフォーマンスを向上させることができます。記載されている順に次のタスクを実行して、データを SQL Server に並列で一括コピーします。

- 1 sp\_dboption を使用して、データベースオプションの [truncate log on checkpoint (trunc. log on chkpt.)] を TRUE に設定します (\*)。
- 2 sp\_dboption を使用して、データベースオプションの [select into/bulkcopy] を TRUE に設定します。

ログを出力するよう設定した一括コピーでは、すべての行の挿入のログが記録されます。そのため、大きな一括コピー操作を行うと、多くのログレコードが生成されることがあります。これらのログレコードは、ログを出力するよう設定した一括コピー操作のロールフォワードとロールバックの両方に使用できます。

ログを出力しないよう設定した一括コピーでは、一括コピーされた行のある新しいページの割り当てだけがログに記録されます。これにより、必要なログの量が大幅に軽減され、一括コピー操作が高速化されます。ログを出力しない操作を行ったら、すぐにバックアップを作成して、トランザクションのログ記録を再開できるようにする必要があります。

- 3 テーブルにインデックスがないこと、またはテーブルにインデックスがある場合は、バルクコピーを開始する前に、インデックスが空になっていることを確認してください。
- 4 ターゲットテーブルを複製しようとしていないことを確認してください。
- 5 eOption を BCPHINTS に設定した `bcp_control` を使用して、TABLOCK ヒントが指定されていることを確認してください。  
**備考：** 順序付けされたデータと ORDER ヒントを使用しても、データのロード時には、クラスタ化インデックスは EIM テーブルに存在しないため、パフォーマンスには影響しません。
- 6 複数のクライアントから 1 つの EIM テーブルにデータを一括コピーした後は、DBCC DBREINDEX を使用して、テーブルのクラスタ化インデックスを再作成する必要があります。

## TempDB

：これは、さまざまなクエリーの実行時に必要な一時領域として Microsoft SQL Server によって使用されるデータベースです。TEMPDB の初期サイズを少なくとも 100 MB に設定して、自動的な増大に対応し、SQL Server でユーザーのアクティビティに応じて一時データベースを拡張できるようにします。

## 設定パラメータ

追加パラメータは SQL Server のパフォーマンスに直接影響するため、次のガイドラインに従って設定する必要があります。

- **SPIN COUNTER**：このパラメータは、Microsoft SQL Server が指定リソースを取得するために行う最大試行回数を指定します。デフォルトの設定は、ほとんどの場合に適しています。
- **MAX ASYNC I/O**：このパラメータは、実行できる非同期入出力 (I/O) の数を設定します。デフォルトは 32 で、ファイルごとに最大で 32 の未処理の読み込みと 32 の未処理の書き込みを行うことができます。ディスクサブシステムを特殊化していないサーバーでは、この値を増やしても利点はありません。RAM キャッシュや RAID ディスクセットを搭載したインテリジェントディスクコントローラなど、高性能のディスクサブシステムを導入しているサーバーでは、複数の非同期 I/O リクエストを受け入れる機能があるため、この値を増やすことでパフォーマンスが向上することがあります。
- **MAX DEGREE OF PARALLELISM**：このオプションを使用して、Microsoft SQL Server での並列クエリープランの生成方法を設定します。このオプションを 1 に設定して、並列クエリープラン生成を無効にします。この設定は、予期しないクエリープランが生成されるのを避けるために、必ず行ってください。
- **LOCKS**：このオプションを使用して、サーバー全体での使用に Microsoft SQL Server が割り当てるロックの数を指定します。ロックは、テーブルや行などのデータベースリソースへのアクセスを管理するために使用されます。このオプションは 0 に設定して、Microsoft SQL Server がシステム要件に基づいて、動的にロックの割り当てを管理できるようにする必要があります。
- **AUTO CREATE STATISTICS**：このオプションを使用して、クエリーを最適化できるように、必要に応じて、SQL Server でデータベースカラム用の新しい統計を作成できます。このオプションは有効にしてください。
- **AUTO UPDATE STATISTICS**：このオプションを使用して、Microsoft SQL Server で自動的にデータベース統計を管理し、クエリーを効率良く最適化できるように、必要に応じて、統計を更新します。このオプションは有効にしてください。

## Oracle データベース

ここでは、Oracle データベースプラットフォームの EIM のチューニングのヒントについて説明します。

### 過剰なテーブルの断片化の回避

EIM を実行する前に、経験豊富な DBA に相談して、EIM テーブルと Siebel ベーステーブルに挿入するデータを格納するために必要なディスク領域を評価してください。また、たとえば Oracle のサポートを得て、これらのテーブルとインデックスのエクステントのサイズを適切に定義してください。

エクステントの割り当て操作と割り当て解除操作 (truncate コマンドや drop コマンドなど) では、多くの CPU リソースが要求されることがあるため、過剰な拡張を避け、テーブルとインデックスのエクステントのサイズを小さくしておくことが重要です。

### Oracle データベースでセグメントの拡張が発生しているかどうかを確認するには

- 次の SQL 文を使用して、エクステントの数が 10 以上になっているオブジェクトを特定します。

**備考：** 10 個のエクステントは、セグメントのエクステントの目標数ではありません。

```
SELECT segment_name, segment_type, tablespace_name, extents
FROM dba_segments
WHERE owner = (Siebel table_owner)
and extents > 10;
```

断片化を削減するため、適切な保存パラメータを使用してオブジェクトを再構築できます。オブジェクトを再構築するときは、オブジェクトのデフォルトやトリガーなどの問題があるため、常に注意してください。

### EIM テーブルの削除

EIM テーブルからデータを削除するときは、DELETE コマンドではなく、TRUNCATE コマンドを使用します。TRUNCATE コマンドは、データブロックを開放して最高水準点を再設定します。DELETE コマンドではこれらの処理は行われません。このため、処理中に追加ブロックが読み込まれます。また、EIM テーブルのインデックスを削除して再作成し、空のブロックを開放してください。

### アーカイブのログ作成の無効化

初期データのロード時には、アーカイブのログ作成を無効にしておくことをお勧めします。この機能を有効にすると、データのロードが完了した後に、時間を指定した復旧を行うことができます。

## FREELIST パラメータ

複数の EIM プロセスでそれぞれ異なるバッチまたはバッチ範囲を使用する場合は、EIM テーブルに対して複数の EIM プロセスを実行できます。ただし、共有オブジェクトでロックの競合が発生する可能性があります。1 つの EIM テーブルに対して複数のジョブを並列実行するには、EIM 処理で使用されるテーブルとインデックスの FREELIST パラメータが適切に設定されていることを確認する必要があります。

これには、EIM テーブルとインデックス、およびベーステーブルとインデックスが含まれます。このパラメータの値は、レコードの挿入に利用できるメモリに格納される、ブロック ID の数を指定します。通常、この値は、少なくとも 1 つの EIM テーブルに対して実行する並列ジョブの数の半分に設定する必要があります（たとえば、FREELIST を 10 に設定すると、1 つの EIM テーブルに対し、並列ジョブを 20 まで実行できます）。

このパラメータは、オブジェクトが作成されたときに設定されます。このパラメータのデフォルトは 1 です。特定オブジェクトのこのパラメータ値を確認するには、次のクエリーを使用できます。

```
SELECT SEGMENT_NAME, SEGMENT_TYPE, FREELISTS
FROM DBA_SEGMENTS
WHERE SEGMENT_NAME='OBJECT NAME TO BE CHECKED';
```

このパラメータを変更するには、オブジェクトを再構築する必要があります。ここでも、オブジェクトを再構築するときは、オブジェクトのデフォルトやトリガーなどの問題があるため、常に注意してください。

## オブジェクトを再構築するには

- 1 アクセス権を持つテーブルからデータをエクスポートします。
- 2 テーブルを削除します。
- 3 必要な FREELIST パラメータを指定して、テーブルを再作成します。
- 4 データをテーブルにインポートします。
- 5 必要な FREELIST パラメータを指定して、インデックスを再構築します。

## テーブルのキャッシュ

パフォーマンスを向上させるには、頻繁にアクセスする小さなテーブルをキャッシュに置くという方法もあります。BUFFER\_POOL\_KEEP の値は、LRU アルゴリズムでフラッシュされないバッファキャッシュの割り当てを決定します。割り当てを決定したら、メモリに特定のテーブルを配置できます。これにより、このテーブルにアクセスするときのパフォーマンスが向上します。また、初めてテーブルにアクセスした後は、テーブルは常にメモリ内に保持されます。あるいは、テーブルがメモリからはき出され、次回使用時にディスクアクセスが必要になる場合もあります。

確保されている領域に割り当てたメモリの容量は、バッファキャッシュメモリ全体 (DB\_BLOCK\_BUFFERS で定義) から減算される点に注意してください。このような操作を行うのは、S\_LST\_OF\_VAL テーブルが適しています。テーブルをキャッシュに保持するための構文は、次のとおりです。

```
ALTER TABLE S_LST_OF_VAL CACHE;
```

## テーブルの更新

update 文に 255 以上の NVL 関数が含まれている場合は、ハッシュキーがオーバーフローするため、Oracle では誤ったデータが更新されます。これは、Oracle に固有の問題です。この問題を避けるため、update 文で使用する NVL 関数は 255 未満にしてください。

## IBM DB2 UDB

ここでは、IBM DB2 UDB データベースプラットフォームの EIM のチューニングのヒントについて説明します。

ここに記載されている情報と『Siebel Bookshelf』に収録されている情報のほかに、IBM から Redbook 『Siebel 7.8 with IBM DB2 UDB V8.2 Handbook』が出版されています。この Redbook は、IBM DB2 UDB データベースプラットフォーム上の Siebel Business Applications に関する追加情報を提供します。詳しくは、<http://www.redbooks.ibm.com> を参照してください。

次に示す、Siebel EIM のチューニングに関するヒントを検討してください。

- EIM テーブルのロード時に IBM DB2 load replace オプションを使用します。  
**備考：** IBM DB2 ロードオプションを使用して EIM テーブルを削除することもできます。これを行うには、空の (Null) 入力ファイルを使用して、LOAD REPLACE モードでロードオプションを実行します。これにより、指定した EIM テーブルが即座に削除されます。
- EIM テーブルとベーステーブルに独立したテーブルスペースを使用します。
- 大規模な EIM ロードの場合や、多数の EIM タスクが並行して実行される場合は、個々の EIM テーブルを独立したテーブルスペースに配置します。
- EIM と大規模なベーステーブルには、大きいページサイズを使用します。これまでの経験から、ページサイズを 16 KB または 32 KB にすると良好なパフォーマンスが得られることがわかっています。より大きいページサイズを使用すれば、1 ページにより多くのデータを収めることができるとともに、インデックス B ツリー構造のレベル数も減少します。
- 同様に、EIM と大規模なベーステーブルには、大きいエクステントサイズを使用します。
- テーブルスペースコンテナが論理ディスクおよび物理ディスク、ならびにデータベースサーバーの入出力 (I/O) コントローラに均等に分配されていることを確認します。
- EIM テーブルとターゲットベーステーブルに独立したバッファプールを使用します。EIM の初期ロードはかなり大規模であり、通常はオンラインユーザーがいないため、相当量のメモリを EIM とベーステーブルバッファプールに割り当てることをお勧めします。

- 新しいデータをロードした後に、ディスク上のデータのクラスタ化が不十分である場合は、テーブルを再編成します。RUNSTATS コマンドを実行した結果、クラスタ化が低下したこと（クラスタ化インデックスのクラスタ化が 80% 未満である）、およびテーブルの再編成が必要であることがわかった場合は、REORGCHK コマンドを使用します。REORGCHK コマンドの使用法について詳しくは、Siebel SupportWeb の「FAQ 2072」を参照してください。

**備考：** 変換スケジュールに時間を割り当て、エンドユーザーが新しいデータを含むシステムにアクセスする前にテーブルの再編成と統計データの収集を行えるようにします。

- IBM DB2 スナップショットモニタを使用して、パフォーマンスが最適であるかどうかを確かめ、パフォーマンスのボトルネックがあれば検出して解決します。
- 初期ロード時には、logretain をオフにできます。ただし、本番環境に移行する前にオンに戻してください。

**備考：** logretain が有効になっているときは、データベースのフルコールドバックアップを行う必要があります。

- EIM テーブルおよび関連するベーステーブルについては、VOLATILE に設定できるようにテーブルを変更します。これを行うと、インデックスがテーブルスキャンよりも優先されます。
- 複数の EIM プロセスが同じカタログテーブルの更新を同時に試みた場合、EIM プロセスを並行して実行すると IBM DB2 UDB データベースでデッドロックおよびタイムアウトが発生します。これを避けるには、EIM 設定ファイル（.IFB ファイル）で UPDATE STATISTICS パラメータを「FALSE」に設定します。
- 各 EIM プロセスで UPDATE STATISTICS を実行すると、データベースサーバーリソースが大幅に消費されます。データベース管理者は、RUNSTATS コマンドを使用して EIM プロセス外で統計データを更新することをお勧めします。
- 次に示す DB2 レジストリ値の設定を考慮します。

レジストリ値	設定
DB2_CORRELATED_PREDICATES =	YES
DB2_HASH_JOIN =	NO
DB2_PARALLEL_IO =	"*"
DB2_STRIPPEDED_CONTAINERS =	RAID デバイスをテーブルスペースコンテナとして使用するとき

- 次に示す DB2 データベースマネージャ設定パラメータの設定を考慮します。

レジストリ値	設定
INTRA_PARALLEL =	NO（大規模なインデックスの作成時に使用される場合があります）
MAX_QUERYDEGREE =	1（大規模なインデックスの作成時には増やすことができます）
SHEAPTHRES =	100,000（利用可能メモリや、SORTHEAP 設定などの要因で決まります）

- 次に示すデータベースパラメータの設定を考慮します。

レジストリ値	設定
CATALOGCACHE_SZ =	6400
DFT_QUERYOPT =	3
LOCKLIST =	5000
LOCKTIMEOUT =	120 (30 ~ 120 の範囲)
LOGBUFSZ =	512
LOGFILESZ =	8000 以上
LOGPRIMARY =	20 以上
LOGRETAIN =	NO (EIM の初期ロード時のみ)
MAXLOCKS =	30
MINCOMMIT =	1
NUM_IOCLEANERS =	データベースサーバーに搭載されている CPU の数
NUM_IOSERVERS =	DB2 コンテナを含むディスクの数
SORTHEAP =	10240 (この設定は EIM の初期ロード専用です。本番では、64 ~ 256 の範囲に設定してください。)  SORTHEAP に指定した値は、SHEAPTHRES の値を変更した結果に影響を及ぼします。たとえば、SORTHEAP = 10000 の場合は、SHEAPTHRES = 100000 と設定すると、EIM バッチを 9 個までしか実行できなくなります。  複数の EIM バッチを並行して実行する場合は、必ず十分な物理メモリを割り当ててください。そうすれば、メモリスワッピングやメモリページングが発生しなくなります。
STAT_HEAP_SZ =	8000

## IBM DB2 UDB for z/OS

DB2 の設定については、データベースマネージャ設定パラメータ (DSNZPARAM) の一覧 (JCL から) が『Implementing Siebel Business Applications on DB2 UDB for z/OS and OS/390』に記載されています。

IBM DB2 について詳しくは、次のトピックを参照してください。

- 158 ページの「EIM のための IBM DB2 ローディングプロセス」
- 159 ページの「IBM DB2 ローディングプロセスに関する一般的な推奨事項」

## EIM のための IBM DB2 ローディングプロセス

図 3 は、IBM DB2 のロードプロセスを示しています。

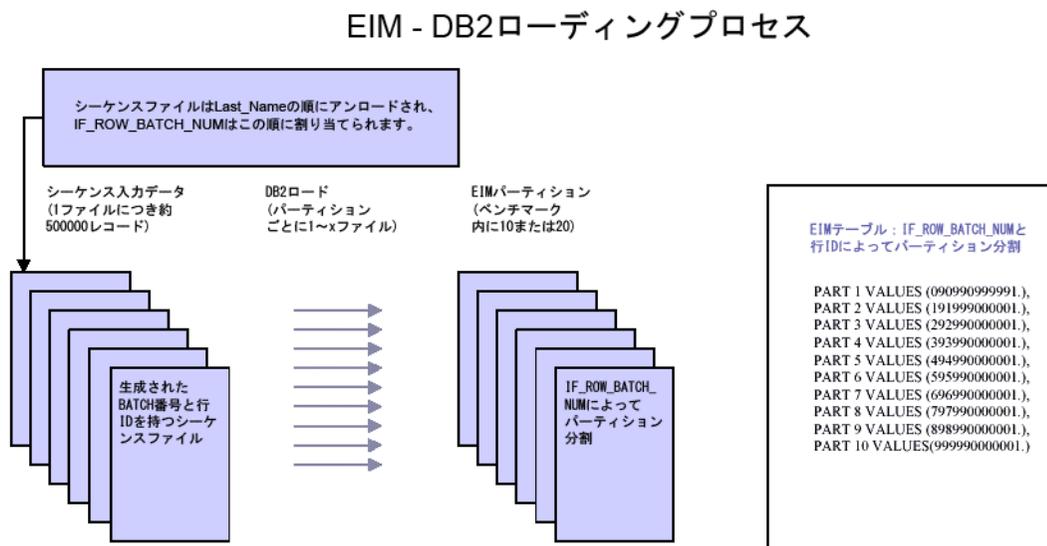


図 3 EIM のための IBM DB2 ローディングプロセス

詳しくは、『Siebel Enterprise Integration Manager Administration Guide』を参照してください。

## IBM DB2 ローディングプロセスに関する一般的な推奨事項

次に示す一般的な推奨事項は、EIM のために IBM DB2 のローディングプロセスを実行する際に適用されます。

- .IFB ファイルで ONLY/IGNORE BASE TABLES パラメータまたは ONLY/IGNORE BASE COLUMNS パラメータを使用して、EIM が実行する処理の量を減らします。IGNORE BASE COLUMNS オプションを使用すると、外部キーを除外することができます。これにより、処理要件と解決できないキーのエラーログエントリが両方とも減ります。キーワード ONLY と IGNORE は、相互に排他的である点に注意してください。たとえば、次の設定では、オプション IGNORE BASE TABLES と ONLY BASE COLUMNS が除外されます。

```
ONLY BASE TABLES = S_CONTACT
```

```
IGNORE BASE COLUMNS = S_CONTACT.PR_MKT_SEG_ID
```

- 親と子を別々にインポートします。取引先、住所、チームなどのデータは、可能な限り、同じ EIM テーブルを使用して同時にロードしてください。
- バッチ内のすべての EIM テーブルデータをデータベースキャッシュに格納できるようなバッチサイズを使用します（約 2,000 レコード、DB2/390 の場合は 5000 レコード）。拡張パラメータを使用して、バッチ範囲を扱えるように EIM を設定することができます。変数名を .IFB ファイルに指定してください。
- 同じ EIM テーブルに対して複数の EIM プロセスを実行できます。これは、すべての EIM プロセスが異なるバッチまたはバッチ範囲を使用する場合に限られます。ただし、EIM パフォーマンスの制限となるのは、主としてアプリケーションサーバーではなくデータベースです。同じベーステーブルに対して複数の EIM ストリームが同時に実行されると、共有オブジェクトに対してロックの競合が発生することがあります。異なるベーステーブル（たとえば、S\_ORG\_EXT と S\_ASSET）に対して複数の EIM ジョブストリームを同時に実行することができます。
- 可能であれば、営業時間外の最小ユーザー活動時間中に EIM を実行します。これにより、接続ユーザーの負荷が減少し、EIM プロセスが最大処理容量を使用できるようになります。
- 初期データベースロード時には、Docking: Transaction Logging システム属性を（[管理 - アプリケーション] > [システム設定] で）「FALSE」に設定します。これにより、モバイルクライアントの同期に使用される Siebel ドッキングテーブルに対するトランザクション活動が削減されます。
- [サーバー管理] 画面からデータベーストリガーを削除して、データベーストリガーを無効にします。これはスループットレートの向上にも役立ちます。EIM のロードが完了したら、トリガーを再適用してください。トリガーがないと、ワークフローポリシーや割当マネージャなどのコンポーネントは、新しいデータおよび更新されたデータには機能しません。
- 処理する EIM テーブルに必須カラムの ROW\_ID、IF\_ROW\_STAT、および IF\_ROW\_BATCH\_NUM が正しく入力されていることを必ず確認します。これを行う最も効果的なタイミングは、データソースまたはステージング領域から EIM テーブルを入力したときであり、データをクレンジングした後です。
- 特定の処理要件がない限り、EIM 処理のためにデータを EIM テーブルにロードする前に、EIM テーブルが空であることを確認します。EIM テーブル内の競合を回避するために、適切なバッチ番号が使用されていることを必ず確認してください。自動化ルーチンを使用している場合は、データソースからのロード間に EIM テーブルを切り捨てるとパフォーマンスの維持に役立ちます。

- IBM DB2 データベース上で Siebel アプリケーションを実行しているときに、S\_LST\_OF\_VAL ベーステーブルの更新中に EIM が応答を停止することがときどきあります。これはデータに問題があるためです。S\_LST\_OF\_VAL ベーステーブルの BU\_ID カラムには、明確に異なる値が 1 つしかないかほとんどありません。そのため、ほとんどまたはすべての行に同じ BU\_ID カラム値があると、DB2 オプティマイザーは S\_LST\_OF\_VAL テーブルのすべての行に対してテーブルスキャンを実行します。

この問題を避けて、クエリーを迅速化するには、次の SQL 文を実行して統計データを変更してください。

```
update sysibm.sysindexes set firstkeycard=1000 where name='S_LST_OF_VAL_M2';  
  
update sysibm.syscolumns set colcard = 1000 where tbname='S_LST_OF_VAL' and  
name='BU_ID';
```

**備考：** 処理するデータによっては、事前に他の SQL 文の実行が必要な場合があります。

## EIM を最適化するためのデータ管理ガイドライン

EIM ローディングプロセスを実行する際には、次の推奨事項が適用されます。

- EIM マッピングチャートは、EIM テーブルカラムの多くがその値をレガシーデータベースフィールドからではなく、不変のリテラル文字列から取得することを示しています。この種の情報を EIM テーブルに入れないようにしてください。実際のレガシーデータを EIM テーブルからベーステーブルへ移動する処理速度が低下するためです。
- EIM には、不変のリテラル文字列をベーステーブルのカラムに入力する代替手段があります。つまり、DEFAULT COLUMN 文を使用する方法です。この方法を使用すれば、ベーステーブルにインポートしなければならないデフォルトリテラルを指定することができ、リテラルを EIM テーブルから取得する必要がありません。たとえば、EIM マッピングチャートでは、Default Organization は EIM\_CONTACT の CON\_BU の定数値として示されています。この値が S\_CONTACT の BU\_ID に移動されます。同じ結果は、.IFB ファイル中の設定 DEFAULT COLUMN = CON\_BU, Default Value によって得られます。このほかにも、リテラル文字列を EIM テーブルから .IFB ファイルに移動する機会が多くあります。

## EIM を最適化するための実行パラメータガイドライン

次に示す推奨事項は、EIM ローディングプロセスを実行する際の実行パラメータ設定のためのものです。

- TRIM SPACES を FALSE に設定しないでください。TRIM SPACES パラメータを使用すると、末尾のスペースが Siebel ベーステーブルに格納されます。Siebel アプリケーションは、1 文字を超えるほとんどすべてのテキストカラムで VarChar を使用するため、この処理が行われるとディスクスペースの使用効率が低下する可能性があります。TRIM SPACES を FALSE に設定した場合も、テーブルスペースデータ用の貴重なバッファプールスペースが浪費される可能性があります。
- IGNORE BASE TABLES パラメータまたは ONLY BASE TABLES パラメータのいずれかを使用して、挿入または更新するテーブルの数を制限します。ONLY BASE TABLES パラメータの使用をお勧めします。これは、リストが通常はより短く、自己文書化型であるためです。これらのパラメータを使用すると、EIM がロードを試みるテーブルの数が制限され、さらにユーザーインターフェイスが使用しないテーブルのスペースが節約されるので、パフォーマンスが向上します。
- IGNORE BASE COLUMNS パラメータまたは ONLY BASE COLUMNS パラメータのいずれかを使用して、挿入または更新するテーブルの数を制限します。ONLY BASE COLUMNS パラメータの使用をお勧めします。これは、リストが通常はより短く、自己文書化型であるためです。これらのパラメータを使用すると、EIM が解決を試みる外部キーの数が制限されるため、パフォーマンスが向上します。
- USING SYNONYMS パラメータを .IFB ファイルで FALSE に設定します。この論理演算子は、EIM に対して、インポート時に取引先略称の処理を省略するように指示します。これにより処理の量が削減されます。取引先の住所を複数使用する予定がある場合は、USING SYNONYMS パラメータを FALSE に設定しないでください。さもないと、EIM は住所を適切な取引先に結び付けなくなります。
- ベーステーブルがすでに完全にロードされていて、しかもそのテーブルが他のテーブルのロードおよび更新に使用される EIM テーブルのプライマリテーブルであるときは、挿入を抑制します。コマンド形式は「INSERT ROWS = *table name*, FALSE」です。
- ベーステーブルがすでに完全にロードされていて、外部キーの追加などの更新を必要としないが、そのテーブルが他のテーブルのロードおよび更新に使用される EIM テーブルのプライマリテーブルであるときは、更新を抑制します。コマンド形式は「UPDATE ROWS = *table name*, FALSE」です。

## EIM タスク時の Siebel Server の監視

Siebel Server の監視に際しては、Siebel アプリケーションサーバーおよび Siebel データベースサーバーで EIM タスクを実行するために、十分なプロセッサおよびメモリリソースを割り当ててあることが前提となります。

Microsoft Windows 2000 を Siebel Server のオペレーティングシステムとして使用している場合は、Microsoft Windows のパフォーマンスモニタを使用して、ハードウェアが使用しているプロセッサおよびメモリの量を確認することができます。

Sun Solaris または IBM AIX を Siebel Server のオペレーティングシステムとして使用している場合は、*vmstat* および *iostat* を使用して、ハードウェアが使用しているプロセッサおよびメモリの量を確認することができます。



# 11 パフォーマンスのための Siebel Remote のチューニング

この章では、Siebel Remote のパフォーマンスの向上に貢献する可能性があるチューニングについて説明します。ここで説明する内容は、次のとおりです。

- 163 ページの「Siebel Remote について」
- 164 ページの「Siebel Remote Server のコンポーネントのチューニング」
- 167 ページの「Siebel Remote の展開におけるモバイル Web クライアントのチューニング」

Siebel Remote について詳しくは、『Siebel Bookshelf』の『Siebel Remote/Replication Manager 管理ガイド』を参照してください。Siebel SupportWeb には、Siebel Remote のパフォーマンスの問題を解決するための「Troubleshooting Steps」、「Technical Notes」などのドキュメントも含まれます。

## Siebel Remote について

モバイル Web クライアント（通常、モバイル環境の切断モードでリモート操作される）は、Siebel Remote を使用して Siebel Server に接続し、更新されたデータやファイルを交換できます。このプロセスは同期と呼ばれます。Siebel Remote ではモバイルコンピューティングが実現されるため、出先の社員は、サーバーに接続しているユーザーやモバイルユーザーなど、社内すべてのユーザーと最新の情報を共有できます。

Siebel Remote は、次のコンポーネントを使用してデータとファイルの交換を管理します。

- Database Extract (エイリアス DbXtract)
- Generate New Database (エイリアス GenNewDb)
- Parallel Database Extract (エイリアス PDbXtract)
- Synchronization Manager (エイリアス SynchMgr)
- Transaction Merger (エイリアス TxnMerge)
- Transaction Processor (エイリアス TxnProc)
- Transaction Router (エイリアス TxnRoute)

各コンポーネントについて詳しくは、『Siebel Remote/Replication Manager 管理ガイド』を参照してください。

164 ページの「Siebel Remote Server のコンポーネントのチューニング」の項では、これらのコンポーネントを設定して Siebel Remote の展開のパフォーマンスを最適化する方法について説明します。

## Siebel Remote Server のコンポーネントのチューニング

ここでは、Siebel Remote Server の特定コンポーネントのパフォーマンスを向上する方法について説明します。これには、次の項目があります。

- 164 ページの「Database Extract コンポーネントと Parallel Database Extract コンポーネントのスループットの向上」
- 165 ページの「Transaction Router コンポーネントのチューニング」

### Database Extract コンポーネントと Parallel Database Extract コンポーネントのスループットの向上

次のリストに、Database Extract コンポーネントと Parallel Database Extract コンポーネントのスループットを向上するためのヒントを示します。

- 複数インスタンスの実行

Database Extract コンポーネントの複数インスタンスを同時に実行することによってスループットが向上します。Database Extract コンポーネントの各インスタンスには、一時テーブルが必要です。このテーブルは S\_DOCK\_INITM\_*N* という名前です。ここで、*N* の値は、TS Table Number (エイリアス TSTableNum) パラメータと同じです。TS Table Number には、Database Extract コンポーネントで使用する一時テーブルの数を指定します。たとえば、TS Table Number が 1 の場合、一時テーブル 1 (S\_DOCK\_INITM\_1) は Database Extract コンポーネントの実行中のインスタンスで使用されます。

デフォルトでは、48 個の一時テーブルを使用できます。テーブルを追加する必要がある場合は、Siebel Tools を使用して作成します。

使用する一時テーブルの推奨数は、使用しているデータベースプラットフォームにより異なります。たとえば、次のように設定します。

- Microsoft SQL Server と IBM DB2 Universal Database

実行する Database Extract コンポーネントの各インスタンスに対して 1 つの一時テーブルを使用します。たとえば、Database Extract コンポーネントのインスタンスを 11 個実行する場合は、11 個の一時テーブルを使用します。

- Oracle Enterprise Server

使用する一時テーブルの数は、このデータベースサーバーがアクセスできる共有プールのサイズにより異なります。共有プールのサイズが 300 MB 未満の場合、一時テーブルを 1 つ使用し、Database Extract コンポーネントのインスタンスを 1 つ実行することをお勧めします。共有プールのサイズが 600 MB を超える場合、Database Extract コンポーネントの各インスタンスに 1 つの一時テーブルを使用するとスループットが向上する場合があります。

■ Transaction Router コンポーネント

Database Extract コンポーネントまたは Parallel Database コンポーネントの複数のインスタンスを実行する場合は、テーブルのロックを回避するために Transaction Router コンポーネントを停止します。

■ ユーザーのリストを同時に抽出

Database Extract コンポーネントと Parallel Database Extract コンポーネントを使用して、抽出するユーザーをリストあたり 50 ~ 100 ユーザーのグループに分割し、これらのユーザーリストを同時に抽出します。これらのコンポーネントによって、リストのすべてのユーザーの Enterprise 可視性データを一度に抽出できます。

■ Truncate TS テーブルパラメータを「True」に設定

通常の場合、S\_DOCK\_INITM\_N からの削除はログ出力され、長時間かかるので、このパラメータ（エイリアス TruncateTSTable）を「True」に設定するとパフォーマンスが向上することがあります。

■ デフラグメントテーブル

データおよび S\_DOCK\_INITM\_N テーブルを抽出するデフラグメントテーブルです。

前述のコンポーネントとパラメータについて詳しくは、『Siebel Remote/Replication Manager 管理ガイド』を参照してください。Database Extract コンポーネントのパフォーマンス上の問題については、Siebel SupportWeb の「Troubleshooting Steps 15」を参照してください。

## Transaction Router コンポーネントのチューニング

ここでは、次のソースから発生する Transaction Router コンポーネントのパフォーマンス上の問題を解決または回避する方法について説明します。

■ 可視性に関連するトランザクション

■ ドッキングルールとデータ配布

■ 実行速度が遅いクエリー

■ Transaction Router のスループットの向上

Transaction Router のパフォーマンス上の問題について詳しくは、Siebel SupportWeb の次のドキュメントを参照してください。

■ Troubleshooting Steps 8

Transaction Router のパフォーマンス上の問題を診断および解決する方法について説明します。

■ Troubleshooting Steps 38

Siebel Remote の実装のトランザクションバックログを監視および管理する方法について説明します。

## 可視性に関連するトランザクション

Transaction Router のパフォーマンスの問題の根本原因が、可視性に関連したトランザクションにあると診断した場合は、次の 2 つの解決策を検討します。

- すべてのモバイルユーザーとエリアノードの再抽出  
詳しくは、『Siebel Remote/Replication Manager 管理ガイド』を参照してください。
- Transaction Router コンポーネントのタスクが、バックログをクリアするまで処理を続行できるようにする  
Transaction Router が可視性に関連するすべてのトランザクションの処理を完了すると、バックログの処理が速くなるはずですが、Transaction Router の追加タスクを起動してパフォーマンスを向上することもできますが、Siebel Server またはデータベースエンジンのサポート数を超えるタスクは起動しないでください。

## ドッキングルールとデータ配布

Transaction Router のパフォーマンスの問題の根本原因がドッキングルールに関連するトランザクションにあると診断した場合は、Siebel 技術サポートにサービスリクエストを記録して、次の情報を提供してください。

- Transaction Router のタスクの RDBMS トレース
- Transaction Router のログファイル
- `SIEBEL_ROOT¥siebsrvr¥Docking¥txnproc` ディレクトリにある Transaction Router が処理している .dx ファイル
- visrule スクリプトの実行結果  
visrule スクリプトについて詳しくは、Siebel SupportWeb の「Troubleshooting Steps 8」を参照してください。

## 実行速度が遅いクエリー

Transaction Router のパフォーマンスの問題の根本原因が実行速度の遅いクエリーにあると診断した場合は、次の項目についてデータベース管理者に問い合わせてください。

- 実行速度が遅いクエリーに必要なテーブルのすべてのインデックスが存在し、有効であること。  
すべてのインデックスが存在し、有効であることを確認する方法については、『Siebel Data Model Reference』を参照してください。
- 実行速度が遅いクエリーに必要なテーブルとインデックスにデフラグが必要かどうかを確認します。

## Transaction Router のスループットの向上

次の要因は、Transaction Router コンポーネントのスループットに影響する場合があります。

- Siebel EIM (Enterprise Integration Manager) のバッチサイズが大きい

可能な場合は、Siebel EIM がデータをインポートする際に処理するバッチのサイズを小さくすることをお勧めします。さらに、トランザクションをマスタートランザクションログ (S\_DOCK\_TXN\_LOG) ではなく、Siebel File System にログ出力することをお勧めします。

詳しくは、『Siebel Enterprise Integration Manager Administration Guide』を参照してください。

- Siebel 割当マネージャのバッチサイズが大きい

バッチファイルのサイズが大きいと Transaction Router コンポーネントのパフォーマンスに影響するので、可能な場合は、Siebel 割当マネージャのバッチファイルのサイズを小さくします。詳しくは、『Siebel Assignment Manager 管理ガイド』を参照してください。

前述の 2 つの場合には、変更する前に Transaction Router コンポーネントのスループットの向上が Siebel EIM コンポーネントと Siebel 割当マネージャコンポーネントのスループットの低下より重要であるかどうかを確認する必要があります。

## Siebel Remote の展開におけるモバイル Web クライアントのチューニング

ここでは、Siebel Remote の展開におけるモバイル Web クライアントのパフォーマンスを最適化する方法について説明します。次のトピックが含まれています。

- [167 ページの「アプリケーション設定ファイルのパラメータの最適化」](#)

- [169 ページの「同期のための最善の方法」](#)

- [169 ページの「適切なルーティングモデルの選択」](#)

モバイル Web クライアントのパフォーマンスのチューニングの追加情報については、[第 5 章「パフォーマンスのための Siebel Web クライアントのチューニング」](#)を参照してください。

## アプリケーション設定ファイルのパラメータの最適化

ここでは、Siebel アプリケーション設定ファイルで指定するパラメータの値を変更することにより、モバイル Web クライアントのパフォーマンスを最適化する方法について説明します。

## DockTxnsPerCommit

このパラメータの値には、Siebel Remote がコミットを実行する前にローカルデータベースに適用するトランザクションの数を指定します。多くのトランザクションが絶えず作成される環境で、DockTxnsPerCommit の値を大きくすると、大量のデータの初期化タスクと同期タスクに必要な時間が短縮される場合があります。そのようなシナリオでは、さまざまな値 (1000、2000、3000 など) をテストし、各自の環境にどの値が最も適しているかを決定します。

DockTxnsPerCommit パラメータは、アプリケーション設定ファイルの [Local] セクションにあります。デフォルト値は 500 です。

## AutoStopDB

ユーザーが Siebel アプリケーションが終了した後も、SQL Anywhere データベースエンジンの実行が継続されるように、AutoStopDB が「FALSE」に設定されていることを確認します。これによって、後で Siebel アプリケーションを再起動するために必要な時間を短縮できます。AutoStopDB を「TRUE」に設定すると、SQL Anywhere データベースエンジンは Siebel アプリケーションが終了したときに自動的に停止します。

AutoStopDBIn パラメータは、アプリケーション設定ファイルの [Local] セクションで設定します。デフォルト値は FALSE です。

## SQL Anywhere データベースエンジンのキャッシュへのメモリの割り当て

SQL Anywhere データベースエンジンが使用できるメモリ量 (特にキャッシュ) は、パフォーマンスに影響する主要な要因の 1 つです。SQL Anywhere データベースエンジンはメモリをさまざまな目的で使用しますが、主要な用途の 1 つは、必要になるたびにデータベースからデータを取得する必要があるように、繰り返しアクセスするデータを保持することです。

キャッシュに使用するメモリ量を指定するには、siebel.cfg ファイルの ConnectString パラメータの -c コマンドラインオプションに値を設定します。たとえば、次のように入力します。

```
-c15m -ch25m
```

これによって、15 MB 以上のメモリがキャッシュに割り当てられます。パラメータ値 (ch) は、キャッシュに割り当てるメモリ量の上限 25 MB を指定しています。

デフォルトでは、これらのパラメータ値は利用可能なメモリ総量のパーセンテージとして表します。たとえば、次のように入力します。

```
-c5p -ch7p
```

これによって、キャッシュメモリには、利用可能メモリの 5% 以上、7% 以下が割り当てられます。

SQL Anywhere データベースエンジンのメモリキャッシュに多くのメモリを割り当てると、ローカルマシンの他のアプリケーションが使用できるメモリ量は減少します。

ガイドラインとして、マシンメモリ総量の 80% から、マシン上のすべてのアプリケーションで通常使用するメモリ量を引いた量を使用してください。たとえば、512 MB のメモリを搭載するローカルマシンで、Siebel アプリケーションを含むすべてのアプリケーションをロードした後の通常使用量が 328 MB の場合は、SQL Anywhere データベースエンジンのキャッシュに 82 MB のメモリを割り当てることができます。

SQL Anywhere データベースエンジンのキャッシュに割り当てることができるメモリ量の上限を決定するためのテストも実行する必要があります。

**注意：** キャッシュに割り当てるメモリ量を、ページングが発生するレベルまで増やすことは避けてください。ページングは、メモリ使用量が利用可能なメモリ総量を超えた場合に発生し、パフォーマンスの低下の原因になります。

## ソート照合

SortCollation パラメータは、ローカルデータベースからのデータ取得を最適化するためにバイナリに設定する必要があります。SortCollation パラメータは、アプリケーション設定ファイルのデフォルト部分ではありません。Siebel アプリケーションの設定ファイルに手動で追加する必要があります。SortCollation パラメータの値は、アプリケーション設定ファイルの [Local] セクションで設定します。

このパラメータについては、『Siebel システム管理ガイド』を参照してください。SortCollation の現在の状態を確認するには、Siebel SupportWeb の「Alert 801」を参照してください。

## 同期のための最善の方法

ここでは、Siebel モバイル Web クライアントと Siebel Remote Server 間のデータ同期を最適化する際に役立つポイントを示します。次のポイントに留意してください。

### ■ 頻繁に同期する

頻繁に同期すると、各同期セッションにおいて転送してコミットするトランザクションの数を減らすことができます。同期セッション間の時間が長くなるほど、送信するデータは多くなります。

### ■ Siebel モバイル Web クライアントに対して TrickleSync を有効にする

Siebel モバイル Web クライアントが Siebel Enterprise ネットワークに接続するたびに、TrickleSync はデータベース同期を実行します。詳しくは、『Siebel Remote/Replication Manager 管理ガイド』を参照してください。

### ■ 時間ベースのフィルタを使用して、サーバーからクライアントへの特定の日付より古いデータの転送を抑止する

### ■ ドッキングオブジェクトを無効にする

## 適切なルーティングモデルの選択

パフォーマンスを向上する 1 つの方法は、リモートユーザーに送信するデータ量を減らすことです。そのためには、適切なルーティングモデルを選択します。適切なルーティングモデルがない場合は、各自の環境に適したルーティングモデルの開発を支援する Oracle の Siebel エキスパートサービスに問い合わせてください。



# 12 パフォーマンスのための顧客設定のチューニング

この章では、Siebel Tools または Siebel スクリプト言語を使って行う顧客設定により Siebel アプリケーションのパフォーマンスに関連して発生する一般的な問題を回避する方法について説明します。ここで説明する内容は、次のとおりです。

- 172 ページの「顧客設定のための一般的な最善の方法」
- 179 ページの「Siebel スクリプトに関する最善の方法」
- 183 ページの「データオブジェクト層についての最善の方法」
- 188 ページの「ビジネスオブジェクト層についての最善の方法」
- 192 ページの「ユーザーインターフェイスオブジェクト層についての最善の方法」

アプリケーションの開発情報については、『Siebel Bookshelf』および『Siebel Tools Online Help』の次のマニュアルを参照してください。

- 『Configuring Siebel Business Applications』
- 『Using Siebel Tools』
- 『Siebel Developer’s Reference』
- 『Siebel Object Types Reference』
- 『Siebel Object Interfaces Reference』
- 『Siebel eScript Language Reference』
- 『Siebel VB Language Reference』

## 顧客設定のための一般的な最善の方法

ここでは、Siebel Tools を使った顧客設定のための一般的な最善の方法をいくつか紹介します。

ハードウェアリソースを最適に使用し、システムを適切に設定することにより、パフォーマンス面の目標を達成するために役立ちます。リソースと要件について慎重に考慮し、システムパフォーマンスを継続的にテストおよび監視する必要があります。

Siebel アプリケーションのアーキテクチャでは、データベースのインデックス、データキャッシュ、RDBMS カーソル、効率的な SQL 生成、ネイティブデータベース API などを使用して最適なパフォーマンスを引き出すための設計とチューニングがなされています。ただし、カスタム設定には、さまざまなパフォーマンス面の落とし穴が潜在しています。また、そのパフォーマンス面での影響は、大きなデータベースを持ち、サーバー間で大量のデータ配信を行う環境で大きくなる場合があります。このような問題を回避するために、ここで示すガイドラインとその他のマニュアルのガイドラインに従います。

この項のトピックに加えて、次の項目も参照してください。

- 179 ページの「Siebel スクリプトに関する最善の方法」
- 183 ページの「データオブジェクト層についての最善の方法」
- 188 ページの「ビジネスオブジェクト層についての最善の方法」
- 192 ページの「ユーザーインターフェイスオブジェクト層についての最善の方法」

詳しくは、『Configuring Siebel Business Applications』および『Siebel Bookshelf』の他のマニュアル、その他の資料を参照してください。

## その他の設定ガイドライン

次に、最適なパフォーマンスを維持するために必要になるその他の設定ガイドラインをいくつか紹介します。

- **インデックス化されていないカラムまたは結合されたカラムのソート条件の使用を避ける**：詳しくは、183 ページの「ソートおよび検索時のデータベースインデックスの管理」および他の関連トピックを参照してください。
- **ケース・インセンシティブティ（大文字小文字の区別を行わない）を使用しない**：大文字と小文字を区別しないクエリーを使用すると、大文字と小文字を区別しないデータベース操作をサポートするために、データベースレベルの複雑さが追加されるので、パフォーマンスの問題が発生する可能性が高くなります。

ケース・インセンシティブティを有効にする前に、ビジネス要件とパフォーマンスの基準をよく確認することを強くお勧めします。さらに、この機能を有効にする場合は、本番データベースの完全なコピーを使用してパフォーマンスをテストする必要があります。パフォーマンスへの影響は、設定の複雑さおよび本番データベースのサイズに従って増大します。

設定の最適化と要件の確認については、Siebel エキスパートサービスに問い合わせることをお勧めします。ケース・インセンシティブティはデータベースプラットフォームの制約なので、データベースプラットフォームベンダーにも確認する必要があります。

アプリケーションまたは指定したフィールドに対するケース・インセンシティブティの設定については、『Siebel アプリケーション管理ガイド』を参照してください。

- **クエリーでのケース・インセンシティブティの使用を制限する**：大文字と小文字を区別する検索は、大文字と小文字を区別しないクエリーよりパフォーマンスが良くなります。Siebel アプリケーションでは、デフォルトで大文字と小文字を区別します。ケース・インセンシティブティは、アプリケーション全体または指定したフィールドに対して有効にできます。一般に、データベースのサイズが大きくなるほど、ケース・インセンシティブティのクエリーが返すレコード数が多くなり、データベース全体のパフォーマンスは低下します。全体的なパフォーマンスは、ケース・インセンシティブティのクエリーを実行するユーザー数にも影響を受けます。エンドユーザーは、クエリーでの大文字と小文字の区別の有無を強制的に指定できます。

アプリケーションまたは指定したフィールドに対する大文字と小文字の区別の設定については、『Siebel アプリケーション管理ガイド』を参照してください。

- **過度に複雑なユーザーインターフェイスの設定を回避します**。一般的に、1 つのビューに多数のアプレットを含めたり（通常は 4 アプレットまで）、1 つのアプレットに多数のフィールドを含めたりしないでください。
- **ビュー内のビジネスコンポーネントの数を制限します**。ビュー内のアプレットで使用されている各種のビジネスコンポーネントの数が過度に多くなると、ビューを開いたときのデータの表示が遅くなる場合があります。これは、各アプレットにデータを読み込むことが必要になるためです。
- **ビュー内の仮想ビジネスコンポーネントの数を制限します**。単一のビューで使用する仮想コンポーネントの数は、2 つまでにしてください。
- **ビジネスコンポーネントまたはアプレット内のフィールド数を制限します**。ビジネスコンポーネント内のフィールド数、またはリストアプレット内のリストカラム数に制限はありません。ただし、ビジネスコンポーネントにアクティブなフィールドが多数含まれる場合は、パフォーマンスが劣化します。また、データベースシステムによっては、処理不可能な大きさのクエリーが生成される場合もあります。[188 ページの「アクティブなフィールドの数の制限」](#)も参照してください。

特に、関連するビューでマスターアプレット内に表示されるフィールド数を削減します。この情報は静的なものであり、不要な場合もあります。これによって、ビュー上でユーザーがスクロールしなくてもデータを表示できる領域が増えます（これにより得られる利点には、使い勝手の良さもあります）。

エンドユーザーは、[表示カラム] メニューオプションを使用することにより、リストアプレットに表示されるフィールド数を増減させることができます。しかし、各アプレットに表示する最適なデフォルトのフィールド数を指定しておくことをお勧めします。また、デフォルトで非表示になるフィールドを含めて、最小限必要になるフィールドの総数を指定しておくこともお勧めします。

- **必須フィールドの数の制限**：必須フィールドは、データベースクエリーにより必ず取得されます。その結果、ビジネスコンポーネントの必須フィールド（Required ユーザープロパティが「TRUE」のフィールド）の数を制限すると、パフォーマンスが向上します。[188ページの「アクティブなフィールドの数の制限」](#)も参照してください。
- **返却されるレコードの数を制限します**。ビジネスコンポーネントに返されるレコードの数を制限するには、ビジネスコンポーネントまたは該当するアプレットまたはリンクに検索条件を追加したり、ビュー上にデフォルトの定義済みクエリーを定義したりできます。
- **ビジネスコンポーネント内の結合、拡張テーブル、およびプライマリ ID フィールドの数を制限します**。結合により、メインテーブル内の各行の取得時に、結合テーブルへの行取得処理が余分に発生することから、パフォーマンスが低下します。拡張テーブルおよびプライマリ ID フィールドでも結合が使用されるので（ただし、明示的に定義されるものではなく、暗黙的に定義されます）、行ごとに追加の取得処理が発生します、

ビジネスコンポーネント内で定義する結合、拡張テーブル、およびプライマリ ID フィールドが増えると、それに伴ってメインテーブル以外のテーブルで要求される行の取得回数が増大し、結果としてパフォーマンスの低下につながります。

- **フィールド内の Link Specification プロパティの使用を制限します。** フィールドの Link Specification プロパティを「TRUE」に設定した場合も、パフォーマンスが低下することがあります。この値を TRUE に指定すると、このフィールドの値がリンクを通じて詳細ビジネスコンポーネントのフィールドにデフォルト値として渡されます。

マスタービジネスコンポーネントが（現在のビジネスオブジェクト内に）1 つ以上の詳細ビジネスコンポーネントとのリンク関係を持つ場合で、これらの詳細ビジネスコンポーネントがいずれかのフィールドの Pre Default Value プロパティ、Post Default Value プロパティ、または Calculated Value プロパティで「Parent」式を使用するときこの設定が必要になります。マスタービジネスコンポーネントは、表示されるすべての詳細レコードにフィールド値を提供する必要があります。

Force Active プロパティの場合と同様に、Link Specification プロパティが「TRUE」に設定されたフィールドは、ビジネスコンポーネントに対するクエリーが実行されるたびに取得されます。
- **外部結合ではなく内部結合を使用します。** 内部結合は、すべての外部キーの参照先が有効である場合に、結合テーブルで使用でき、その結果としてオーバーヘッドが軽減されます。

たとえば、詳細ビジネスコンポーネントからそのマスターへの結合では、マスターの存在が保証されます。Join オブジェクト定義の Outer Join Flag プロパティを「FALSE」に設定することにより、結合を内部結合として設定できます。これにより、この結合を使用するクエリーのパフォーマンスが向上します。一般に、二重外部結合は使用しないでください。
- **多対多リンクの Cascade Delete を適切に設定します。** Link オブジェクト定義の Cascade Delete プロパティは、多対多リンクで使用する場合に正しく設定する必要があります。そうしないと、関連アプレット内の最初の挿入または削除が異常に低速になります。多対多関係で使用されるリンクオブジェクトの定義では、Inter Table プロパティに非 NULL 値を含みます。このようなリンクの Cascade Delete プロパティは「None」に設定する必要があります。
- **不要なソートボタンを削除します。** この機能が不要でないリストアプレット内のリストカラムからソートボタンを削除します。
- **ビュー内でのスクロールの必要性を軽減します。** 可能な限り常に、スクロールを必要としないビューを設計します（これにより得られる利点には、使い勝手の良さもあります）。
- **チューニングされた PDQ を用意します。** 大半のユーザー要件に対応するチューニングされた PDQ（定義済みクエリー）を用意します。これによって、ユーザーが好ましくない複雑なクエリーを作成してしまう可能性が減少します。また、適切なクエリーの作成に関するガイダンスをエンドユーザーに提供することもできます。
- **ビジネスサービスをキャッシュします。** ユーザーセッションで常にアクセス可能であることが必要なビジネスサービスをキャッシュします。このためには、適切な各ビジネスサービスオブジェクトの定義で Cache パラメータを TRUE に設定します。サービスはセッション単位でキャッシュされるので、ビジネスサービスのキャッシュはメモリに影響します。セッションで頻繁にアクセスされるビジネスサービスだけに、キャッシュ可能なマークを付けるようにしてください。
- **Count と Sum を実行する計算フィールドを避けます。** Counts と Sums を実行する計算フィールドの使用をできるだけ避けてください。そのようなフィールドがアクティブになっていると、パフォーマンスの低下の原因になります。

## 生成された SQL のパフォーマンスの問題に関する分析

パフォーマンスに関するトラブルシューティングは、反復して行われるプロセスです。設計時および開発時に、パフォーマンスに関係する問題を考慮する必要があります。MVG、ビジネスコンポーネントのソート条件と検索条件、結合、拡張テーブル、インデックスなど、潜在的に困難を伴う可能性のある部分の変更にご注意します。

次に、テスト環境内で実際のデータ量および配信量を使用して、ボトルネックを特定するためのアプリケーションのテストを行います。テスト作業の焦点を最も低速で、最も重要かつ最も複雑に設定されたビューに合わせます。

テスト時または本番時にパフォーマンスの問題が見つかった場合、次の手順は Siebel アプリケーションによって作成された SQL 文を分析することです。これは、パフォーマンス分析で使用可能な最も有用な診断ツールの 1 つです。

## Siebel Developer Web クライアント内での SQL スプールの指定

Siebel Tools 内で設定に変更を加えた後、実行時に Siebel アプリケーションによって生成される SQL をスプールします。これによって、設定に関連するパフォーマンスの問題をトラブルシューティングします。

生成された SQL を追跡ファイルにスプールするには、Siebel Developer Web クライアント (Siebel データベースに接続するクライアント) 内で、コマンドラインオプション `/s sql_trace_file` を使って Siebel アプリケーションを起動します。

Siebel Developer Web クライアントのインストールおよび実行については、使用しているオペレーティングシステム用の『Siebel インストールガイド』を参照してください。

この SQL 追跡ファイルには、現在のセッション時に生成されたすべての固有の SQL 文と、各 SQL 文の処理に費やされた時間が含まれます。セッションの終了後に、追跡ファイルをテキストエディタで開き、検査できます。セッションからスプールされた SQL を格納する単純なテキストファイルである SQL 追跡ファイルは、すべての新規セッションに上書きされます。

`/s sql_trace_file` オプションを指定するには、Siebel アプリケーションを起動する [スタート] メニューの項目またはデスクトップのショートカットのプロパティを変更します。次の例は、Siebel Developer Web クライアントを使って Siebel Call Center から生成された SQL をスプールするコマンドラインです。

```
"D:\Program Files\siebel\7.8\web client\bin\siebel.exe
/c D:\Program Files\siebel\7.8\web client\bin\enu\agent.cfg /s siebel_sql.txt"
```

パスを指定しない場合、`"D:\Program Files\siebel\7.8\web client\bin"` などの Siebel クライアントのルート bin ディレクトリ内に SQL 追跡ファイルが作成されます。

- SQL のスプールを行う目的で、Siebel eService などの顧客アプリケーションを実行するための Siebel Developer Web クライアントへのショートカットを作成できます。たとえば、このショートカットでアプリケーション設定ファイル `eservice.cfg` を参照できます。
- Object Manager SQL Log (ObjMgrSqlLog) パラメータを 4 に設定することにより、Application Object Manager (AOM) コンポーネントに対する SQL のスプールをコンポーネントレベルで有効にできます。詳しくは、『Siebel システム管理ガイド』を参照してください。
- また、Siebel オブジェクトインターフェイスを通じて SQL スプールをプログラマ的に開始および停止できます。このためには、Application オブジェクト上の `TraceOn` メソッドおよび `TraceOff` メソッドを使用します。これらの方法については、『Siebel Object Interfaces Reference』を参照してください。

## SQL 追跡ファイルを使用したパフォーマンスのトラブルシューティング

前述したように、設定済みの特定のビューなどで設定を変更した場合に、その変更に関連する SQL 追跡ファイルを生成できます。SQL 追跡ファイルの内容を分析することにより、想定されるパフォーマンスの問題を確認します。

SQL 追跡ファイルを分析する場合は、次の要因に注意する必要があります。

- SQL 文の数と複雑さ。
- SQL 文の実行時間。これは、SQL 実行時間と、その結果を返すためにかかった時間を加えた時間です。クライアント側の処理の時間は含まれません。
- 検索条件を指定する WHERE 句内の条件。
- ソート条件を指定する ORDER BY 句の条件。一般に、ソートするデータの量を減らすために、クエリーで最初に WHERE 句を使用してデータをフィルターすることをお勧めします。ユーザーのニーズに合わせたソート条件を適用することにより、追加のシステムリソースを必要とする、ユーザー固有のソート操作の可能性を軽減できます。
- 結合の使用。

**備考：** 同じ SQL 文が繰り返し実行される場合、Siebel アプリケーションでは最初のクエリーの文全体が表示されます。同じクエリーの以降の各繰り返しでは、バインド変数のみが表示されます。繰り返し実行されるクエリーは、そのクエリーで使用されているバインド変数セットにより認識できます。

SQL 文は、ハウスキーピングクエリーを含むすべてのクエリーで表示されます。システム処理で必要になるクエリーもあります。たとえば、権限を取得するためにユーザーのログインを検索したり、カレンダーで今日のアラームを特定したりする処理です。ピックリストに入力するための S\_LST\_OF\_VAL テーブルに対するクエリーが使用されることもあります。SQL 追跡ファイルには、ビューに入力するためのクエリーが含まれることもあります。このようなクエリーは、それがアクセスするテーブルに基づいて、容易に識別できる必要があります。

## SQL クエリープランを使用したパフォーマンスのトラブルシューティング

SQL 追跡ファイル内で問題のあるクエリーが見つかったら、Oracle の SQL\*Plus など、RDBMS で提供されているデータベースクエリーツールを使ってより詳細な情報を入手できます。

追跡ファイルからデータベースクエリーツールに SQL 文をコピーして貼り付けた後、Siebel データベースに対するクエリーを実行し、クエリープランを生成します。クエリープランは、実行したクエリーに関する各種統計の詳細なレポートです。SQL Anywhere データベースに対するクエリープランの生成例については、[178 ページの「クエリープランの取得例」](#)を参照してください。

クエリープランを使って、次のことをチェックします。

- インデックスの使用
- 一時テーブルの使用
- 順次のテーブルスキャンの使用

最後に、低速になる可能性のあるクエリーを特定するために、得られた結果と通常のアプリケーション（つまり、カスタム設定されていないアプリケーション）を比較します。

検索条件またはソート条件の変更や、ベーステーブル上での新しいインデックスの作成を行うことにより、多くのパフォーマンス面の問題を解決できます。

**注意：** 特別に訓練された Oracle 担当者のみが既存の Siebel インデックスを変更できます。この理由は、ユーザーインターフェイスからクエリーのパフォーマンスを向上させるためにインデックスを変更した結果として、Siebel EIM など他のモジュールのパフォーマンスに悪影響が及ばないようにするためです。詳しくは、183 ページの「ソートおよび検索時のデータベースインデックスの管理」を参照してください。

ビジネスコンポーネントの検索条件とソート条件のプロパティを変更する前に、潜在的なパフォーマンスへの影響を考慮します。SQL を追跡ファイルにスプールすることにより、アプリケーションで各アプレットを通じてビジネスコンポーネントへのクエリーが実行されるときのどのインデックスが使用されるかを分析できます。

本番データセットと同規模のデータセットに対してクエリープランを実行します。本番データベースに 200,000 レコードが含まれる場合に、30 レコードのテストデータセットに対するクエリーのパフォーマンスを分析しても、有効な結果は得られません。

次の要領で、検査するビューに優先順位を付けると便利です。

- **第 1 の優先順位：** 最も大きなパフォーマンスのボトルネックを持つことがわかっているビュー
- **第 2 の優先順位：** 最も頻繁にアクセスされるビュー
- **第 3 の優先順位：** 最も複雑に設定されたビュー（標準の Siebel アプリケーションと比較します）

評価のためのベンチマークとして、標準の Siebel アプリケーションと比較します。事前に選択されたビュー経由のルートに従って、標準の Siebel アプリケーションから追跡ファイルを取得することが役立つ場合が少なくありません。その後、できるだけ近いルートに従って、カスタム設定されたアプリケーションから別の追跡ファイルを取得します。この 2 つの追跡ファイルを比較し、前述のリストに示した箇条書き項目の相違に注意します。

**備考：** クエリープランの確認時には、各クエリーが適用されるビジネスオブジェクトを追跡します。S\_APP\_QUERY 文を探すことにより、新しいビジネスオブジェクトが開かれた場所がわかります。アクセスされたビジネスオブジェクトは、このクエリーの下にあるバインド変数の文によって表されます。

バインド変数は、返されるレコードを決定する値です。RDBMS は、同じ SQL 文が再使用される場合に、一般的には疑問符または一連の疑問符が出現する各場所で、バインド変数の値を SQL 文に置き換えます。たとえば、S\_APP\_QUERY 文の目的がビジネスオブジェクトを開くことにあるために、S\_APP\_QUERY 文内ではビジネスオブジェクトのバインド変数が使用されます。

問題が潜在することを示す次の兆候を監視します。

- 不要なフィールドにアクセスされます。特に、ユーザーインターフェイス内で公開されていないフィールド、計算フィールドで必要とされないフィールド、または詳細レコードに値を渡す目的で使用されるフィールドへのアクセスに注意します。
- 不要な結合が発生します。特に、アクセスされないテーブルでの発生に注意します。
- 同じテーブルに対して不要な複数の結合が発生します。これは、重複する結合または Multi Value Link (MVL) のオブジェクト定義の存在、または同じ外部キーを使った結合の発生を示唆している可能性があります。
- 次の名前に似た複数の短いクエリーが存在します。

```
...FROM
```

```
SIEBEL.S_ADDR_PER T1
```

短いクエリーが何度も現れる場合、一般的にはプライマリ結合を持たない MVG がリストアプレットでアクセスされていることを示唆します。システムは、マスタレコードごとに、その詳細レコードを取得するためのセカンダリクエリーを実行しています。セカンダリクエリーは、ログファイルに現れる短いクエリーです。セカンダリクエリーが現れた場合、通常はプライマリ結合の必要性を最もよく示す兆候といえます。

短いクエリーが 1 回だけ現れる場合も同じ状況を示唆していますが、ただしプライマリ結合を持たない MVG のアクセスはフォームアプレット内で行われています。いずれの場合でも、[190 ページの「プライマリ ID フィールドの使用によるパフォーマンスの向上」](#)で説明するように、プライマリ結合を使用することにより解決できます。

## クエリープランの取得例

次に、Siebel モバイル Web クライアントを使用してローカル SQL Anywhere データベースに対して実行する際に、クエリープランを取得する手順の例を示します。

### 追跡ファイル内で SQL 文のクエリープランを取得するには

- 1 Siebel クライアントインストールディレクトリ (Siebel Mobile) にある Interactive SQL (dbisqlc.exe) プログラムを実行します。
- 2 SQL 追跡ファイルの SQL 文を分析するには、その SQL 文をコピーして Interactive SQL プログラムの [コマンド] ペインに貼り付けます。
- 3 バインド変数の参照を対応するバインド変数値に置き換えます。
- 4 [実行] ボタンをクリックします。

ローカル SQL Anywhere データベースに対してクエリーが実行します。[統計] ペインに分析情報が表示されます。

## データベースデータに対する SQL クエリー

Siebel アプリケーションの基盤となるデータベースに対してクエリーを実行することにより、読み取り専用ベースで情報を取得できます。

**注意：** 更新クエリーを Siebel データベースに対して直接実行すべきではありません。すべてのデータ操作と再構成は、Siebel Tools または Siebel アプリケーションを通じて実行する必要があります。

## Siebel スクリプトに関する最善の方法

ここでは、Siebel eScript または Siebel VB を使用する Siebel スクリプト、またはスクリプトの代わりに宣言的な手段を使用するためのガイドラインを示します。

### Siebel スクリプトに代わる宣言的な手段の使用

顧客は、データの検証、データの変更に対する応答、宣言的な手段によって最適に解決されるその他の用途でスクリプトを使用することがよくあります。このためには、Siebel Tools を使ってプロパティを定義するか、またはビジネスサービスマソッドの呼び出しを指定します。

スクリプトは多くの場合に不要であり、その使用は最小限に抑えるか、回避する必要があります。これは、パフォーマンスの問題を招くほか、リスクおよび複雑さの増加、保守作業の増大を伴い、Siebel アプリケーションですでに使用可能な機能を重複させる場合があるためです。

たとえば、一般的な VB 式と比較演算子を使用できる Validation フィールドプロパティを使って、ユーザーインターフェイスまたは Siebel オブジェクトインターフェイスを通じて入力されたデータのフィールドの検証または文字列操作を実行できます。

Validation プロパティの式には、LoginId()、LoginName()、LookupValue()、ParentFieldValue()、PositionId()、PositionName()、Today() などのメソッドを含めることができます。

入力された個人名の先頭を大文字にする場合など、データ検証コンテキストによっては Force Case フィールドプロパティが役立つこともあります。

サポートされる式および演算子については、『Siebel Developer’s Reference』を参照してください。

MVL オブジェクト定義の Auto Primary プロパティを設定することでも、それ以外の場合にはスクリプトで可能になる処理を実行するために役立つときがあります。たとえば、ビジネス要件として MVG 内でプライマリレコード（プライマリ住所やプライマリ所有者など）として最初のレコードを割り当てる場合、Auto Primary に値 Default を設定します。

プライマリ ID フィールドの使用については、[190 ページの「プライマリ ID フィールドの使用によるパフォーマンスの向上」](#) および『Configuring Siebel Business Applications』を参照してください。

宣言的なメソッドとともに、検証ルールの適用される各フィールドに適切なデータを入力するようにユーザーに案内する、カスタマイズしたエラーメッセージを表示する場合などに、スクリプトを使用することもできます。

データの変更に対するカスタム応答などの機能は、スクリプトにより処理されることもありますが、宣言的な手段が最も適しています。次に、このようなメカニズムをいくつか示します。その多くは、組み合わせて使用できます。

- アプレット、ビジネスコンポーネント、フィールド、コントロール、リストカラム、およびその他のオブジェクトの定義（例：Required、Pre-Default、Post Default、Search Spec、Type Field、Type Value）のユーザープロパティ
- Siebel ワークフロー
- ステートモデル
- Siebel Personalization
- ランタイムイベント

- 名前付きメソッド
- ビジネスサービス
- 可視性の設定

スクリプトのガイドラインについては、『Configuring Siebel Business Applications』を参照してください。これらのトピックの多くについては、『Siebel Bookshelf』で説明されています。

## 最適なパフォーマンスのための Siebel スクリプトガイドライン

ここでは、Siebel eScript または Siebel VB による Siebel スクリプトの適切な使用についてガイドラインを示します。

このガイドラインやその他のガイドラインについては、次を参照してください。

- 『Siebel eScript Language Reference』
- 『Siebel VB Language Reference』
- 『Siebel Object Interfaces Reference』
- 『Configuring Siebel Business Applications』

次に、Siebel スクリプトを適切に使用するためのガイドラインをいくつか紹介します。

- **宣言的な代替手段を使用します。** 一般的に、機能要件の実現のためにスクリプトを使用する前に、他のあらゆる可能性を試みる必要があります。179 ページの「[Siebel スクリプトに代わる宣言的な手段の使用](#)」も参照してください。
- **フィールド検証などのシンプルなクライアント側機能ではブラウザスクリプトを使用します。** ブラウザスクリプトは、クライアント側での簡単な手続きロジック、たとえばフィールド検証の実行、ユーザーに対するブロックメッセージまたはアラートの表示などを実行する用途で最も適しています。このような用途のいくつか、特にフィールドの検証では、サーバーとのやり取りを削減できます。しかし、使用するブラウザスクリプトがより複雑になると、パフォーマンスが低下することがあります。

たとえば、Set/Get Profile 属性の呼び出しを使用したり、複数のビジネスサービスマソッドを起動したりすると、要求されるサーバーとのやり取りが増え、パフォーマンスの問題に結びつく可能性があります。メッセージを表示するスクリプトに機能を追加した場合も、同様の影響が及ぶ場合があります。

**備考:** Immediate Post Changes フィールドプロパティを設定した場合も、サーバーとのやり取りに同様の影響が生じます。このプロパティは、動的に更新する必要がある制約されたピックリストと計算フィールドのみで使用してください。

- **サーバービジネスサービスからブラウザスクリプトに大きな結果セットを返しません。** サーバースクリプトを起動するブラウザスクリプトは、簡単な値または単一のレコードを返すべきであり、大きな結果セットを返すべきではありません。
- **フィールドレベルまたはコントロールレベルイベントのスクリプトを最小限に抑えます。** フィールドレベルまたはコントロールレベルのイベントは、他のほとんどのタイプのイベントより発生頻度が高くなります。したがって、そのようなイベントからスクリプトを呼び出すと、スケーラビリティに大きく影響します。頻繁に発生するイベントに対するスクリプトの記述を避けるか、またはそのようなイベントのスクリプトを簡素化します。そのようなイベントの例としては、BusComp\_PreGetFieldValue()、WebApplet\_PreCanInvokeMethod()、WebApplet\_ShowControl() などがあります。

- **アプレットレベルおよびビジネスコンポーネントレベルのイベントで簡単なスクリプトを使用します。** Change Record イベントなどのアプレットコンポーネントまたはビジネスコンポーネントのイベントは発生頻度が高いので、それらに対して記述するスクリプトは非常に簡単にする必要があります。このようなイベントで複雑な処理または I/O 集約的な処理を実行すると、パフォーマンスに悪影響があります。
- **Siebel eScript スクリプトのデータをキャッシュします。** Siebel eScript スクリプトのさまざまな場所から同じ SQL ステートメントを実行すると、多数のスクリプト API 呼び出しと、不必要なビジネスコンポーネントクエリーが生成される場合があります。パフォーマンスへの影響を減らすために、呼び出し間でデータが変化しないと仮定して、スクリプト内で一部のデータをキャッシュできます。キャッシュするデータが過度に複雑であるか、またはデータ量が大きすぎる場合など、スクリプトレベルでデータのキャッシュを抑制する必要がある場合もあります。
- **変数を宣言します。** 必要に応じて、変数の宣言およびデータタイプの指定を行うことにより、使用メモリの削減とパフォーマンスの向上につながる場合があります。
- **作成したオブジェクトが不要になった場合にはすべて破棄します (Siebel eScript)。** 理論的には、Siebel eScript インタプリタがオブジェクトのクリーンアップを処理しますが、多数のオブジェクトインスタンス層を含む複雑なコードが存在する場合に、インタプリタが適切な時期にオブジェクトをリリースできないことがあります。オブジェクトを破棄またはリリースすることにより、サーバーのメモリなど、リソースへの影響を最小限に抑えるために役立ちます。

Siebel オブジェクトを作成したプロシージャ内で、そのオブジェクトを明示的に破棄する必要があります。Siebel eScript 内でオブジェクトを破棄するには、オブジェクトを NULL に設定するか、またはオブジェクトを含む変数を別の値に設定します。最善の方法は、作成時とは逆順にオブジェクトを破棄することです。つまり、親オブジェクトを破棄する前に、子オブジェクトを破棄します。

- **適切なメソッドを使ってスクリプトが定義されていることを確認します。** 正しいメソッドを使ってスクリプトを定義しないと、パフォーマンスに影響する可能性があります。たとえば、挿入または更新が完了したときにレコードレベルで特別なコードを実行する必要がある場合は、BusComp\_SetFieldValue() ではなく BusComp\_WriteRecord() によりスクリプトを呼び出します。その理由は、SetFieldValue イベントが WriteRecord イベントよりはるかに頻繁に発生するからです。特殊な呼び出しメソッドの使用を制限します。
  - **スクリプトを正しいビューに実装していることを確認します。** スクリプトを正しいビューに実装しない場合、パフォーマンスに大きな影響を与える可能性があります。このスクリプトが一般的なデータ操作、ナビゲーション要件、およびビジネス要件に基づいて設定内の適切な場所に実装されていることを確認します。
  - **冗長なリポトリオブジェクト設定を回避します。** 不要なオブジェクトの検証を実行しません。メソッドの呼び出しを実行するたびに、パフォーマンスコストが生じます。以下に、フィールドのアクティブ化に関連するこの問題を例示して説明します。
  - **ActivateField() メソッドの使用を控えます (Siebel eScript)。** 使用しないフィールドをアクティブ化しません。ActivateField() メソッドの使用を控えます。このメソッドを使用すると、クエリーによって取得される列数が増えるとともに、結合を含む複数のサブクエリーが生じる場合があります。このような処理では、大量のメモリが使用される可能性があり、アプリケーションのパフォーマンスが低下する恐れがあります。
- 不要なフィールドのアクティブ化 (すでにアクティブなフィールドのアクティブ化) を実行しません。メソッドの呼び出しを実行するたびに、パフォーマンスコストが生じます。
- システムフィールドは、デフォルトですでにアクティブ化されているので、アクティブ化しません。このようなフィールドには Created、Created By、Updated などがあります。
  - すでにアクティブであるその他のフィールドもアクティブ化しません。Siebel Tools 内の Force Active フィールドプロパティで、フィールドをアクティブ化する必要があるかどうかを確認します。

- **ExecuteQuery() メソッドの使用を控えます (Siebel eScript)**。ビジネスコンポーネントを実行するための ExecuteQuery() メソッドの呼び出しを削除することで、パフォーマンスが大きく向上する場合があります。スクリプト間で特定のビジネスコンポーネントレコードの値を共有する場合に、各スクリプト内で ExecuteQuery() を個別に起動するよりも、共有変数を使用する方がより優れた方法です。
- **NextRecord() メソッドではなく SetSearchSpec() メソッドを使用します (Siebel eScript)**。特定のレコードが見つかるまで取得済みメソッドのリストを検索する NextRecord() メソッドを使用するのではなく、特定のレコードを取得する SetSearchSpec() メソッドを使用することにより、パフォーマンスが向上する場合があります。
- **ForwardOnly カーソルモードを使用します (Siebel eScript)**。ForwardBackward が必要な場合を除いて、ExecuteQuery() で ForwardOnly カーソルモードを使用します。ForwardBackward を使用すると、大量のメモリが消費され、その結果としてアプリケーションのパフォーマンスが低下する場合があります。
- **適切なエラー処理を行います**。適切なエラー処理を行うことで、最適なパフォーマンスを維持するのに役立ちます。エラー処理は重要ですが、パフォーマンスコストも発生します。スクリプトのエラー処理機能の使い方に関するその他のガイドラインについては、Siebel SupportWeb の「Technical Note 514」を参照してください。
- **ネストしたクエリループを避けます**。ネストしたクエリループには多数のサブクエリが含まれる場合があります。パフォーマンスに大きな影響を与える可能性があります。この方法は、極力控えます。反復数を最小限に抑えるために、ネストしたクエリループを正しい順序で実装します。スクリプトの記述方法によっては、ネストしたクエリループが暗黙に起動される場合があることに注意してください。
- **this オブジェクト参照を使用します (Siebel eScript)**。この特殊なオブジェクト参照 *this* は、「現在のオブジェクト」の eScript での略称です。アクティブなビジネスオブジェクトおよびコンポーネントへの参照の代わりに、This オブジェクト参照を使用する必要があります。

たとえば、ビジネスコンポーネントのイベントハンドラでは、パフォーマンスに大きな影響を与える可能性がある ActiveBusComp () ではなく、*this* を使用する必要があります。次に、例を示します。

```
function BusComp_PreQuery()
{
  this.ActivateField("Account");
  this.ActivateField("Account Location");
  this.ClearToQuery();
  this.SetSortSpec( "Account(DSCENDING)," +
  " Account Location(DSCENDING)");
  this.ExecuteQuery();return (ContinueOperation);}
}
```

- **Switch 要素 (Siebel eScript) を使用します**。Switch 要素は、単一変数の値に基づいて、必要な選択肢のうちから任意の数を選択するようにプログラムに指示します。この要素を使用すると、ネストした一連の If 文を使用するよりも優れたパフォーマンスが得られ、その保守もより容易になります。
- **Select Case 要素 (Siebel VB) を使用します**。Select Case 要素は、単一変数の値に基づいて、必要な選択肢のうちから任意の数を選択するようにプログラムに指示します。この要素を使用すると、ネストした一連の If 文を使用するよりも優れたパフォーマンスが得られ、その他にも各種の利点があります。
- **カスタムスクリプトをテストします**。スクリプトが完全にテストされ、最適化がなされていること、ビジネスニーズに対応するために必要なレベルを超えて複雑ではないことを確認します。

## データオブジェクト層についての最善の方法

ここでは、最適なパフォーマンスを得るために、データオブジェクト層内で選択した要素を設定する最善の方法を説明します。

### マルチリンガル LOV クエリーとキャッシュのパフォーマンス

マルチリンガル変数リスト (MLOV) フィールドは、ビジネスコンポーネントレベルよりも下位で実装されます。有効なターゲットカラムを持ち、MLOV を参照するフィールドは、セッションの現在の言語設定に一致する表示値を返します。

表示時には、基礎となる言語非依存のコードが Siebel アプリケーション検索により対応する表示値に変換されます。しかし、検索およびソートでは、値テーブルのリスト (S\_LST\_OF\_VAL) に対するデータベースの結合が実行されます。S\_LST\_OF\_VAL テーブルに直接関連するすべての設定が Siebel アプリケーションの MLOV 機能に対して適合していることを確認します。

MLOV を含むビューを最初に表示するとき、MLOV を含むフィールドごとに S\_LST\_OF\_VAL テーブルに対して個別のクエリーが実行されます。このクエリーでは、MLOV のすべての表示値が取得され、その値がメモリ内の LOV キャッシュに書き込まれます。以降、同じセッション中に同じビューが表示される場合は、再度クエリーが実行されることなく、キャッシュから値が取得されます。

**備考：** 1 つ以上の MLOV フィールドを含むリストアプレット内のレコードを複数表示すると、メモリ使用量が増加し、パフォーマンスが低下する可能性があります。この問題は特に、特定の論理結果セットに対して複数のフェッチを実行する場合、つまりレコードをスクロールする場合に明確に現れます。また、この動作を自動化する目的でクライアント側のエクスポートが実行される場合、またはビジネスコンポーネント上で NextRecord メソッドが繰り返し起動される場合に常に、この問題が現れることがあります。一般的には、リストアプレット内での MLOV フィールドの使用を控えるか、または MLOV を含むリストアプレットからのクライアント側エクスポートを無効にすることをお勧めします。

MLOV の設定については、『Configuring Siebel Business Applications』および『Siebel Global Deployment Guide』を参照してください。

### ソートおよび検索時のデータベースインデックスの管理

「データベースインデックス」は、テーブルに関連付けられた RDBMS 内のデータ構造です。データベースインデックスにより、テーブル内の全レコードへの参照が提供されるため、すばやい検索とフィルターが可能になります。また、特定の順序にソートされることから、その順序でのソートが高速に行われます。Siebel Database Server は、インデックスを使ってクエリーの結果セットを効率的に取得およびソートします。

Siebel Data Model 内で提供されているインデックスは、標準の Siebel アプリケーションで最適なパフォーマンスが得られるようにチューニングされています。カスタムのソートまたはフィルターの要件を持つ新しいビジネスコンポーネントを追加する場合は、その要件に対応しながら、結果セットを効率的に提供できるデータベースインデックスが存在することを確認する必要があります。新しいインデックスを追加することが必要になる場合もあります。

インデックスを追加するには、オブジェクトタイプ Index および Index Column を使用します。Siebel Tools 内でインデックスを作成し、データベースの拡張を適用すると、その結果としてデータベース内にインデックスが追加されます。

**備考：** カスタムインデックスを追加しても、必ずしもパフォーマンスが向上するとは限りません。場合によっては、パフォーマンスが低下することもあります。インデックスの増分値は、主にデータの均質性と分布に依存します。

データが不均質である場合、大半またはすべての値が固有です（固有である行 ID 値など）。データがそれほど不均質でない場合、つまり反復値が多い場合（均質な場合）は、インデックスはそのコストと比較してあまり大きな利点を持ちません。

ブール値フィールドでは、一般的にインデックスの価値はほとんどありません。最も頻度の低い値に対するクエリーの実行時に、パフォーマンスの利点が得られる可能性があります。均質に分布するより頻度の高い値に対するクエリーの実行時には、利点はほとんど得られません。リスト値に制約されるフィールドなど、その他の均質なデータにも同様のガイドラインが適用されます。

インデックスを作成すると、一般的には SELECT 処理のパフォーマンスが向上します。ただし、Siebel EIM によって実行される場合など、バッチ UPDATE 処理および INSERT 処理のパフォーマンスが大幅に低下する可能性があります。

Siebel エキスパートサービスとあらゆるカスタムインデックスの要件について議論する必要があります。

## Sort Specification

ビジネスコンポーネント、ピックリスト、または定義済みクエリーの Sort Specification プロパティにより、クエリーで取得されるレコードがソートされます。またこのプロパティに基づいて、SQL の ORDER BY 句が生成されます。ソート条件で指定した順序に対応するインデックスが存在する必要があります。このようなインデックスが存在しない場合、RDBMS エンジンが一時テーブル内で結果セット全体を物理的にソートします。

インデックスには、すべてのフィールドのベースカラムを含める必要があります。また、ベースカラムは同じ順序で使用する必要があります。ソート条件で使用されているよりも多いカラムをインデックス内で指定できます。ただし、その逆、つまりソート条件で使用されているよりも少ないカラムをインデックス内で指定することはできません。

たとえば、Contact ビジネスコンポーネントのソート条件 Last Name、First Name が S\_CONTACT ベーステーブルの少なくとも 1 つのインデックスによりサポートされている場合を考えます。該当するインデックスの 1 つに S\_CONTACT\_U1 があります。S\_CONTACT\_U1 には、LAST\_NAME、FST\_NAME、MID\_NAME、PR\_DEPT\_OU\_ID、OWNER\_PER\_ID、および CONFLICT\_ID の各カラムがこの順序で含まれています。このような状況で担当者を First Name の順序に並べるソート条件が求められる場合は、カスタムインデックスを作成する必要があります。

結合カラムではインデックスを使用できないので、結合カラムに対してソートを実行しないでください。

## Search Specification

ビジネスコンポーネント、アプレット、リンク、またはピックリストの Search Specification プロパティに条件を指定すると、その条件に適合する行が基盤となるテーブルから選択的に取得されます。ここで指定する検索条件に基づいて、発行される SQL の WHERE 句が生成されます。この条件をサポートするインデックスが存在することが必要です。このようなインデックスが存在しない場合、RDBMS はクエリーによって返される行だけではなく、テーブル内のすべての行をスキャンする可能性があります。

このインデックスには、検索条件内のフィールドにより参照されているすべてのカラムを含める必要があります。

[取引先 (個人)] などの [営業員] ビューまたは組織のアクセス制御が実装されている場所で、ユーザーが交差テーブルに対して非正規化されたカラム (たとえば S\_ORG\_EXT 内の NAME および LOC) をクエリーまたはソートした場合、おそらく十分なパフォーマンスが得られます。Siebel アプリケーションでは、ベーステーブル内のレコードの可視性を決定するために交差テーブルを使用します。また、交差テーブルにインデックスを作成してパフォーマンスを向上させることができます。

詳しくは、[185 ページの「標準カラムの再使用」](#)を参照してください。

**備考:** 交差テーブルに対して非正規化されていないカラムがクエリーまたはソートに含まれる場合、インデックスが使用されないためにパフォーマンスが低下します。

## 標準カラムの再使用

アプリケーションのアーキテクチャとデータモデルには、最適なパフォーマンスを得るためのチューニングがなされています。この最適化は、適切なインデックスの使用、データのキャッシュ、および効率的な SQL の生成に加えて、特定のテーブル上のカラムを非正規化することで実現できます。非正規化したカラムのインデックスを作成することによって、アプリケーションで元のテーブルのカラムではなく、非正規化したカラムを使って検索処理またはソート処理を行うことにより、複雑な SQL 文のパフォーマンスを向上させることができます。

**備考:** 既存のフィールドの再マップ、特に User Key カラムに基づくフィールドを同じテーブル内の他のカラムに再マップしないでください。

**注意:** Siebel エキスパートサービスのサポートを受けることなく、カスタムの非正規化カラムを使用しないでください。非正規化カラムでは、インデックスをマスターテーブルまたは詳細テーブルではなく交差テーブルに直接配置できるので、パフォーマンスを向上させることが可能です。しかし、非正規化カラムが誤って設定されている場合は、非正規化カラムのデータとそのソースの同期が取れなくなる可能性があります。これにより、ソートの不整合からデータの破損に至るまで、さまざまな問題が発生する可能性があります。

## 例：S\_ORG\_EXT テーブル内の NAME および LOC の再使用

S\_ORG\_EXT テーブルのカラム NAME および LOC が S\_ACCNT\_POSTN テーブルの ACCNT\_NAME および ACCNT\_LOC に非正規化されています。

Visibility Applet Type プロパティに Sales Rep が設定されたビュー内で取引先を名前および場所でソートする場合、Siebel アプリケーションは S\_ACCNT\_POSTN テーブルの非正規化カラム ACCNT\_NAME および ACCNT\_LOC を使用します。これによって、インデックスの使用が可能になります。

取引先の名前と場所が拡張カラムに保存されていれば（たとえば X\_NAME と X\_LOC）、NAME と LOC の代わりに拡張カラムを使ってソートすることが必要になっていたはずですが、これらの拡張カラムのインデックスが作成されていたとしても、そのインデックスが S\_ACCNT\_POSTN 上ではなく S\_ORG\_EXT 上にあるために、アプリケーションが既存のインデックスを使って必要な結合の作成とデータのソートを行うことができません。したがって、パフォーマンスは大幅に低下することになります。

## 取引先（個人）ビューのクエリープラン

次の第 1 の SQL 文は、標準の [取引先（個人）] ビューで生成されます。クエリープランを見ると、データベースがこの文を実行するために多数のインデックスを使用することがわかります。

```
SELECT
    T1.LAST_UPD_BY,
    T1.ROW_ID,
    T1.CONFLICT_ID,
    .
    .
    .
    T10.PR_EMP_ID,
    T2.DUNS_NUM,
    T2.HIST_SLS_EXCH_DT,
    T2.ASGN_USR_EXCLD_FLG,
    T2.PTNTL_SLS_CURCY_CD,
    T2.PAR_OU_ID
FROM
    SIEBEL.S_PARTY T1
        INNER JOIN SIEBEL.S_ORG_EXT T2 ON T1.ROW_ID = T2.PAR_ROW_ID
        INNER JOIN SIEBEL.S_ACCNT_POSTN T3 ON (T3.POSITION_ID = ?, 0.05)
    AND T2.ROW_ID = T3.OU_EXT_ID
        INNER JOIN SIEBEL.S_PARTY T4 ON (T4.ROW_ID = T3.POSITION_ID, 0.05)
        LEFT OUTER JOIN SIEBEL.S_PRI_LST T5 ON T2.CURR_PRI_LST_ID = T5.ROW_ID
        LEFT OUTER JOIN SIEBEL.S_INVLOC T6 ON T2.PR_FULFL_INVLOC_ID =
T6.ROW_ID
        LEFT OUTER JOIN SIEBEL.S_ORG_EXT T7 ON T2.PAR_OU_ID = T7.PAR_ROW_ID
```

```

LEFT OUTER JOIN SIEBEL.S_ORG_EXT_SS T8 ON T1.ROW_ID = T8.PAR_ROW_ID
LEFT OUTER JOIN SIEBEL.S_INT_INSTANCE T9 ON T8.OWN_INST_ID =
T9.ROW_ID
LEFT OUTER JOIN SIEBEL.S_POSTN T10 ON T2.PR_POSTN_ID = T10.PAR_ROW_ID
LEFT OUTER JOIN SIEBEL.S_USER T11 ON T10.PR_EMP_ID = T11.PAR_ROW_ID
LEFT OUTER JOIN SIEBEL.S_ADDR_ORG T12 ON T2.PR_ADDR_ID = T12.ROW_ID
LEFT OUTER JOIN SIEBEL.S_INDUST T13 ON T2.PR_INDUST_ID = T13.ROW_ID
LEFT OUTER JOIN SIEBEL.S_ASGN_GRP T14 ON T2.PR_TERR_ID = T14.ROW_ID
LEFT OUTER JOIN SIEBEL.S_POSTN T15 ON T3.POSITION_ID = T15.PAR_ROW_ID
LEFT OUTER JOIN SIEBEL.S_USER T16 ON T15.PR_EMP_ID = T16.PAR_ROW_ID
LEFT OUTER JOIN SIEBEL.S_ORG_SYN T17 ON T2.PR_SYN_ID = T17.ROW_ID
LEFT OUTER JOIN SIEBEL.S_ORG_BU T18 ON T2.BU_ID = T18.BU_ID AND
T2.ROW_ID = T18.ORG_ID
LEFT OUTER JOIN SIEBEL.S_PARTY T19 ON T18.BU_ID = T19.ROW_ID
LEFT OUTER JOIN SIEBEL.S_ORG_EXT T20 ON T18.BU_ID = T20.PAR_ROW_ID
WHERE
((T2.INT_ORG_FLG != 'Y' OR T2.PRTNR_FLG != 'N') AND (T3.ACCNT_NAME >= ?))
ORDER BY
T3.POSITION_ID, T3.ACCNT_NAME

```

クエリープラン：

T3(S\_ACCNT\_POSTN\_M1), T2(S\_ORG\_EXT\_P1), T1(S\_PARTY\_P1), T15(S\_POSTN\_U2), T10(S\_POSTN\_U2), T4(S\_PARTY\_P1), T12(S\_ADDR\_ORD\_P1), T13(S\_INDUST\_P1), T7(S\_ORG\_EXT\_U3), T16(S\_USER\_U2), T11(S\_USER\_U2), T17(S\_ORG\_SYN\_P1), T6(S\_INVLOC\_P1), T5(S\_PRI\_LST\_P1), T14(S\_ASGN\_GRP\_P1), T18(S\_ORG\_BU\_U1), T19(S\_PARTY\_P1), T20(S\_ORG\_EXT\_U3), T8(S\_ORG\_EXT\_SS\_U1), T9(se)

## 取引先（個人）ビューのクエリープラン - ORDER BY 句の相違

[取引先（個人）] で生成される次の第 2 の SQL 文では、その ORDER BY 句が異なります。S\_ORG\_EXT のカラム NAME および LOC のインデックスが作成されていても、データベースではこのインデックスを使用できません。このため、一時テーブルを使用する場合と比較して、パフォーマンスが低下します。ORDER BY 句でカラム NAME および LOC の代わりにカラム X\_NAME および X\_LOC を使用する場合も、同じ状態が生じます。

次の例では、前の例のクエリープランと異なる ORDER BY 句を示します。

```

WHERE
((T2.INT_ORG_FLG != 'Y' OR T2.PRTNR_FLG != 'N') AND
(T3.ACCNT_NAME >= ?))
ORDER BY
T3.ACCNT_NAME, T3.POSITION_ID

```

クエリープラン：TEMPORARY TABLE

T3(S\_ACCNT\_POSTN\_M1), T2(S\_ORG\_EXT\_P1), T1(S\_PARTY\_P1), T15(S\_POSTN\_U2), T10(S\_POSTN\_U2), T4(S\_PARTY\_P1), T12(S\_ADDR\_ORD\_P1), T13(S\_INDUST\_P1), T7(S\_ORG\_EXT\_U3), T16(S\_USER\_U2), T11(S\_USER\_U2), T17(S\_ORG\_SYN\_P1), T6(S\_INVLOC\_P1), T5(S\_PRI\_LST\_P1), T14(S\_ASGN\_GRP\_P1), T18(S\_ORG\_BU\_U1), T19(S\_PARTY\_P1), T20(S\_ORG\_EXT\_U3), T8(S\_ORG\_EXT\_SS\_U1), T9(se)

## ビジネスオブジェクト層についての最善の方法

ここでは、最適なパフォーマンスを得るために、ビジネスオブジェクト層内で選択した要素を設定する最善の方法を説明します。

### Cache Data プロパティによるビジネスコンポーネントのパフォーマンスの向上

AOM 上で同じユーザーセッションにより使用されるビジネスコンポーネントの内容をキャッシュするには、ビジネスコンポーネントの Cache Data プロパティを TRUE に設定します。

反復するクエリーの対象になる可能性があり、ユーザーセッション中に変更されることがまれな半静的なデータの場合に、Cache Data を TRUE に設定することが適しています。

ビジネスコンポーネントによっては、デフォルトで Cache Data が TRUE に設定されます。これは、PickList Generic や Internal Product などのビジネスコンポーネントで行われる動作です（190 ページの「プロパティの使用によるピックリストのパフォーマンスの向上」を参照）。

ユーザーセッション内で変化する可能性があるトランザクションデータのビジネスコンポーネントでは、キャッシュデータを FALSE にする必要があります。

### アクティブなフィールドの数の制限

ビジネスコンポーネントに基づくアプレットを含むビューにユーザーが移動する場合など、ビジネスコンポーネントのインスタンスの作成時には、ビジネスコンポーネントごとにフィールドオブジェクト定義のインスタンスが作成されます。このようにしてインスタンスを作成されたフィールドはすべて、対応するリストカラム、その他のフィールドコントロールを持つユーザーインターフェイスにフィールドが表示されていなくても、Siebel データベースに対して発行される生成済み SQL の SELECT 文に含まれます。

インスタンスの作成されるフィールドセットには、Force Active プロパティに TRUE が設定されたフィールドも含まれます。Force Active 設定に TRUE が設定されている場合、現在のアプレットにフィールドが表示されないときでも、ビジネスコンポーネントへのアクセス時に毎回、フィールドデータを取得する必要があることをシステムに指示します。これによって、毎回 SQL クエリーにこのフィールドが追加されることとなります。

Force Active を TRUE に設定した場合、それに関連したパフォーマンスコストが発生します。Force Active の設定は、MVL または結合に基づくフィールドの場合に、パフォーマンスに大きな影響を与えます。これは、Siebel アプリケーションがこのようなカラムのデータを取得するために SQL クエリー内で関係を作成する必要があるためです。

ほとんどの場合に、Force Active プロパティは必要ありません。一般的に、厳密に必要な場合を除いて、Force Active を TRUE に設定しないでください。

生成されるクエリーにフィールドを含める必要がある場合で、そのフィールドがユーザーインターフェイスに表示されないときのみ、Force Active を使用します。

## 計算フィールドを使用するためのガイドライン

計算フィールドを使用することにより、テーブルに直接保存されないデータをユーザーインターフェイス内で簡単にアクセスおよび表示できます。ただし、計算フィールドには、関連コストが存在します。したがって、それぞれの要件を満たすために適切に使用し、誤用を避けることが大切です。

ビジネスコンポーネントに計算フィールドの値に関するクエリーが実行されると常に、その計算フィールドが評価されます。計算フィールドの大量使用、または特定のコンテキスト内での計算フィールドの使用がパフォーマンスに影響する場合があります。次に、ガイドラインをいくつか示します。

- 計算フィールドの使用を控えます。計算フィールドの使用について、有効なビジネスケースが存在することを確認します。
- 計算フィールドに定義する式をできる限り簡素にします。
- MVG ビジネスコンポーネントの詳細レコードなどで、Sum、Count、Min、または Max の計算を実行する計算フィールドの使用をできる限り控えます。特に、リストアプレットや [追加情報] フォームアプレット内でこのようなフィールドを使用しないようにします。このような式の使用コストは、詳細レコードの数に大きく依存する場合があります。

データの合計値を計算する場合は常に、パフォーマンスへの影響が生じます。合計値を計算するレコードの数を制限することが大切です。たとえば、見積りレポートまたは経費レポート内で明細項目の合計値を計算しても、リソースはそれほど消費されません。しかし、すべての商談で予想される売上の合計値を計算すると、リソースはかなり消費されます。

すべての商談で予想される売上の合計値の計算は、チャートの作成時に実行されます。しかし、チャートの作成が頻繁に行われることはあまりありません。[商談] リストビューは、日常的な検索やデータ入力で頻繁にアクセスされます。

**注意:** リストカラム内に `sum([MVfield])` を含めないでください。リストカラム内に `sum([MVfield])` を含めると、リスト内のレコードごとに別個のクエリーの実行が必要になり、大きなパフォーマンスの問題を引き起こします。

- 現在の言語に応じて異なる値を提供する複雑な式で計算フィールドを定義しないようにします。
- 計算フィールドで、別のフィールドの値を直接コピーすることは避けます。
- 特に、基盤となる RDBMS でサポートされていない関数を計算フィールドで使用する場合は、検索条件に計算フィールドを含めないようにします。
  - この関数をサポートする RDBMS は、計算を効率的に実行するアルゴリズムを持ち、計算値を結果セットとともに返します。ただし、EXISTS、MAX、Count などの関数が含まれている場合は、複数のサブクエリーが実行され、パフォーマンスに影響する可能性があります。
  - 関数が RDBMS 内でサポートされない場合、Siebel アプリケーションが適切な計算を実行するために結果セット全体の再スキャンが必要になることがあり、したがってクエリー結果の取得にかかる時間が大幅に増加する可能性があります。

RDBMS で関数がサポートされる場合は結果の返却前に計算を済ませることができるのに対し、RDBMS で関数がサポートされない場合は Application Object Manager またはクライアントによりメモリ内で計算を実行することが必要になります。

**備考:** RDBMS レベルで計算フィールドがサポートされる場合でも、検索条件をサポートするインデックスが存在しないなど（たとえば LIKE 関数を使用する場合など）、検索条件で計算フィールドを使用することがパフォーマンスの低下につながる理由はその他にも存在します。183 ページの「ソートおよび検索時のデータベースインデックスの管理」を参照してください。

## プロパティの使用によるピックリストのパフォーマンスの向上

同じユーザーセッションにより使用される特定ピックリストの内容をキャッシュするには、PickList Generic ビジネスコンポーネントの Cache Data プロパティを TRUE に設定します。デフォルトでは、このプロパティは TRUE になっています。

**備考：** PickList Generic の表示 LOV データに基づくピックリストは、おそらくユーザーセッション中に変化することではなく、したがってキャッシュ処理に適しています。その他のビジネスコンポーネントの表示データに基づくピックリストは、ユーザーセッション中に変化する可能性があり、したがって一般的にはキャッシュ処理に適していません。

また、適切な各ピックリストオブジェクトの定義の Long List プロパティを TRUE に設定します。Long List を TRUE に設定すると、現在のピックリストレコード上でフォーカスが維持されず、したがって多数のレコードを含むピックリストでそのパフォーマンスが向上します。Long List のデフォルトの設定は、ピックリストのオブジェクト定義ごとに異なります。

## プライマリ ID フィールドの使用によるパフォーマンスの向上

プライマリ ID フィールドを使用しないで設定した MVG では、各親レコードと各子レコードセットの表示時に、個別のクエリーが必要になります。たとえば、1 レコードあたり 2 つの MVG を含む 10 レコードを表示するリストアプレットでは、アプレットに入力するために計 21 回のクエリーが必要になります。つまり、親レコードに入力するために 1 回のクエリー、さらに MVG に入力するために 20 回のクエリー（親レコード 1 つあたりに 2 回）が必要です。この場合に実行されるクエリーの回数は、実際に必要とされる回数をはるかに上回っています。

マスタービジネスコンポーネント上で Primary ID フィールドを設定することにより、不要なクエリーを回避できます。プライマリ ID フィールドは、詳細ビジネスコンポーネント内の親レコードから 1 つのプライマリ子レコードへの外部キーとして使用できます。これにより、アプリケーションが SQL 結合を使って 1 回のクエリーを実行することで、アプレット内の親レコードおよびプライマリ子レコードの値を表示できます。つまり、ユーザーが MVG を開いてすべての子レコードのリストを表示するまで、MVG に対する追加クエリーの実行が延期されます。

リストアプレットは一般的に、多数のレコードにアクセスし、その各レコードには 1 つ以上の MVG が関連付けられていることが考えられるため、プライマリ ID フィールドの使用により最も大きなパフォーマンス面の利点を受けます。プライマリ ID フィールドにより、親レコードごとに各 MVG に対してクエリーを実行する必要がなくなります。

またフォームアプレットでは、一度に 1 つの親レコードのみにアクセスする場合でも、Primary ID フィールドの恩恵を受けることができます。Primary ID フィールドにより、アプリケーションではフォームアプレット上の各 MVG について複数回のクエリーを実行しなくても、表示する新しい親レコードごとに 1 回のクエリーを発行するだけで済みます。これによって、ユーザーがレコード間を移動するときのパフォーマンスの向上が期待できます。

状況によっては、プライマリ ID フィールドを設定することが望ましくなかったり、不可能であったりする場合があります。

- Microsoft SQL Server を使用している場合は、プライマリ結合の作成時に Microsoft ソフトウェアでは禁止されている二重外部結合が生じます。
- マルチバリューフィールドの唯一の用途が、詳細レコードの値を合計する場合。

プライマリ ID フィールドを設定する方法については、『Configuring Siebel Business Applications』を参照してください。

## Check No Match プロパティによるパフォーマンスへの影響

ほとんどの場合、プライマリ ID フィールドの実装のために使用される Multi Value Link オブジェクト定義の Check No Match プロパティは FALSE に設定する必要があります。Check No Match プロパティを TRUE に設定すると、特に大半の親レコードが MVG 内で子レコードを定義していない条件では、パフォーマンスに悪影響を与える可能性があります。

Check No Match プロパティは、プライマリ結合を通じて子レコードが見つからない場合に、MVG に入力するために別個のクエリーを使用する必要があるかどうかを定義します。

- Check No Match を FALSE に設定した場合、アプリケーションは次のことを行います。
  - 親レコードのプライマリ ID フィールドが無効であるかまたは NULL 値を持つ場合、MVG 内に子レコードが存在するかどうかを特定するためのセカンダリクエリーが実行されます。子レコードが存在しない場合、プライマリ ID フィールドには値 *NoMatchRowId* が設定されます。
  - 親レコードのプライマリ ID フィールドの値が *NoMatchRowId* である場合には、MVG 内に子レコードがないことを *NoMatchRowId* が示しているため、アプリケーションはセカンダリクエリーを実行しません。このような余分な SQL クエリーの実行を回避することでパフォーマンスが向上します。  
**備考：** *NoMatchRowId* は永続的な設定ではありません。プライマリ ID フィールドは *NoMatchRowId* に設定された後でも変更できます。
- Check No Match を TRUE に設定した場合は、プライマリ結合でプライマリ子レコードが見つからなかった親レコードごとに、別個の SQL クエリーが実行されます。これにより、子レコードが存在しない場合を除いて、マルチバリューフィールドの表示は空欄にはなりません。しかし、SQL クエリーが余分に実行されるため、パフォーマンスは低下します。

次の場合に、Check No Match プロパティを TRUE に設定することが適切です。

- マルチバリューグループで MVG を経由しなくてもレコードを追加できる場合。たとえば取引先住所は Account ビジネスコンポーネントではなく Contact ビジネスコンポーネントの Business Address マルチバリューグループを通じて実際に挿入できます。
- Siebel EIM を通じて詳細ビジネスコンポーネントにレコードを追加できる場合。

Multi Value Link オブジェクト定義の設定については、『Configuring Siebel Business Applications』を参照してください。

## ユーザーインターフェイスオブジェクト層についての最善の方法

ここでは、最適なパフォーマンスを得るために、ユーザーインターフェイスオブジェクト層内で選択した要素を設定する最善の方法を説明します。

### グリッドレイアウトに関連するパフォーマンスの問題の解決

グリッドレイアウト機能は、開発者が Siebel ビューで効果的に使用できるフォームアプレットを作成することを可能にします。ただし、アプレット設計の選択によっては、パフォーマンスに悪影響が及ぶ場合があります。

一般的にこの種のパフォーマンスの問題は、ラベルやフィールドなどのユーザーインターフェイスコントロールの配置に関連し、空白セルを含めたグリッドベースのフォームアプレット内の総セル数とその要因になります。パフォーマンスへの影響は、ユーザーインターフェイス要素の数、アプレットのサイズ、その他の要因に依存します。

次の作業を行うことで、ユーザーインターフェイスのパフォーマンスを最適化できます。

- スタックセットのラベルまたはフィールドを同じ幅にします。これによって、隣接する必要な空白セルの数を削減できます。
- ラベルのスタックセットの配置を揃えます。
- ラベルを隣接するフィールドと同じ高さにします。
- 不要と思われる水平方向または垂直方向の空白セルを除去します。

**備考：** 想定される使い方に配慮して、全オプションの尺度を設定します。一般的に、ビューレイアウトで適切なスペースを空けると使い勝手が向上します。

グリッドレイアウト機能の使用については、『Configuring Siebel Business Applications』を参照してください。

## アプレットトグル使用時のパフォーマンスの維持

アプレットトグル機能は、異なるビジネスコンポーネントに基づく複数のアプレットがビュー内の同じ場所を占有する場合に役立ちます。同時に表示されるアプレットは、親アプレット内のフィールド値（動的トグル）またはユーザーの選択（静的トグル）に左右されます。

動的トグルアプレットが同じビジネスコンポーネントに基づくのに対し、静的トグルアプレットは異なるビジネスコンポーネントに基づく場合もあります。

一般に、Siebel アプリケーションのアプレットトグル、特に動的トグルを設定する場合、1 アプレットあたりのアプレットトグルとフィールドの数を最小にすることにより、ユーザーアプリケーションセッションのメモリおよび CPU の使用量を削減できます。

アプレットトグル、特に動的トグルを使用する場合は、考えられるパフォーマンスへの影響に注意することが大切です。

- ユーザーが動的アプレットトグルの親アプレット内でレコードを選択すると、その全アプレットトグルのビジネスコンポーネントとフィールドのインスタンスが作成され、メモリ内でキャッシュされます。また、そのすべてのフィールドに対するクエリーが実行されます。

このクエリーは、ユーザーが親レコード内の関連フィールド値を変更したときに表示される可能性がある他のアプレットトグルへの入力のために使用されます。ただし、ユーザーが親アプレット内で異なるレコードを選択するたびに、トグルビジネスコンポーネント内のすべてのフィールドが要求されることとなります。

また、動的アプレットトグルを含むビューでは、ビューレイアウトがキャッシュされないことに注意してください。

- ユーザーが静的アプレットトグルを含むビューに移動すると、デフォルトで表示されるアプレットのビジネスコンポーネントとフィールドのインスタンスが作成され、メモリ内にキャッシュされます。また、そのフィールドに対するクエリーが実行されます。ユーザーがアプレットトグル内の他のアプレットに移動すると、そのアプレットのビジネスコンポーネントのインスタンス作成およびキャッシュが行われ、別のクエリーが実行されます。

どの場合でも、ユーザーが別の画面に移動するまで、キャッシュされたオブジェクトがメモリ内で保持されます。



# 13 パフォーマンスのためのオペレーティングシステムのチューニング

この章では、Siebel Enterprise のインストールにおけるパフォーマンスとスケーラビリティの向上を目的としたチューニング手順について説明します。この章では、次のトピックについて説明します。

- 196 ページの「Siebel Server のパフォーマンス向上のための Microsoft Windows のチューニング」
- 197 ページの「すべての UNIX プラットフォーム用の Siebel Server のチューニング」
- 198 ページの「すべての UNIX プラットフォーム用の Siebel Web Server Extension のチューニング」
- 199 ページの「AIX 用 Siebel Business Applications のチューニング」
- 204 ページの「Solaris 用の Siebel Business Applications のチューニング」
- 208 ページの「HP-UX 用の Siebel Business Applications のチューニング」

この章の手順を実行する前に、使用しているオペレーティングシステム用の『Siebel インストールガイド』に説明されている、Siebel Gateway Name Server および Siebel Server のインストールに関する最小限必要な設定手順を完了しておく必要があります。

チューニングと監視の詳細については、『Siebel システム監視および診断ガイド』および『Siebel システム管理ガイド』を参照してください。

**備考：** この付録に記載されている設定は、Siebel Industry Applications のための Siebel Call Center などの Siebel 標準アプリケーションを使用する制御されたラボ環境に基づいたものです。これらの設定を使用して得られる各サイトのパフォーマンスの向上の度合いは、個々の実装に依存します。サポートされているオペレーティングシステムプラットフォームに対するその他のチューニング推奨事項については、各ベンダーに問い合わせてください。

# Siebel Server のパフォーマンス向上のための Microsoft Windows のチューニング

ここでは、Siebel アプリケーションのパフォーマンスを最適化するように Microsoft Windows オペレーティングシステムを設定する方法について説明します。

## データスループットの最大化

データスループットの設定を [ファイル共有のデータスループットを最大にする] (デフォルト) から [ネットワークアプリケーションのデータスループットを最大にする] に変更すると、次のような効果があります。

- 対称マルチプロセッシング (SMP) のスケーラビリティの向上
- ネットワークパフォーマンスの向上
- Siebel アプリケーションの物理メモリの割り当ての増加

この設定について詳しくは、Microsoft のマニュアルを参照してください。

## 4GT RAM チューニング機能をオンにする

プロセスあたりのアドレス制限を 2 GB から 3 GB に拡張できます。これによって、オペレーティングシステムが使用できる物理 RAM の容量が 2 GB から 1 GB に減少します。この差分 (1 GB) はアプリケーションに割り当てられます。この機能は、4GT RAM チューニングと呼ばれます。この設定方法については、Microsoft のマニュアルを参照してください。

**備考：** 各 Siebel プロセス (Application Object Manager) が使用できる RAM は、2 GB までです。

## すべての UNIX プラットフォーム用の Siebel Server のチューニング

サポートされている UNIX プラットフォームで実行されるすべての Siebel Server マシンについては、この項で説明されている環境変数を設定すると、サーバーリソースを適切に管理し、CPU 使用率の制限内で正常に運用することができます。

### Siebel Assert 作成用の環境変数

Siebel Server マシンまたは Web サーバーマシンでは、環境変数 `SIEBEL_ASSERT_MODE` によって assert ファイルを作成するかどうかを指定します。デフォルト値の 0 に設定すると、assert ファイルの作成が無効になるので、ディスク容量を節約し、パフォーマンスを向上できます。

この変数は、システム診断プログラムを実行している場合のみゼロ以外の値に設定する必要があります。また、設定する場合は、必ず Siebel テクニカルサービスにご相談ください。

この変数については、『Siebel システム監視および診断ガイド』を参照してください。

### オペレーティングシステムのリソース制限用の環境変数

Siebel Server の該当するプロファイルの変数を定義する次のいずれかの方法を使用して、環境変数 `SIEBEL_OSD_MAXLIMITS` を設定します。

■ C シェル：

```
setenv SIEBEL_OSD_MAXLIMITS 1
```

■ Korn シェルまたは Bourne シェル：

```
SIEBEL_OSD_MAXLIMITS=1;export SIEBEL_OSD_MAXLIMITS
```

この変数を 1 に設定すると、オペレーティングシステムのリソースの最大値が適用されます。リソースには、`coredumpsize`、`cputime`、`filesize`、`descriptors`、`maxmemory` などが含まれます。

## オペレーティングシステム Latch の環境変数

Siebel Server のタスク総数が 500 を超える場合、負荷を管理するためにここで説明されている環境変数を設定する必要があります。SIEBEL\_OSD\_NLATCH は名前付きラッチを制御し、SIEBEL\_OSD\_LATCH は名前なしラッチを制御します。ラッチは、mutexes (mutual exclusion object) に似ていますが、プロセス間の通信に使用されます。

SIEBEL\_OSD\_NLATCH と SIEBEL\_OSD\_LATCH が定義されていない場合、値はそれぞれ 5000 と 1000 になります。これらの値で十分な場合、または Siebel Server のタスク総数が 500 未満の場合、これらの変数を設定する必要はありません。

**備考：** これらの変数を変更する前に、`stop_server` コマンドを使用して Siebel Server を停止し、`cleansync` ユーティリティを実行します。このユーティリティについては、Siebel SupportWeb を参照してください。

次の式に基づいて、Siebel Server マシンの SIEBEL\_OSD\_NLATCH および SIEBEL\_OSD\_LATCH を設定します (Siebel Server の該当するプロファイルで変数を定義します)。

■  $SIEBEL\_OSD\_NLATCH = 7 \times (\text{すべてのコンポーネントに対する累積の MaxTasks}) + 1000$

■  $SIEBEL\_OSD\_LATCH = 1.2 \times (\text{すべてのコンポーネントに対する累積の MaxTasks})$

同じ Siebel Server で 2 つのマルチスレッド化されたサーバーコンポーネント (SCCObjMgr\_enu と WfProcMgr) を有効にしたと仮定します。SCCObjMgr\_enu に対して MaxTasks = 500、WfProcMgr に対して MaxTasks = 100 を指定します。この例では、パラメータ値を次のように設定します。

■  $SIEBEL\_OSD\_NLATCH = 5200 = 7 \times [500 + 100] + 1000$

■  $SIEBEL\_OSD\_LATCH = 720 = 1.2 \times [500 + 100]$

# すべての UNIX プラットフォーム用の Siebel Web Server Extension のチューニング

Siebel アプリケーションを UNIX プラットフォームで実行できるように、Siebel Web Server Extension (SWSE) をチューニングする必要があります。

## UNIX プラットフォーム用に SWSE をチューニングするには

- 1 SWSE のインストールディレクトリで、bin サブディレクトリに移動します。
- 2 vi などのテキストエディタを使用し、編集用に `eapps.cfg` ファイルを開きます。
- 3 適切な AnonUserName ユーザー名とパスワードを設定します。これは、ユーザー認証戦略によって異なります。詳しくは、『Siebel セキュリティガイド』を参照してください。
- 4 `GuestSessionTimeout` を 60 に設定します。

**備考：** この設定は、ユーザーがログインせずに参照するアプリケーションシナリオに適しています。

- 5 Web サーバーを再起動してこの設定を有効にします。

# AIX 用 Siebel Business Applications のチューニング

ここでは、AIX で Siebel アプリケーションを実行する上での Web サーバー、オペレーティングシステム、および Siebel Enterprise コンポーネントを設定およびチューニングする方法について説明します。次のトピックが含まれています。

- 199 ページの「AIX 用の IBM HTTP Server のチューニング」
- 201 ページの「AIX 用の Siebel Server のチューニング」
- 203 ページの「AIX 用のカーネル設定のチューニング」

## AIX 用の IBM HTTP Server のチューニング

ここでは、IBM HTTP Server (IHS) Web サーバーのスケラビリティとパフォーマンスを最適化する環境変数の推奨値について説明します。これらの設定を環境に合わせて調整することで、Web サーバーのパフォーマンスをさらに最適化することができます。

次の環境変数を `webserver_root/bin/startapa` で設定します。ここで、`webserver_root` は、Web サーバーがインストールされているルートディレクトリです。

```
export AIXTHREAD_SCOPE=S
export AIXTHREAD_MNRATIO=1:1
export AIXTHREAD_MUTEX_DEBUG=OFF
export AIXTHREAD_RWLOCK_DEBUG=OFF
export AIXTHREAD_COND_DEBUG=OFF
export CORE_NAMING=true
export YIELDLOOPTIME=number_of_CPUs_on_web_server_machine
export SPINLOOPTIME=1000
export MALLOCMULTIHEAP=heaps:number_of_CPUs_on_web_server_machine,considersize
export MALLOCTYPE=buckets
export LDR_CNTRL=IGNOREUNLOAD@LOADPUBLIC@PREREAD_SHLIB@MAXDATA=0x60000000
```

MALLOCMULTIHEAP パラメータと YIELDLOOPTIME パラメータの値には、Web サーバースマシンの CPU 数を含める必要があります。たとえば、CPU が 2 つある場合、これらのパラメータは次のようになります。

```
export MALLOCMULTIHEAP=heaps:2,considersize
export YIELDLOOPTIME=2
```

## IBM HTTP Server のスレッド数を設定するには

- 1 テキストエディタを使用して、`web_server_install/conf/httpd.conf` ファイルの `workers.c` セクションのパラメータの値を設定します。ここで、`web_server_install` は、Web サーバーがインストールされているルートディレクトリです。パラメータ値を次のように設定します。

<code>ThreadLimit</code>	<code>N</code>
<code>StartServers</code>	<code>1</code>
<code>ServerLimit</code>	<code>1</code>
<code>MaxClients</code>	<code>N</code>
<code>MinSpareThreads</code>	<code>1</code>
<code>MaxSpareThreads</code>	<code>N</code>
<code>ThreadsPerChild</code>	<code>N</code>
<code>MaxRequestsPerChild</code>	<code>0</code>

ここで、各項目は次のようになります。

`N` は、同時ユーザー（スレッド）の最大数の 1 または 1.2 倍。適切なパラメータ値は、Web サーバーがサポートする必要がある同時ユーザー数の 1.2 倍以下に設定します。

`MaxClients` パラメータまたは `ThreadLimit` パラメータを高い値に設定すると、メモリ使用量が増加します。メモリ使用量を減らす必要がある場合は、これらのパラメータに設定する値を小さくしてください。

Web サーバーは、デフォルトで永続的な接続を提供します。Web サーバーが永続的な接続を提供しないようにする場合は、`KeepAlive` パラメータを `Off` に設定することを考慮してください。これによって、接続を解放して再利用できるようにします。ただし、TCP/IP 接続を閉じてから新しい接続をセットアップするためにメモリが使用されるので、コストがかかります。

`KeepAliveTimeout` パラメータの値を小さくすると、スレッドはより迅速に再利用されます。

- 2 `httpd.conf` ファイルで、次の値も設定します。

- `ServerName` の値は、SWSE のインストールに使用したプライマリインターネットアドレスと同じにする必要があります。
- `User` と `Group` の値を有効なマシンユーザー ID とグループの値に変更します。
  - Siebel アプリケーションが必要とするアクセス権以外のファイルアクセス権をこのユーザー ID に許可しないのが理想です。ただし、このユーザーには SWSE インストールディレクトリおよびそのサブディレクトリに対する読み取り、書き込み、および実行の完全なアクセス権を与える必要があります。
  - グループは、このサーバーの実行専用を作成することをお勧めします。

**注意：** セキュリティ上の理由により、`User` または `Group` に `root` は使用しないことをお勧めします。

- `UseCanonicalName` の値は、`Off` に設定することをお勧めします。Web サーバーを負荷分散している場合は、必ず `Off` に設定してください。
- IHS の CGI 機能を使用しない場合は、CGI モジュールをロードする行をコメントにすることができます。これによって、子プロセスが常に 1 つになるので、IHS プロセスの追跡が簡単になります。この行は次のとおりです。

```
LoadModule cgid_module modules/mod_cgid.so
```

## AIX 用の Siebel Server のチューニング

AIX には、Siebel Server のパフォーマンスを最適化するためのチューニング可能な環境変数がいくつか用意されています。これらの環境変数とそれぞれの値は、Siebel Server が起動するときの起動パラメータとして使用します。201 ページの表 5 および 202 ページの表 6 で、これらの各環境変数とそれぞれの推奨設定について説明します。

表 5 最適化に使用する \$SIEBEL\_ROOT/siebenv の環境変数

環境変数	値	説明
AIXTHREAD_SCOPE	S	競合範囲を設定します。S は、システムベースの競合範囲 (1:1) を意味します。
AIXTHREAD_MNRATIO	1:1	実行可能 pthreads を処理するために採用する必要があるカーネルスレッドの数の M:N の比率を設定します。
AIXTHREAD_MUTEX_DEBUG	OFF	デバッガで使用されるアクティブなミューテックスのリストを管理します。
AIXTHREAD_RWLOCK_DEBUG	OFF	デバッガで使用される読み書きロックのリストを管理します。
AIXTHREAD_COND_DEBUG	OFF	デバッガで使用される条件変数のリストを管理します。

表 6 最適化に使用する \$SIEBEL\_ROOT/bin/siebmtshw の環境変数

環境変数	値	説明
SPINLOOPTIME	1000	別のプロセッサに譲るまでにビジーロックが再試行される回数を設定します。
YIELDLOOPTIME	4	ビジーロックがブロックされるまでにプロセッサを譲る回数を設定します (libpthreads の場合のみ)。この変数は、少なくとも CPU の数に設定します。
MALLOCTYPE	buckets	malloc buckets は、デフォルトのアロケータのバケットベースの拡張をオプションで提供します。この機能によって、小さな割り当て要求を大量に発行するアプリケーションで malloc のパフォーマンスが向上します。  malloc buckets を有効にすると、定義済みのブロックサイズ範囲内にある割り当て要求が malloc buckets で処理されます。他の要求はすべて、デフォルトのアロケータが通常の方法で処理します。
MALLOCMULTIHEAP	heaps:n	プロセスプライベートセグメント内のヒープの数を設定します。n には、サーバー上のプロセッサの数と同じ数を指定する必要があります。
LDR_CNTRL	IGNOREUNLOAD@LOADPUBLIC@PREREAD_SHLIB@MAXDATA=0x60000000	LOADPUBLIC オプションは、システムローダーに対して、アプリケーションから要求されたすべてのモジュールをグローバル共有ライブラリセグメントにロードするように指示します。LDR_CNTRL を、ユーザーの環境、または可能であれば追加のメモリを要求する実行可能プログラムを起動するシェルスクリプト内で設定します。  この MAXDATA の値は、この環境から起動されるすべての実行可能プログラムに対して 256 MB のセグメントを 6 つ予約します。この値は、デフォルトの実行可能プログラム設定よりも優先されます。デフォルト値は、実行可能プログラムによって異なります。デフォルト値を使用すると、Siebel コンポーネントは、MaxTasks パラメータに対して最大値 5000 をサポートできます。この値によって、MaxTasks は 9000 まで設定できます。  環境に応じて、最大で 7 つのセグメントを予約することができます。このように多くのセグメントを使用できない場合、Siebel Server は非常に早く終了します。

## AIX 用のカーネル設定のチューニング

AIX での Siebel Server または Web サーバーのパフォーマンスを最適化するためにチューニングできる AIX カーネル設定が多数あります。これには、仮想メモリ管理設定や TCP 設定などがあります。これらの設定を変更するには、root 権限が必要です。AIX 5.2 では、vmtune の代わりに vmo を使用してカーネル設定をチューニングすることもできます。

AIX のカーネル設定については、オペレーティングシステムのベンダーのマニュアルを参照してください。

### vmtune を使用してカーネル設定を変更するには

- 1 vi などのテキストエディタを使用し、編集用に /etc/rc.net ファイルを開きます。
- 2 vmtune の設定を次のように変更します。

```
if [ -f /usr/samples/kernel/vmtune ] ; then
    /usr/samples/kernel/vmtune -p 5 -P 8 -f 720 -F 768 -b 200 -s 1
```

minfree (vmtune では -f) と maxfree (vmtune では -F) の値の設定には、次の式を使用します。

- $\text{minfree} = \text{number\_of\_CPUs} \times 120 = 6 \times 120 = 720$
- $\text{maxfree} = \text{number\_of\_CPUs} \times (120 + \text{maxpgahead}) = 6 \times (120 + 8) = 768$

ここで、各項目は次のとおりです。

*number\_of\_CPUs* = チューニングしている AIX サーバーの CPU 数 (6 など)

*maxpgahead* = maxpgahead (vmtune に対しては -R) パラメータの値 (8 など)

- 3 ネットワークオプションを次のように変更します。

```
if [ -f /usr/sbin/no ] ; then
    /usr/sbin/no -a rfc1323=1
    /usr/sbin/no -a tcp_sendspace=24576
    /usr/sbin/no -a tcp_recvspace=24576
    /usr/sbin/no -a rfc2414=1
    /usr/sbin/no -a tcp_init_window=3
    /usr/sbin/no -a use_isno=0
    /usr/sbin/no -a tcp_nagle_limit=0
```

- 4 次のようにすべてのユーザー制限 (ulimit) が、-1 (無制限) に設定されていることを確認します。

```
ulimit -a
```

**備考:** 制限の設定を変更するには、すべての ulimit パラメータ値を -1 (無制限) に変更して、/etc/security/limits ファイルを更新します。

- 5 変更を保存し、エディタを終了します。
- 6 マシンを再起動して新しい設定を有効化します。

# Solaris 用の Siebel Business Applications のチューニング

ここでは、Solaris で Siebel アプリケーションを実行できるように Web サーバー、オペレーティングシステム、および Siebel Enterprise コンポーネントを設定およびチューニングする方法について説明します。次のトピックが含まれています。

- 204 ページの「Solaris 用の Sun Java System Web Server のチューニング」
- 205 ページの「Solaris 用のカーネル設定のチューニング」
- 206 ページの「Solaris 用の Siebel Server のパフォーマンスの最大化」
- 207 ページの「Solaris 用の AOM インスタンスのチューニング」

## Solaris 用の Sun Java System Web Server のチューニング

負荷が高い Web サーバーでは、ユーザーが Web サーバーへの接続の問題に直面している場合があります。この問題を解決するには、`ndd` コマンドを使用して、`tcp_conn_req_max_q` と `tcp_conn_req_max_q0` のデフォルト値を変更します。`ndd` コマンドの使用方法については、[207 ページの「Solaris 用の AOM インスタンスのチューニング」](#)を参照してください。

次の手順を使用して、Sun Java System Web Server のパフォーマンスを最適にチューニングすることも必要です。

### Sun Java System Web Server をチューニングするには

- 1 `vi` などのテキストエディタを使用して、`Webserver_Root/config/magnus.conf` ファイルを開きます。ここで、`Webserver_Root` は、Sun Java System Web Server のルートパスです。
- 2 `RqThrottle` パラメータを 1200 に設定します。  
`RqThrottle` パラメータでは、Web サーバーで処理できる同時トランザクションの最大数を指定します。デフォルト値は 512 です。この値を 2048 に変更すると、実行されるトランザクションの遅延を最小限に抑えることができます。
- 3 `MaxKeepAliveConnections` パラメータを追加するか、または値を変更して、1000 に設定します。デフォルト値は 200 です。
- 4 変更を `magnus.conf` ファイルに保存します。
- 5 Web サーバーを再起動します。  
Sun Java System Web Server のパラメータに前述の変更を加えたら、Sun Java System Web Server をホストするワークステーションで次のパラメータを変更します。
- 6 編集用に `/etc/system` を開きます。

- 7 次の Solaris システムパラメータを設定します。

パラメータ	範囲	デフォルト値	チューニング後の値	説明
rlim_fd_max	/etc/system	1024	8192	開いているファイル記述子制限を処理します。予想されるロード(関連ソケット、ファイル、パイプが存在する場合)分を計上する必要があります。
rlim_fd_cur	/etc/system	64	8192	

- 8 Sun Java System Web Server をホストしているワークステーションを再起動します。

## Solaris 用のカーネル設定のチューニング

1 つの Solaris 環境で複数の Siebel Server または Web サーバーを実行するには、Solaris カーネルパラメータを Solaris サーバーの各リリース固有の推奨値に設定する必要があります。Solaris で実行している Siebel Server または Web サーバーに固有のパラメータ推奨値については、Oracle の Siebel エキスパートサービスにお問い合わせください。

一般的に Siebel アプリケーションでは(特に Siebel Server では)、パフォーマンスに重大な影響を与える Solaris カーネルパラメータ設定が多数あります。これらのパラメータには、ファイル記述子、スタックサイズ、メモリ、セマフォなどの要素を管理するパラメータが含まれます。

Solaris カーネルパラメータは、設定ファイル /etc/system 内にあります。これらのパラメータの設定を変更するには、このファイルを手動で編集し、変更を保存してシステムを再起動する必要があります。

通常、Solaris カーネルメモリパラメータの設定は比較的低い値に設定します。ただし、Siebel Server アプリケーションなどの大容量メモリモデルのアプリケーションの場合は、一部のパラメータに割り当てられている値を増やすことをお勧めします。

**注意：** Solaris 環境で Siebel Server を実行する際にデフォルト値以下の Solaris カーネルパラメータを使用すると、一部の Siebel Server コンポーネントでは、パフォーマンス上の重大な問題が発生し、SIGABRT エラーや SIGSEV エラーとなる場合があります。

### Seibel Server 用に Solaris カーネル設定をチューニングするには

1 vi などのテキストエディタで、編集用に /etc/system ファイルを開きます。

2 一般的な設定を行う次の行を追加するか、または変更します。

```
set rlim_fd_cur = 8192
set rlim_fd_max = 8192
```

3 共有メモリの設定を行う次の行を追加するか、または変更します。最初の行で、Solaris 32 ビットか 64 ビットを選択します。

```
set shmsys:shminfo_shmmax = 0xffffffff [or] 0xffffffffffffffff
set shmsys:shminfo_shmmni = 1024
set shmsys:shminfo_shmseg = 1024
```

4 メッセージキューの設定を行う次の行を追加するか、または変更します。

```
set msgsys:msginfo_msgmax = 4096
```

5 セマフォを設定する次の行を追加するか、または変更します。

```
set semsys:seminfo_semaem = 16384
set semsys:seminfo_semopm = 100
set semsys:seminfo_semmni = 4096
set semsys:seminfo_semmns = 16384
set semsys:seminfo_semmnu = 4096
set semsys:seminfo_semume = 2500
set semsys:seminfo_semmsl = 500
```

6 変更を保存し、エディタを終了します。

7 マシンを再起動して新しい設定を有効化します。

## Solaris 用の Siebel Server のパフォーマンスの最大化

Solaris で Siebel Server を実行する際に CPU パフォーマンスを最大化するには、次の手順に示すように 4 MB のヒープサイズと 64 KB のスタックサイズの最適設定で、MPSS (Multiple Page Size Support) を使用します。

### 4 MB のヒープサイズと 64 KB のスタックサイズの最適設定で MPSS をセットアップするには

1 vi などのテキストエディタで、編集用に /etc/system ファイルを開きます。

2 ファイルに次の行を追加します。

```
set kernel_cage_enable=1
```

3 サーバーを再起動します。

4 ファイルに次の行を含めて MPSS の設定ファイル (mpss.cfg) を作成します。

```
sieb*:4M:64K
```

ここで、4M はヒープサイズ (4 MB)、64K はスタックサイズ (64 KB) です。

- 5 次の行を `$Siebel_root/siebsrvr/bin/siebmtshw` ファイルに追加します。

```
LD_PRELOAD=/usr/lib/mpss.so.1
```

```
MPSSCFGFILE= フルパス (ステップ 4 で作成した設定ファイルのファイル名を含む)
```

```
MPSSERRFILE= フルパス (エラーが発生した場合に生成する MPSS エラーログのファイル名を含む)
```

```
export LD_PRELOAD MPSSCFGFILE MPSSERRFILE
```

## Solaris 用の AOM インスタンスのチューニング

50 を超える Application Object Manager のインスタンス (AOM 用マルチスレッドプロセス) を実行する Solaris マシンでは、1 つ以上のプロセスが正しく起動しなくても、残りのプロセスは正しく機能している場合があります。起動しないプロセスのログファイルには、それらが正しく起動していないことが示されます。このような場合は、`ndd` コマンドを使用して `tcp_conn_req_max_q` と `tcp_conn_req_max_q0` のデフォルト値を変更します。

### TCP 値を変更するには

- 1 `root` としてログインします。

- 2 `ndd` コマンドを実行します。

**備考：** 太字は応答を表しています。

```
ndd /dev/tcp
```

```
name to get/set ? tcp_conn_req_max_q
```

```
value ? 1024
```

```
name to get/set ? tcp_conn_req_max_q0
```

```
value ? 4096
```

- 3 `vi` などのテキストエディタを使用して、次の行を `/etc/system` ファイルに追加します。

```
set tcp:tcp_conn_req_max_q = 1024
```

```
set tcp:tcp_conn_req_max_q0 = 4096
```

- 4 変更を保存し、エディタを終了します。

## HP-UX 用の Siebel Business Applications のチューニング

ここでは、HP-UX で Siebel アプリケーションを実行できるように Web サーバー、オペレーティングシステム、および Siebel Enterprise コンポーネントを設定およびチューニングする方法について説明します。次のトピックが含まれています。

- 208 ページの「HP-UX 用の HP Apache Web Server のチューニング」
- 210 ページの「HP-UX 用のカーネル設定のチューニング」
- 211 ページの「HP-UX スケジューラの許可の設定」

### HP-UX 用の HP Apache Web Server のチューニング

ここでは、HP Apache Web Server の環境変数の推奨される初期設定について説明します。これらの設定を環境に合わせて変更することで、Web サーバーのパフォーマンスをさらに最適化することができます。

HP Apache Web Server の ThreadLimit デフォルト値は 64 ですが、はるかに大きい数値に設定することもできます。最大設定値は、カーネル設定によって異なります。ThreadsPerChild と MaxClients は関連するディレクティブです。

- ThreadLimit = 20000 は、HP Apache Web Server でサポートされる最大値です。この値を、システムがサポートする数値にリセットできます。  
**備考：** ThreadLimit ディレクティブは、他のディレクティブを実行する前に実行する必要があります。
- ThreadsPerChild = 各子スレッド数。この値には、ThreadLimit を超える値を設定できません。
- MaxClients = 最大接続。この値には、ThreadsPerChild を超える値を設定できません。

## HP Apache Web Server のスレッド数を設定するには

- 1 テキストエディタを使用して、`web_server_install/conf/httpd.conf` ファイルの `workers.c` セクションのパラメータの値を設定します。ここで、`web_server_install` は、Web サーバーがインストールされているルートディレクトリです。パラメータ値を次のように設定します。

<code>ThreadLimit</code>	<code>N</code>
<code>StartServers</code>	<code>1</code>
<code>ServerLimit</code>	<code>1</code>
<code>MaxClients</code>	<code>N</code>
<code>MinSpareThreads</code>	<code>1</code>
<code>MaxSpareThreads</code>	<code>N</code>
<code>ThreadsPerChild</code>	<code>N</code>
<code>MaxRequestsPerChild</code>	<code>0</code>

ここで、各項目は次のとおりです。

`N` は、同時ユーザー（スレッド）の最大数の 1.2 倍または 1.5 倍。適切なパラメータの値は、Web サーバーがサポートする必要がある同時ユーザー数より大きくする必要があります。ただし、パラメータ値をここで説明している値より高く設定すると、不必要に追加メモリを消費することになります。

**備考：** 多重化セッションを使用していない場合、カーネルパラメータ `max_thread_proc` を `2N` より大きい値に設定します。

- 2 `User` と `Group` の値を有効なマシンユーザー ID とグループの値に変更します。
  - Siebel アプリケーションが必要とするアクセス権以外のファイルアクセス権をこのユーザー ID に許可しないのが理想です。ただし、このユーザーには SWSE インストールディレクトリおよびそのサブディレクトリに対する読み取り、書き込み、および実行の完全なアクセス権を与える必要があります。
  - グループは、このサーバーの実行専用を作成することをお勧めします。

**注意：** セキュリティ上の理由により、`User` または `Group` に `root` は使用しないことをお勧めします。

- 3 `MaxKeepAliveRequests` を 0 に設定します。

## HP-UX 用のカーネル設定のチューニング

HP-UX カーネルパラメータを次のような値に変更します（推奨ガイドライン）。これらの変更を行うには、HP-UX System Administration Manager (SAM) ツールを使用します。

nproc	4096	-	4096
ksi_alloc_max	32768	-	(NPROC*8)
max_thread_proc	4096	-	4096
maxdsiz	0x90000000	-	0x90000000
maxdsiz_64bit	2147483648	-	2147483648
maxfiles	4000	-	4000
maxssiz	401604608	-	401604608
maxssiz_64bit	1073741824	-	1073741824
maxtsiz	0x40000000	-	0x40000000
maxusers	128	-	128
msgmap	4098	-	(NPROC+2)
msgmni	4096	-	(NPROC)
msgtql	4096	-	(NPROC)
ncallout	8000	-	8000
nclist	2148	-	(100+16*MAXUSERS)
nctime	35840	-	(8*NPROC+2048+VX_NCSIZE)
nfile	67584	-	(16*NPROC+2048)
ninode	34816	-	(8*NPROC+2048)
nkthread	7184	-	((NPROC*7)/4)+16
nproc	4096	-	4096
nsysmap	8192	-	((NPROC)>800?2*(NPROC):800)
nsysmap64	8192	-	((NPROC)>800?2*(NPROC):800)
semmap	1026	-	1026
semnmi	1024	-	1024
semnms	16384	-	((NPROC*2)*2)
semnmu	2048	-	2048
semume	256	-	256
shmmax	0x40000000	Y	0x40000000
shmmni	1024	-	1024
shmseg	1024	Y	1024
vps_ceiling	64	-	64

## HP-UX スケジューラの許可の設定

次のように変更すると、Siebel Server は起動時に HP-UX スケジューラを実行し、HP-UX の Siebel Business Applications のパフォーマンスが向上します。この変更を行うには、ルート権限が必要です。

### HP-UX スケジューラに許可を設定するには

- 1 /etc/privgroup ファイルに次の行を追加します。必要であれば、このファイルを作成します。

```
-g RTSCHED
```

- 2 ファイルを保存して終了します。

- 3 次のコマンドを実行します。

```
setprivgrp -f /etc/privgroup
```

- 4 次のコマンドを実行して、グローバル RTSCHED 許可が設定されていることを確認します。

```
getprivgrp
```

コマンドが成功すると、システムは次のように応答します。

```
global privileges:RTSCHED
```



# 14 Siebel ARM による Siebel アプリケーションのパフォーマンスの監視

この章では、Siebel Application Response Measurement (Siebel ARM) の機能を使用したパフォーマンスの監視の概要について説明します。ここで説明する内容は、次のとおりです。

- 213 ページの「Siebel Application Response Measurement について」
- 214 ページの「Siebel ARM パラメータおよび変数について」
- 217 ページの「Siebel ARM の有効化と設定」
- 218 ページの「Siebel ARM に関する最善の方法」

Siebel ARM 機能が収集するデータの分析方法については、219 ページの「Siebel ARM データの分析」を参照してください。

**備考：** 追加の診断ツールおよび監視ツールは、Siebel 管理フレームワークを使って利用できます。Siebel 環境での Siebel 管理フレームワークの導入はオプションです。Siebel 管理フレームワークについては、『Siebel システム監視および診断ガイド』を参照してください。

## Siebel Application Response Measurement について

Siebel ARM は、Siebel Business Applications の重要なパフォーマンスデータを取り込むためのフレームワークです。このデータは、バイナリファイル形式で保存されます。Siebel ARM は、Siebel Server インフラストラクチャ内の重要な監視ポイントで応答時間を取り込みます。これらの Siebel ARM 監視ポイントは、Siebel インフラストラクチャ内で次の個別のエリアに分類されます。

- **Web Server Time** : Web サーバー上でリクエストに費やされた時間。
- **Infra-Network Time** : Web サーバーと Siebel Server の間でリクエストに費やされた時間 (ネットワーク時間を含みます)。
- **Siebel Server Time** : Siebel Server とデータベースサーバーの間でリクエストに費やされた時間 (Server Thread (SMI) と任意のデータベース層の呼び出し間の時間)。
- **Database Time** : 任意の Siebel データベース層呼び出しに費やされた時間です。
- **Application-Specific Time** : インフラストラクチャのアプリケーション固有のエリアで費やされた時間。

Siebel ARM 機能は、次のリストのインフラストラクチャおよびアプリケーション固有エリアでシステムパフォーマンスを監視します。次のエリアは Siebel ARM 出力に表示されるとおりにリストされています。エリア名の後のかっこ内の名前は、エリア記号を表し、Siebel ARM 出力にも表示されます。

- SARM Framework (SARM)
- Siebel Repository (SRF)
- Web Engine (SWE)
- 割当マネージャ (AM)
- Build Web Page (SWEPAGE)
- Fulfillment Engine (FSFULFILL)

- Web Server Plugin (SWSE)
- Database Connector (DBC)
- Application Server (INFRA)
- Workflow (WORKFLOW)
- eScripts (SCRIPT)
- Request Manager (SRM)
- Request Broker (SRB)
- File System Manager (FSM)
- Business Service (BUSSRVC)
- Email Response (EMR)
- Security / Authentication (SEC)
- Object Manager (OBJMGR)
- Preventative Maintenance Engine (FSPREVMNT)
- Siebel Loyalty (LOY)
- Handheld Sync (HHSYNC)
- SmartScript (SMARTSCRIPT)
- Siebel Anywhere (SIEBANYWHERE)
- Communications Channel Manager (CSMM)
- Communications Server Service (CSS)
- Customer/Order Management - Configurator (COMCFG)
- EAI Transports (EAITRANSP)
- MWC Profiler (MWC)<sup>1</sup>
- Communications Outbound Manager (COM)
- Universal Inbox (UINBOX)

1. MWC = Mobile Web Client

前述の各エリアには、そのエリアのタイミングとパフォーマンスを定義する、1 つまたは複数のサブエリアが含まれます。Siebel ARM ファイルに存在するエリアとサブエリア数は、粒度レベルによって異なります。このレベルは、パラメータ SARM Granularity Level によって設定されます。このパラメータについて詳しくは、[214 ページの「Siebel ARM パラメータおよび変数について」](#)を参照してください。

Siebel ARM ファイルの形式については、[219 ページの「Siebel ARM ファイルについて」](#)を参照してください。

## Siebel ARM パラメータおよび変数について

Siebel Server の次のパラメータ、および Web サーバーの次の環境変数により、Siebel ARM 機能の有効化と設定を行います。Siebel ARM パラメータおよび環境変数は、機能が同じであり、命名規則も似ています。

各 Siebel ARM パラメータと同等な環境変数のリストについては、[214 ページの表 7](#) を参照してください。この表に続いて、各パラメータと環境変数について説明します。

表 7 Siebel ARM パラメータおよび環境変数

パラメータ表示名	パラメータエイリアス	環境変数名
SARM Granularity Level	SARMLevel	SIEBEL_SARMLevel
SARM Buffer Size	SARMBufferSize	SIEBEL_SARMBufferSize
SARM Period	SARMPeriod	SIEBEL_SARMPeriod
SARM Max Number of files	SARMMaxFiles	SIEBEL_SARMMaxFiles
SARM Data File Size	SARMFileSize	SIEBEL_SARMFileSize

これらのパラメータと変数を使った Siebel ARM の有効化については、[217 ページの「Siebel ARM の有効化と設定」](#)を参照してください。

## SARM Granularity Level

Siebel ARM ファイルに記録される応答測定の詳細レベルを指定し、Siebel ARM 機能を有効または無効にすることができます。このパラメータまたは環境変数の設定は次のとおりです。

- **0 (オフ)**：この設定はデフォルト値であり、Siebel ARM を無効にします。
- **1 (ARM)**：この設定は、Application Response Measurement (ARM) の標準に基づいて、一般的なアプリケーションパフォーマンスを記録します。このレベルでは、Siebel ARM はプロセスおよびコンポーネント境界、サードパーティ製のソフトウェアの呼び出し、データベース測定、ワークフローの実行、スクリプトパフォーマンスなどの情報を収集します。一般的なパフォーマンス監視にはこのレベルを使用します。
- **2 (詳細)**：この設定は、レベル 1 の情報に加え、ワークフロー実行のステップ、大きなオブジェクトの作成、大きなファイルの読み取り、重要なアーキテクチャエリアの交差などの詳細情報を取り込みます。問題の診断にはこのレベルを使用します。

## SARM Buffer Size

Siebel ARM フレームワークでは、バッファ化されたデータ生成メカニズムが使用されています。Siebel ARM は、データを収集し、それをメモリに保存します。そして、SARM Buffer Size によって定義されたしきい値にメモリ内のデータサイズが達すると、Siebel ARM は保存されたデータを物理ディスク上のファイルに出力します。SARM Buffer Size パラメータまたは環境変数は、バイト単位で指定します。デフォルト値は、5,000,000 バイト (約 5 MB) です。有効な設定範囲は、100,000 ~ 50,000,000 バイトです。

**備考：**また、Siebel ARM は、保存されたデータを経過時間にも基づいてファイルに出力します。これはパラメータまたは環境変数 SARM Period によって定義されます。このパラメータの設定は、SARM Buffer Size によって定義されるしきい値ではなく、ファイルに保存されるデータのサイズを決定します。

たとえば、SARMBufferSize が 5 MB であり、コンポーネントのインスタンス (プロセス) が 5 つある場合、使用される合計メモリは 25 MB です。

## SARM Period

Siebel ARM は、データを収集し、それをメモリに保存します。SARM Period パラメータまたは環境変数によって指定される期間は、SARM Buffer Size の値セットにかかわらず、Siebel ARM が保存されたデータを物理ディスク上のファイルに出力するタイミングを決定します。このパラメータは分単位で指定され、デフォルト値は 3 分です。このパラメータの有効な設定範囲は、1 ~ 60 分です。

**備考：**経過時間に基づいて Siebel Server のパフォーマンスデータを出力する場合にのみ、SARM Period を使用します。Siebel ARM は、SARM Buffer Size 値のみに基づいて Web サーバーのパフォーマンスデータを出力します。

メモリ内のデータサイズに基づいたメモリからのデータの出力については、SARM Buffer Size の説明を参照してください。

## SARM Max Number of Files

コンポーネントインスタンスごとに作成される Siebel ARM ファイルの最大数を指定します。デフォルト値は 4 であり、Siebel ARM が作成するファイル数には、Siebel 指定の上限はありません。パラメータまたは環境変数 SARM Data File Size は、新しいファイルが物理ディスク上に保存されるまでの許容ファイルサイズを設定します。

コンポーネントプロセスごとのアクティブな Siebel ARM ファイル数は、SARM Max Number of Files の値に 1 を加えた値です。つまり、Siebel ARM は、「SARM Max Number of Files+1」番目のファイルが SARM Data File Size に達した後にのみ、プロセスの最も古いファイルを削除します。

これらのパラメータまたは環境変数を使用したメモリ使用量の計算方法については、SARM Data Size の説明を参照してください。

## SARM Data File Size

Siebel ARM が、物理ディスク上の新しいファイルにデータを保存するまでの許容ファイルサイズを指定します。このパラメータは、バイト単位で指定します。デフォルト値は 15,000,000 バイト（15 MB）であり、ファイルサイズには Siebel 指定の上限はありません。

Siebel ARM は、指定のサイズに達するまで現在のファイルにファイルセグメントを追加し続けます。そして、ファイルの制限に達すると、新しいファイルを作成します。パラメータまたは環境変数 SARM Max Number of files は、Siebel ARM によって管理されるファイル数を設定します。

SARM Max Number of Files によって指定されたファイル数に達すると（つまり、サイズ SARM Data File Size のファイルが SARM Max Number of Files 個になると）、次のファイルが SARM Data File Size の制限に達したときに、Siebel ARM は最初のファイル（すなわち、最も古いファイル）を削除します。したがって、使用される最大ディスク容量は、「SARM Max Number of Files+1」に SARM Data File Size を掛けて算出されるバイト数です。このメモリ量は、プロセスあたり（コンポーネントインスタンスあたり）の値です。

たとえば、SARM Data File Size が 15 MB、SARM Max Number of Files が 4、コンポーネントのインスタンス（プロセス）が 5 つ存在する場合、消費される最大ディスク容量は約 375 MB（すなわち 1 ファイルあたり 15 MB×1 プロセスあたり 5 ファイル×5 プロセス（コンポーネントのインスタンス））です。

## Siebel ARM の有効化と設定

Siebel Application Response Measurement (Siebel ARM) の有効化と設定には、次の 2 つのタスクがあります。

- Siebel Server 上の Siebel ARM パラメータの設定
- Web サーバー上の Siebel ARM 環境変数の設定

デフォルトでは、Siebel ARM は無効になっています。

### Siebel Server 上の Siebel ARM パラメータの設定

次の手順を実行して、Siebel Server 上で Siebel ARM を有効化および設定します。

**備考：** Siebel ARM パラメータが表示されない場合、サーバーコンポーネント Server Manager (エイリアス ServerMgr) のパラメータ Show Advanced Objects (エイリアス ShowAdvancedObjects) を「TRUE」に設定します。

#### Siebel Server 上で Siebel ARM の有効化と設定を行うには

- 1 Siebel Server 上で Siebel ARM を有効化するには、パラメータ SARM Granularity Level (エイリアス SARMLLevel) の値を 1 または 2 に設定します。このパラメータと設定については、[214 ページの「Siebel ARM パラメータおよび変数について」](#)を参照してください。

エンタープライズレベル、Siebel Server レベル、またはサーバーコンポーネントレベルのいずれかで Siebel ARM を有効化できます。

- 2 Siebel Server 上の Siebel ARM ファイルの特性を設定するには、その他の Siebel ARM 関連のパラメータを設定します。これらのパラメータについては、[214 ページの「Siebel ARM パラメータおよび変数について」](#)を参照してください。

Siebel Server レベルまたはサーバーコンポーネントレベルで Siebel ARM を設定できます。

Server Manager の GUI またはコマンドラインインターフェイスを使って Siebel Server パラメータを設定する方法、およびパラメータ管理の基本的な情報については、『Siebel システム管理ガイド』を参照してください。

### Web サーバー上の Siebel ARM 環境変数の設定

次の手順を実行して、Web サーバーをホストするマシン上で Siebel ARM を有効化および設定します。

#### Web サーバー上で Siebel ARM の有効化と設定を行うには

- 1 Web サーバーをホストするマシン上で Siebel ARM を有効化するには、環境変数 SIEBEL\_SARMLLevel の値を 1 または 2 に設定します。このパラメータおよび設定については、[214 ページの「Siebel ARM パラメータおよび変数について」](#)の SARM Granularity Level の説明を参照してください。
- 2 Web サーバーをホストするマシン上で Siebel ARM ファイルの特性を設定するには、その他の Siebel ARM 関連の環境変数を設定します。これらのパラメータについては、[214 ページの「Siebel ARM パラメータおよび変数について」](#)を参照してください。

Windows と UNIX での環境変数の設定については、『Siebel システム管理ガイド』を参照してください。

## Siebel ARM に関する最善の方法

Siebel ARM ファイルを変換する場合、次の最善の方法に従うことをお勧めします。

- 本番環境を監視する場合は、Siebel ARM Granularity Level をレベル 1 に設定し、診断目的の場合は、Siebel ARM Granularity Level をレベル 2 に設定します。
- Siebel ARM ファイルの作成を無効にするには、SARM Max Number of Files パラメータを 0 に設定します。サードパーティ製の他の ARM ツールとともに使用するために Siebel ARM を有効化する場合、このシナリオは役立ちます。
- ファイルを変換する前に、Siebel ARM 機能によりデータが Siebel ARM ファイルに書き出されていることを確認します。Siebel ARM 機能では、空の Siebel ARM ファイルを作成してから、データをファイルに書き出します。このプロセスについては、[214 ページの「Siebel ARM パラメータおよび変数について」](#)の SARM Data File Size および SARM Period の説明を参照してください。
- Siebel ARM ファイルが常に空である場合は、SARM Memory Size Limit (エイリアス SARMMemory) または SARM Period (エイリアス SARMPERIOD) の設定をより小さな値に変更します。このプロセスについては、[214 ページの「Siebel ARM パラメータおよび変数について」](#)の SARM Data File Size および SARM Period の説明を参照してください。
- コマンド内で Siebel ARM ファイルを参照する場合に、必要に応じて Siebel ARM ファイルの名前およびパス名が正しいことを確認します。
- Siebel ARM Analyzer Tool が大きな Siebel ARM ファイルを変換できないか、または出力ファイルが大きすぎる場合は、Siebel ARM Analyzer Tool で -p フラグを使用して Siebel ARM ファイルを分割します。-p フラグについての詳細は、[221 ページの表 8](#) を参照してください。
- Siebel ARM ファイルを連結することにより、特定プロセスのパフォーマンスのデータ量を増やします。たとえば、Siebel ARM 機能によりプロセスごとに数多くの Siebel ARM バイナリファイルが保存されるので、各プロセスに対する複数のリクエストのパフォーマンスデータを表示するために、これらのファイルを連結します。保存されるファイルの数については、[214 ページの「Siebel ARM パラメータおよび変数について」](#)の SARM Max Number of Files の説明を参照してください。

**ヒント：** Windows では、サードパーティ製ユーティリティを使って Siebel ARM ファイルを連結します。UNIX では、コマンド `cat list_of_files > filename.sarm` を使って Siebel ARM ファイルを連結します。

**備考：** 同じプロセスの Siebel ARM ファイルだけを連結してください。

- アプリケーションをカスタマイズする前に、Siebel アプリケーションのパフォーマンス分析データを収集します。このベースラインの測定は、何らかのカスタマイズを行った後、Siebel アプリケーションのパフォーマンスを監視するための良い基準になります。
- 個別ユーザーの特定セッションでパフォーマンスの問題が発生する場合は、ユーザーセッション追跡分析を実行します。ユーザーセッション追跡のデータでは、ユーザーが行った各リクエストと、ベースラインとの対比で最も長い時間がかかったリクエストが識別されます。
- ある時点またはあるプロセスのパフォーマンスを診断する場合は、パフォーマンス集計データを使用します。グループごとにデータをレビューすることで、パフォーマンスが不十分なエリアを診断できます。パフォーマンスデータについて大まかにレビューした後、CSV (カンマ区切り値) 形式の分析を実行することにより詳細なレビューの推定を行います。この分析の実行については、[225 ページの「Siebel ARM データの CSV 変換の実行」](#)を参照してください。
- トレンド分析の実施のために、ある期間のパフォーマンス集計データを編集します。

# 15 Siebel ARM データの分析

この章では、Siebel ARM データの分析方法について説明します。次のトピックが含まれています。

- 219 ページの「Siebel ARM ファイルについて」
- 220 ページの「Siebel ARM Analyzer Tool を使った Siebel ARM ファイルの分析」
- 238 ページの「Siebel ARM Query Tool を使った Siebel ARM ファイルの分析」

Siebel ARM データの収集方法を含む、Siebel ARM の機能については、第 14 章「Siebel ARM による Siebel アプリケーションのパフォーマンスの監視」を参照してください。

**備考：** 追加の診断ツールおよび監視ツールは、Siebel 管理フレームワークを使って利用できます。Siebel 環境での Siebel 管理フレームワークの導入はオプションです。Siebel 管理フレームワークについては、『Siebel システム監視および診断ガイド』を参照してください。

## Siebel ARM ファイルについて

Siebel ARM 機能を有効にすると、バイナリ形式の Siebel ARM ファイルが以下の場所に保存されます。

- Windows の Siebel Server ログサブディレクトリ：*SIEBSRVR\_ROOT*\log
- UNIX の Siebel Server ログサブディレクトリ：*SIEBSRVR\_ROOT/enterprises/EnterpriseServerName/SiebelServerName/log*
- Siebel Web Server Extension ログサブディレクトリ：*SWSE\_ROOT*\log

Siebel ARM 機能については、213 ページの「Siebel Application Response Measurement について」を参照してください。

Siebel ARM 機能では、次の例のような名前がバイナリデータファイルに付けられます。

```
T200401081744_P001768_N0006.sarm
```

ここで、各項目は次のとおりです。

- T = タイミング情報が後続に示されることを表す定数値。
- 200401081744 = Siebel ARM ファイルの日時を示します。この例は、このファイルが 2004 年 1 月 8 日 17 時 44 分に保存されたことを示しています。
- P = プロセス ID 情報が後続に示されることを表す定数値。
- 001768 = Siebel ARM がデータを収集するプロセス ID を示します。
- N = Siebel ARM ID 情報が後続に示されることを表す定数値。
- 0006 = リストされているプロセス ID に対する Siebel ARM ログ ID 番号を示します。0000 から 9999 まで増加し、9999 に達すると 0000 に戻ります。

- .sarm = Siebel ARM ファイルの拡張子。

**備考：** Siebel ARM 機能では、Web サーバー上に空の Siebel ARM ファイルを作成してから、そのファイルにデータを出力します。この Siebel ARM ファイルへのデータの保存が開始されるのは、SARM Data File Size パラメータの値に達した後です。このプロセスについては、[214 ページの「Siebel ARM パラメータおよび変数について」](#)のパラメータの説明を参照してください。

バイナリ Siebel ARM ファイルに含まれるデータを分析するには、次のいずれかのツールを使って収集されたデータを判読可能な形式に変換する必要があります。

- Siebel ARM Analyzer Tool

Siebel ARM Analyzer Tool は、Siebel ARM によって収集されたパフォーマンスデータを分析するコマンドラインツールです。Siebel ARM Analyzer Tool について詳しくは、[220 ページの「Siebel ARM Analyzer Tool を使った Siebel ARM ファイルの分析」](#)を参照してください。

- Siebel ARM Query Tool

Siebel ARM Query Tool は、Siebel ARM によって収集されたパフォーマンスデータを分析するコマンドラインツールです。Siebel ARM Query Tool について詳しくは、[238 ページの「Siebel ARM Query Tool を使った Siebel ARM ファイルの分析」](#)を参照してください。

- Siebel 診断ツール

このツールでは、グラフィカルユーザーインターフェイスを使って、Siebel ARM によって収集されたパフォーマンスデータを分析できます。このツールについては、『Siebel システム監視および診断ガイド』を参照してください。

## Siebel ARM Analyzer Tool を使った Siebel ARM ファイルの分析

ここでは、Siebel ARM Analyzer Tool を使って Siebel ARM ファイルを分析する方法について説明します。このツールは、バイナリ Siebel ARM ファイルを分析のために判読可能なアウトプットに変換します。このツールについては、[221 ページの「Siebel ARM Analyzer Tool について」](#)を参照してください。

次の項では、Siebel ARM Analyzer Tool を使ってさまざまなタイプの分析用アウトプットを生成する方法について説明します。

- [222 ページの「パフォーマンス集計分析の実行」](#)
- [223 ページの「呼び出しグラフの生成の実行」](#)
- [224 ページの「ユーザーセッション追跡の実行」](#)
- [225 ページの「Siebel ARM データの CSV 変換の実行」](#)

次の項では、Siebel ARM Analyzer Tool が生成する分析用アウトプットのタイプについて説明します。Analyzer Tool は、その変換分析のタイプに基づいて、XML または CSV のいずれかの形式でアウトプットを作成します。詳しくは、次の資料を参照してください。

- [232 ページの「呼び出しグラフの生成分析およびデータについて」](#)
- [235 ページの「ユーザーセッション追跡分析およびデータについて」](#)
- [237 ページの「Siebel ARM から CSV への変換データについて」](#)

## Siebel ARM Analyzer Tool について

Siebel ARM Analyzer Toolは、Siebel ARM機能によって作成されたファイルを解析し、XML(eXtensible Markup Language) 形式またはカンマ区切り値 (CSV) 形式の分析結果を生成します。Siebel ARM Analyzer Tool は、コマンドラインで手動により実行します。このツールで生成可能なアウトプットのタイプについては、[220 ページの「Siebel ARM Analyzer Tool を使った Siebel ARM ファイルの分析」](#)を参照してください。

このコマンドラインユーティリティは、Microsoft Windows 上では Siebel Server ルートディレクトリの bin サブディレクトリ (BIN) の実行可能プログラム sarmanalyzer.exe、UNIX 上では sarmanalyzer として存在します。[221 ページの表 8](#)では、このツールで使用するパラメータについて説明します。

**注意：** Siebel アプリケーションを監視すると、大きな Siebel ARM ファイルが生成されます。そのため、Siebel ARM Analyzer Tool は、極端に大きなバイナリの Siebel ARM ファイルに十分なメモリを割り当てることができず、変換できない場合があります。その場合には、Siebel ARM Analyzer Tool の `-p` フラグを使って、Siebel ARM ファイルをより小さなファイルに分割します。このフラグについては、[221 ページの表 8](#)を参照してください。

表 8 Siebel ARM Analyzer Tool のフラグ

フラグ	説明
-help	Siebel ARM Analyzer Tool とともにこのフラグを使用することにより、使用可能なフラグのリストとその説明を表示します。
-f	Siebel ARM のファイル引数とともにこのフラグを使用することにより、パフォーマンス集計分析を実行します。詳しくは、 <a href="#">222 ページの「パフォーマンス集計分析の実行」</a> を参照してください。
-o	このフラグを使用することにより、Siebel ARM バイナリファイルの分析結果の出力パスとファイルの名前を指定します。選択した分析に基づいた正しいファイル拡張子、すなわち XML または CSV を指定する必要があります。
-d	このフラグおよび引数 XML または CSV を使用することにより、出力ファイル形式のタイプ：XML (eXtensible Markup Language) または CSV (カンマ区切り形式のリスト) を指定します。
-a	パフォーマンス集計分析の実行時に、引数 AREA または DETAILS とともにこのフラグを使用します。この分析については、 <a href="#">222 ページの「パフォーマンス集計分析の実行」</a> を参照してください。
-i	ユーザーセッション追跡分析の実行時に、ディレクトリ引数とともにこのフラグを使用します。この分析については、 <a href="#">224 ページの「ユーザーセッション追跡の実行」</a> を参照してください。
-s	このオプションフラグを使用することにより、ユーザーセッション追跡の開始時刻を指定します。この引数の形式は yyyy-mm-dd hh:mm:ss です。-e フラグとともにこのフラグを使用することにより、時間帯を指定します。
-e	このオプションフラグを使用することにより、ユーザーセッション追跡の終了時刻を指定します。この引数の形式は yyyy-mm-dd hh:mm:ss です。-s フラグとともにこのフラグを使用することにより、時間帯を指定します。
-p	このオプションフラグを使用することにより、大きな Siebel ARM ファイルをより小さなサイズに分割します。フラグ引数として 0 ~ 50 の値を使用します。これは分割したファイルの MB 単位のサイズを示します。デフォルト値は 14 MB です。Siebel ARM Analyzer Tool は、フラグ引数が 0 の場合はデフォルト値を使用します。分割されたファイルには <code>_Snnnn</code> という文字列が末尾に付加されます。ここで、 <code>nnnn</code> は、分割された順番です。

## パフォーマンス集計分析の実行

次の手順に従ってパフォーマンス集計分析のアウトプットを取得します。

パフォーマンス集計分析およびアウトプットについては、[226 ページの「パフォーマンス集計分析およびデータについて」](#)を参照してください。

### パフォーマンス集計分析を実行するには

**1** Siebel Server ルートディレクトリ内の bin サブディレクトリに移動します。

**2** 次のコマンドを使って Siebel ARM Analyzer Tool を実行します。

```
sarmanalyzer -o output_file_name.xml -a aggregate_argument -f sarm_file_name.sarm
```

ここで、各項目は次のとおりです。

*output\_file\_name.xml* = XML 出力ファイルの名前およびパス。

*aggregate\_argument* = Siebel ARM 後処理ツールでデータを集計するエリアによって、AREA または DETAILS のいずれかを指定します。詳しくは、[226 ページの「パフォーマンス集計分析およびデータについて」](#)を参照してください。

*sarm\_file\_name.sarm* = バイナリ Siebel ARM ファイルの名前およびパス。カンマで区切ったリストを使用すると、複数の Siebel ARM ファイルからデータを集計します。

**3** *output\_file\_name.xml* という名前のファイルの、XML アウトプットをレビューします。パフォーマンス集計分析 XML アウトプットの分析については、[226 ページの「パフォーマンス集計分析およびデータについて」](#)を参照してください。

Siebel ARM Analyzer Tool の実行について詳しくは、[220 ページの「Siebel ARM Analyzer Tool を使った Siebel ARM ファイルの分析」](#) および [221 ページの「Siebel ARM Analyzer Tool について」](#)を参照してください。

## 呼び出しグラフの生成の実行

次の手順に従って呼び出しグラフの生成分析のアウトプットを取得します。

呼び出しグラフの生成分析およびアウトプットについては、[232 ページの「呼び出しグラフの生成分析およびデータについて」](#)を参照してください。

### 呼び出しグラフの生成分析を実行するには

**1** Siebel Server ルートディレクトリ内の bin サブディレクトリに移動します。

**2** 次のコマンドを使って Siebel ARM Analyzer Tool を実行します。

```
sarmanalyzer -o output_file_name.xml -d xml -f sarm_file_name.sarm
```

ここで、各項目は次のとおりです。

*output\_file\_name.xml* = XML 出力ファイルの名前およびパス。

-d xml = 呼び出しグラフの生成分析を指定します。

*sarm\_file\_name.sarm* = バイナリ Siebel ARM ファイルの名前およびパス。

**3** *output\_file\_name.xml* という名前のファイルの、XML アウトプットをレビューします。呼び出しグラフ分析の XML 出力の分析については、[232 ページの「呼び出しグラフの生成分析およびデータについて」](#)を参照してください。

Siebel ARM Analyzer Tool の実行について詳しくは、[220 ページの「Siebel ARM Analyzer Tool を使った Siebel ARM ファイルの分析」](#)を参照してください。

## ユーザーセッション追跡の実行

次の手順に従ってユーザーセッション追跡分析のアウトプットを取得します。この分析を実行する前に、Siebel Server および Web サーバーの Siebel ARM ファイルを手動で収集して、共通のディレクトリに格納します。Siebel ARM Analyzer Tool とともに引数としてこのディレクトリを使用します。

ユーザーセッション追跡分析およびアウトプットについては、[235 ページの「ユーザーセッション追跡分析およびデータについて」](#)を参照してください。

**ヒント：** 記録されるデータ量を減らすには、時間枠パラメータ（`-s` 開始時刻および `-e` 終了時刻）を使用します。

### ユーザーセッション追跡分析を実行するには

1 Siebel Server ルートディレクトリ内の bin サブディレクトリに移動します。

2 Siebel ARM Analyzer Tool は、次のコマンドを使って実行します。

```
sarmanalyzer -o output_file_name.xml -u user_name -i SARM_File_Directory -s  
start_time -e end_time
```

ここで、各項目は次のとおりです。

- `output_file_name.xml` = XML 出力ファイルの名前およびパス。
  - `user_name` = 追跡するセッションのユーザー ID。
  - `SARM_File_Directory` = Web サーバーおよび Siebel Server の Siebel ARM ファイルを含むディレクトリ。
  - `start_time` = オプションでこの変数を設定して、ユーザーセッション追跡の時間帯の開始時刻を定義します。引数の形式は、次のとおりです。yyyy-mm-dd hh:mm:ss
  - `end_time` = オプションでこの変数を設定して、ユーザーセッション追跡の時間帯の終了時刻を定義します。引数の形式は、次のとおりです。yyyy-mm-dd hh:mm:ss
- 3 `output_file_name.xml` という名前のファイルの、XML アウトプットをレビューします。ユーザーセッション追跡の XML 出力の分析については、[235 ページの「ユーザーセッション追跡分析およびデータについて」](#)を参照してください。

Siebel ARM Analyzer Tool の実行について詳しくは、[220 ページの「Siebel ARM Analyzer Tool を使った Siebel ARM ファイルの分析」](#)を参照してください。

## Siebel ARM データの CSV 変換の実行

次の手順に従って、カンマ区切り値（CSV）形式の分析用アウトプットを取得します。

CSV 変換分析およびアウトプットについては、[237 ページの「Siebel ARM から CSV への変換データについて」](#)を参照してください。

### Siebel ARM データから CSV への変換分析の実行

**1** Siebel Server ルートディレクトリ内の bin サブディレクトリに移動します。

**2** 次のいずれかのコマンドを使って、Siebel ARM Analyzer Tool を実行します。

```
sarmanalyzer -o output_file_name.csv -d csv -f sarm_file_name.sarm
```

ここで、各項目は次のとおりです。

*output\_file\_name.csv* = CSV 出力ファイルの名前およびパス。

-d csv = Siebel ARM データの CSV 変換分析を指定します。

*sarm\_file\_name.sarm* = バイナリ Siebel ARM ファイルの名前およびパス。

**3** *output\_file\_name.csv* という名前のファイルの、CSV アウトプットをレビューします。CSV データの分析については、[237 ページの「Siebel ARM から CSV への変換データについて」](#)を参照してください。

**備考：** CSV 変換を実行すると、大きな出力ファイルが作成される場合があります。このファイルは、サードパーティ製のソフトウェアで読み取りができない場合があります。大きな Siebel ARM ファイルを分割するには、`-p` フラグを使用します。このフラグについては、[221 ページの表 8](#)を参照してください。

Siebel ARM Analyzer Tool の実行について詳しくは、[220 ページの「Siebel ARM Analyzer Tool を使った Siebel ARM ファイルの分析」](#)を参照してください。

## Siebel ARM Analyzer 出力ファイルについて

Siebel ARM Analyzer Tool を実行すると、Siebel ARM ファイルの変換タイプに応じて、XML (eXtensible Markup Language) 形式またはカンマ区切り値（CSV）形式のいずれかの出力ファイルが作成されます。

Siebel ARM ファイルの変換および Siebel ARM Analyzer Tool の実行については、[220 ページの「Siebel ARM Analyzer Tool を使った Siebel ARM ファイルの分析」](#)を参照してください。

さまざまなタイプの分析により得られる XML 出力ファイルを表示するには、XML エディタまたは Web ブラウザを使用します。タイミング測定値は XML タグ間に含まれます。

Siebel ARM ファイルから CSV ファイルへの変換により得られる出力ファイルを表示するには、スプレッドシートプログラムなどのサードパーティ製ソフトウェアを使用します。タイミング測定値のタグと値が含まれます。

Siebel ARM は、XML および CSV 出力の両方に含まれるすべてのタイミングをミリ秒単位で記録します。

## パフォーマンス集計分析およびデータについて

パフォーマンス集計分析は、Siebel ARM バイナリファイルに含まれるデータから編成されます。Siebel ARM ファイルでは、計測エリアに基づいてパフォーマンスデータがグループ化されています。

計測エリアおよびそのリストについては、[213 ページの「Siebel Application Response Measurement について」](#)を参照してください。

この形式の Siebel ARM 出力の作成については、[222 ページの「パフォーマンス集計分析の実行」](#)を参照してください。

Siebel ARM ファイルのパフォーマンス集計分析を実行すると、XML (eXtensible Markup Language) 出力ファイルが得られます。このファイルには、計測エリアのタイミングデータが含まれます。

パフォーマンス集計分析の XML 出力に含まれる情報量は、分析を実行する際に -a フラグに使用される引数 (AREA または DETAILS)、および SARM Granularity Level パラメータの設定によって異なります。このパラメータについては、[214 ページの「Siebel ARM パラメータおよび変数について」](#)を参照してください。

パフォーマンス集計の XML 出力ファイルには、-a フラグ引数が DETAILS に設定される場合は、次のタグスキーマが含まれます。分析の実行時に、-a フラグ引数が AREA に設定される場合、タグスキーマは <NumberOfSubAreas> および <SubArea> の情報がないことを除いて同じです。

```
<Area>
  <Name>
  <Symbol>
  <NumberOfSubAreas>
  <Invocations>
    <Recursive>
    <NonRecursive>
  <ResponseTime>
    <Total>
    <Average>
    <StandardDeviation>
    +<Maximum>
    +<Minimum>
  <ExecutionTime>
    <Total>
    <Calls>
    <Average>
    <Maximum>
    <Minimum>
    <PercentOfResponse>
  <RecursiveTime>
    <Total>
    <Calls>
    <Average>
    <Maximum>
    <Minimum>
    <PercentOfResponse>
  <InclusiveMemory>
    <Total>
    <Average>
    <StandardDeviation>
    +<MaxAllocated>
```

```

    +<MaxDeallocated>
<ExclusiveMemory>
  <Total>
  <Average>
  <StandardDeviation>
  +<MaxAllocated>
  +<MaxDeallocated>
<SubArea>
  <Name>
  <Symbol>
  <NumberOfInstances>
  +<Invocations>
  +<ResponseTime>
  +<ExecutionTime>
  +<Memory>
  +<Instance>
  +<Parents>
  +<Children>
<Parents>
  <NumberOfParents>
  <ParentArea>
    <Name>
    <Symbol>
    +<InvocationsFromParents>
    +<ResponseTime>
    +<Memory>
<Children>
  <NumberOfChildren>
  <ChildArea>
    <Name>
    <Symbol>
    +<InvocationsOfChild>
    +<ResponseTime>
    +<Memory>

```

各タグについては、[表 9](#) を参照してください。

表 9 パフォーマンス集計分析のタグ

タグ	説明
Area	Siebel ARM アーキテクチャの特定のエリアについて取り込まれたパフォーマンスデータを示します。1 つまたは複数のエリアのパフォーマンスデータを取り込むことができます。Siebel ARM のエリアについては、 <a href="#">213 ページの「Siebel Application Response Measurement について」</a> を参照してください。
Name	パフォーマンスデータを含むエリアの名前。エリア名のリストについては、 <a href="#">213 ページの「Siebel Application Response Measurement について」</a> を参照してください。
Symbol	パフォーマンスデータを含むエリアの記号。記号名のリストについては、 <a href="#">213 ページの「Siebel Application Response Measurement について」</a> を参照してください。

表 9 パフォーマンス集計分析のタグ

タグ	説明
NumberOfSubAreas	データを含むエリア内のサブエリア数。この数字は、特定の <Area> タグの下に表示される <SubArea> タグの数も示します。
Invocations	<p>監視期間にこのエリアが呼び出された回数。</p> <ul style="list-style-type: none"> <li>■ Recursive: Siebel ARM の重要な機能の 1 つに再帰処理機能があります。再帰的な呼び出しの例としては、あるワークフローステップが Application Object Manager (AOM) 関数を呼び出し、その関数がまた別のワークフローステップを起動するような場合があります。Siebel ARM では、ワークフロー層が呼び出された回数を表すために Recursive および NonRecursive の 2 つの基準を使用します。前述した例では、Recursive が 1 であり、NonRecursive もまた 1 です。応答時間の計算では、ルートレベルの呼び出し、つまり AOM 関数への最初のワークフロー呼び出しだけが考慮されます。実行時間の計算では、両方の呼び出しが考慮されます。</li> <li>■ Nonrecursive: 計測エリアが呼び出された回数。このタグは、ある 1 層で 1 つのリクエストに対応するまでにかかる時間を確認するのに役立ちます。</li> </ul>
ResponseTime	<p>他の子エリアへの呼び出しを含めた、計測エリア（層）の開始から終了までに、リクエストに費やされた時間を示します。これは、他の商用プロファイリングツールでは「インクルーシブタイム」とも呼ばれています。このエリアには、その他に次のようなタグが含まれます。</p> <ul style="list-style-type: none"> <li>■ Total: この計測エリア（層）でリクエストに費やされた合計時間。</li> <li>■ Average: リクエストに対する平均応答時間。</li> <li>■ StandardDeviation: このエリアのリクエスト時間の標準偏差値。</li> <li>■ +&lt;Maximum&gt;: このエリアのリクエストに費やされた最大時間。このタグを展開すると、この時間が費やされた特定の Siebel ARM ノードの詳細を確認できます。Siebel ARM ノードタグについては、<a href="#">232 ページの「呼び出しグラフの生成分析およびデータについて」</a>を参照してください。</li> <li>■ +&lt;Minimum&gt;: このエリアのリクエストに費やされた最小時間。このタグを展開すると、この時間が費やされた特定の Siebel ARM ノードの詳細を確認できます。Siebel ARM ノードタグについては、<a href="#">232 ページの「呼び出しグラフの生成分析およびデータについて」</a>を参照してください。</li> </ul>
ExecutionTime	<p>特定の計測エリアで費やされた合計時間を示します（子孫層で費やされた時間は含みません）。これは、他の商用プロファイリングツールでは「インクルーシブタイム」とも呼ばれています。このエリアには、その他に次のようなタグが含まれます。</p> <ul style="list-style-type: none"> <li>■ Total: 計測エリア（層）の開始から終了までにリクエストに費やされた合計時間。</li> <li>■ Calls: 再帰的な呼び出しおよび非再帰的な呼び出しを含めた総呼び出し数。</li> <li>■ Average: 計測エリア（層）の開始から終了までにリクエストに費やされた平均時間。</li> <li>■ Maximum: 計測エリア（層）の開始から終了までにリクエストに費やされた最大時間。</li> <li>■ Minimum: 計測エリア（層）の開始から終了までにリクエストに費やされた最小時間。</li> <li>■ PercentageofResponse: このエリアで費やされた合計応答時間の割合。</li> </ul>

表 9 パフォーマンス集計分析のタグ

タグ	説明
RecursiveTime	<p>このエリア内で再帰的な呼び出しに費やされた合計時間を示します。すなわち、それ自体を呼び出す場合にこのエリア内で費やされた時間です。</p> <p>このエリアには、その他に次のようなタグが含まれます。</p> <ul style="list-style-type: none"> <li>■ Total : 再帰的なリクエストに費やされた合計時間。</li> <li>■ Calls : 再帰的な呼び出し数。</li> <li>■ Average : 再帰的なリクエストに費やされた平均時間。</li> <li>■ Maximum : 再帰的なリクエストによって費やされた最大時間。</li> <li>■ Minimum : 再帰的なリクエストによって費やされた最小時間。</li> <li>■ PercentageofResponse : このエリアで再帰的に費やされた合計応答時間の割合。</li> </ul>
InclusiveMemory	<p>このエリア、および子または子孫エリアを開始するリクエストによって使用されたメモリ量を示します。メモリ値はバイト単位で記録されます。</p> <p>このエリアには、その他に次のようなタグが含まれます。</p> <ul style="list-style-type: none"> <li>■ Total : このエリアのリクエストによる合計メモリ使用量。</li> <li>■ Average : このエリアのリクエストによる平均メモリ使用量。</li> <li>■ StandardDeviation : このエリアのメモリ使用量の標準偏差値。</li> <li>■ +&lt;MaxAllocated&gt; : このタグを展開すると、最大メモリが割り当てられた Siebel ARM ノードの詳細データを表示します。</li> <li>■ +&lt;MaxDeallocated&gt; : このタグを展開すると、メモリが割り当て解除された Siebel ARM ノードの詳細データを表示します。</li> </ul> <p><b>備考:</b> Siebel ARM ノードタグについては、232 ページの「呼び出しグラフの生成分析およびデータについて」を参照してください。</p>
ExclusiveMemory	<p>このエリアだけを開始するリクエストによって使用されたメモリ量を示します。メモリ値はバイト単位で記録されます。</p> <p>このエリアには、その他に次のようなタグが含まれます。</p> <ul style="list-style-type: none"> <li>■ Total : このエリアのリクエストによる合計メモリ使用量。</li> <li>■ Average : このエリアのリクエストによる平均メモリ使用量。</li> <li>■ StandardDeviation : このエリアのメモリ使用量の標準偏差値。</li> <li>■ +&lt;MaxAllocated&gt; : このタグを展開すると、最大メモリが割り当てられた Siebel ARM ノードの詳細データを表示します。</li> <li>■ +&lt;MaxDeallocated&gt; : このタグを展開すると、メモリが割り当て解除された Siebel ARM ノードの詳細データを表示します。</li> </ul> <p><b>備考:</b> Siebel ARM ノードタグについては、232 ページの「呼び出しグラフの生成分析およびデータについて」を参照してください。</p>

表 9 パフォーマンス集計分析のタグ

タグ	説明
SubArea	<p>指定エリアの特定のサブエリアについて取り込まれたパフォーマンスデータを示します。指定エリアの下、1 つまたは複数のサブエリアのパフォーマンスデータを取り込むことができます。</p> <ul style="list-style-type: none"> <li>■ Name : パフォーマンスデータを含むサブエリアの名前。</li> <li>■ Symbol : パフォーマンスデータを含むサブエリアの記号。</li> <li>■ NumberOfInstances : データを含むサブエリア内のインスタンス数。この数字は、特定の &lt;SubArea&gt; タグの下に表示される &lt;Instance&gt; タグの数も示します。インスタンスはサブエリアを定義する詳細レベルです。</li> <li>■ Invocations : 監視期間にこのサブエリアが呼び出された回数。</li> <li>■ +&lt;ResponseTime&gt; : サブエリアの開始から終了までにリクエストに費やされた時間を示します。このタグを展開すると、タイミングの詳細を確認できます。これらのタグは、エリア ResponseTime タグに定義されているタグと同じです。</li> <li>■ +&lt;ExecutionTime&gt; : サブエリアに費やされた時間を示します。このタグを展開すると、タイミングの詳細を確認できます。これらのタグは、エリア ExecutionTime タグに定義されているタグと同じです。</li> <li>■ +&lt;InclusiveMemory&gt; : このサブエリア、および子または子孫エリアを開始するリクエストによって使用されたメモリ量を示します。メモリ値はバイト単位で記録されます。このタグを展開すると、メモリの詳細を確認できます。これらの展開されたタグは、エリア InclusiveMemory タグに定義されているタグと同じです。</li> <li>■ +&lt;ExclusiveMemory&gt; : このサブエリアだけを開始するリクエストによって使用されたメモリ量を示します。メモリ値はバイト単位で記録されます。このタグを展開すると、メモリの詳細を確認できます。これらの展開されたタグは、エリア ExclusiveMemory タグに定義されているタグと同じです。</li> <li>■ +&lt;Instance&gt; : インスタンスはサブエリアを定義する別の詳細レベルです。このタグを展開すると、インスタンスの詳細を確認できます。これらのタグは、エリアタグに定義されているタグと同じです。</li> <li>■ +&lt;Parents&gt; : サブエリアの親、つまりサブエリアを呼び出したエリアを示します。このタグを展開すると、親のサブエリアの詳細を確認できます。これらのタグは、エリア Parents タグに定義されているタグと同じです。</li> <li>■ +&lt;Children&gt; : サブエリアの子、つまりサブエリアによって呼び出されたエリアを示します。このタグを展開すると、親のサブエリアの詳細を確認できます。これらのタグは、エリア Children タグに定義されているタグと同じです。</li> </ul>

表 9 パフォーマンス集計分析のタグ

タグ	説明
Parents	<p>サブエリアの親、つまり指定エリアを呼び出したエリアを示します。この情報により、呼び出し元またはエリアの呼び出し元と、エリアがその親の応答時間に加算した合計時間と呼び出し回数を確認できます。このエリアには、その他に次のようなタグが含まれます。</p> <ul style="list-style-type: none"> <li>■ NumberOfParents：指定エリアを呼び出す親エリア数。</li> <li>■ ParentArea：Siebel ARM アーキテクチャの特定の親エリアについて取り込まれたパフォーマンスデータを示します。1 つまたは複数の親エリアのパフォーマンスデータを取り込むことができます。</li> <li>■ Name：指定エリアを呼び出す親エリアの名前。</li> <li>■ Symbol：指定エリアを呼び出す親エリアの記号。</li> <li>■ +&lt;InvocationsFromParents&gt;：指定エリアが親エリアによって呼び出された回数。このタグを展開すると、タイミングの詳細を確認できます。</li> <li>■ +&lt;ResponseTime&gt;：親エリアの開始から終了までにリクエストに費やされた時間を示します。このタグを展開すると、親エリアの応答時間の詳細を確認できます。</li> <li>■ +&lt;Memory&gt;：親エリアによって使用されたメモリ量を示します。このタグを展開すると、親のサブエリアの詳細を確認できます。</li> </ul>
Children	<p>親エリアによって呼び出されたエリア、つまり指定エリアによって呼び出されたエリアを示します。エリアの子情報を展開すると、各子の応答時間の内訳がわかります。このエリアには、その他に次のようなタグが含まれます。</p> <ul style="list-style-type: none"> <li>■ NumberOfChildren：指定エリアによって呼び出された子エリア数。</li> <li>■ ChildArea：Siebel ARM アーキテクチャの特定の子エリアについて取り込まれたパフォーマンスデータを示します。1 つまたは複数の子エリアのパフォーマンスデータを取り込むことができます。</li> <li>■ Name：指定エリアによって呼び出された子エリアの名前。</li> <li>■ Symbol：指定エリアによって呼び出された子エリアの記号。</li> <li>■ +&lt;InvocationsOfChild&gt;：子エリアが指定エリアによって呼び出された回数。このタグを展開すると、タイミングの詳細を確認できます。</li> <li>■ +&lt;ResponseTime&gt;：子エリアの開始から終了までにリクエストに費やされた時間を示します。このタグを展開すると、子エリアの応答時間の詳細を確認できます。</li> <li>■ +&lt;Memory&gt;：子エリアによって使用されたメモリ量を示します。このタグを展開すると、子サブエリアの詳細を確認できます。</li> </ul>

## 呼び出しグラフの生成分析およびデータについて

呼び出しグラフの生成分析では、呼び出し参照のマップが生成されます。呼び出しマップ内の各ノードは、計測インスタンス（つまり計測エリアを通じた各リクエストの応答時間）を表します。

計測エリアについては、[213 ページの「Siebel Application Response Measurement について」](#)を参照してください。

この形式の Siebel ARM 出力の作成については、[223 ページの「呼び出しグラフの生成の実行」](#)を参照してください。

Siebel ARM ファイルの呼び出しグラフの生成分析を実行すると、XML (eXtensible Markup Language) 出力ファイルが得られます。Siebel ARM Analyzer Tool は、指定された Siebel ARM ファイルに関する呼び出し参照付きのマップを作成します。この呼び出しマップ内の各ノードは、計測インスタンスを表します。このオプションを使用すると、応答時間データを取り込む各コンポーネントによって行われたすべての呼び出しについて、その情報を含む XML ファイルが生成されます。

この XML 出力ファイルには、呼び出しの詳細を記録する次のタグスキーマが含まれます。各タグについては、[表 10](#)を参照してください。

```
<SarmNode>
  <SarmID>
  <TypeLevel>
  <RootID>
  <ParentSARMID>
  <ParentTimeID>
  <ParentProcID>
  <AreaCodeSymbol>
  <AreaDescription>
  <SubAreaCodeSymbol>
  <SubAreaDescription>
  <Count>
  <Duration>
  <PooledMemoryUsage>
  <PooledMemoryCalls>
  <SystemMemoryUsage>
  <SystemMemoryCalls>
  <AppInt1>
  <AppInt2>
  <AppString1>
  <AppString2>
  +<ChildNode>
</SarmNode>
```

表 10 呼び出しグラフの生成分析のタグ

タグ	説明
SarmNode	このタグ内に含まれるデータは、Siebel ARM ノードのインスタンスを表します。このノードは、Siebel ARM アーキテクチャの計測エリアです。各 Siebel ARM ノードは、多数の子孫ノードを持つこともできますし、子孫ノードを持たなくてもかまいません。
SarmID	Siebel ARM ノードを表す固有の番号。
TypeLevel	Siebel ARM が Siebel ARM ノード情報を記録する粒度レベル。粒度レベルについては、214 ページの「 <a href="#">Siebel ARM パラメータおよび変数について</a> 」を参照してください。
RootID	ルート Siebel ARM ノードの SarmID。
ParentSARMID	リクエストの移動元の親 SarmNode。
ParentTimeID	対応する親 Siebel ARM ノードの開始時刻から生成される固有の ID 番号。
ParentProcID	Siebel コンポーネントの親プロセス ID。これは、OS（オペレーティングシステム）のプロセス ID です。
AreaCodeSymbol	Siebel アーキテクチャ内の計測エリアの記号。Siebel アーキテクチャエリアについては、213 ページの「 <a href="#">Siebel Application Response Measurement について</a> 」を参照してください。
AreaDescription	Siebel アーキテクチャ内の計測エリアの名前。Siebel アーキテクチャエリアについては、213 ページの「 <a href="#">Siebel Application Response Measurement について</a> 」を参照してください。
SubAreaCodeSymbol	Siebel アーキテクチャのエリア内のサブエリアの記号。Siebel アーキテクチャエリアについては、213 ページの「 <a href="#">Siebel Application Response Measurement について</a> 」を参照してください。
SubAreaDescription	Siebel アーキテクチャのエリア内のサブエリアの名前。Siebel アーキテクチャエリアについては、213 ページの「 <a href="#">Siebel Application Response Measurement について</a> 」を参照してください。
Count	Siebel ARM がこの Siebel ARM ノードにアクセスした回数。

表 10 呼び出しグラフの生成分析のタグ

タグ	説明
SubArea	Siebel アーキテクチャのエリア内の詳細な計測。たとえば Siebel ARM は、メソッドの起動 (Invoke Method) やワークフロー実行内の 1 ステップの実行 (Step Execution) にかかる応答時間を取り込みます。
StartTime	Siebel ARM レコードのタイムスタンプの内部表現。
Duration	計測エリアの実行にかかった合計時間。
PooledMemoryUsage	Siebel High Performance Memory Allocator において消費または解放されたメモリ量。
PooledMemoryCalls	High Performance Memory Allocator に対して実行された呼び出しの回数。
SystemMemoryUsage	オペレーティングシステムにおいて消費または解放されたメモリ量。
SystemMemoryCalls	オペレーティングシステムに対して実行された呼び出しの回数。
UserInt1	計測ポイントで取り込まれたコンテキスト情報。この値は、計測エリアに応じて異なります。
UserInt2	計測ポイントで取り込まれたコンテキスト情報。この値は、計測エリアに応じて異なります。
UserString	計測ポイントで取り込まれたコンテキスト情報。この値は、計測エリアに応じて異なります。たとえば、起動されたメソッドまたは初期化されたワークフロープロセスの名前などです。
AppInt1 および AppInt2	計測ポイントで取り込まれたコンテキスト整数値。この値は、計測エリアに応じて異なります。
AppString1 および AppString2	計測ポイントで取り込まれたコンテキスト文字列値。この値は、計測エリアに応じて異なります。たとえば、起動されたメソッドまたは初期化されたワークフロープロセスの名前などです。
+<ChildNode>	このタグを展開すると、指定ノードの子孫ノードに関するパフォーマンスの詳細を表示します。子孫ノードは、親ノードと同様に定義されます。つまりタグ定義は前述のものと同じです。

## ユーザーセッション追跡分析およびデータについて

WebサーバーおよびSiebel ServerからSiebel ARMファイルを使用するユーザーセッション追跡分析を実行すると、XML (eXtensible Markup Language) 出力ファイルが得られます。この XML 出力ファイルには、Siebel ARM ファイルの変換の実行時に指定したユーザーによる各 SWE リクエストの詳細な情報が含まれています。

ユーザーがシステムに複数回ログオンした場合、この出力には複数のセッションが示されます。SWE リクエストは、ログインセッションごとにグループ化され、リクエストが行われた時間によりソートされます。Siebel ARM アーキテクチャについては、[213 ページの「Siebel Application Response Measurement について」](#)を参照してください。

この形式の Siebel ARM 出力の作成については、[224 ページの「ユーザーセッション追跡の実行」](#)を参照してください。

この XML 出力ファイルには、ユーザーセッション追跡の詳細を記録する次のタグスキーマが含まれます。またユーザーセッション追跡データには、パフォーマンス集計分析のタグスキーマが含まれます。

これらのタグについては、[226 ページの「パフォーマンス集計分析およびデータについて」](#)を参照してください。

```
<UserID>
  <Session>
    <SessionID>
      <UserActionID>
        <ID>
          <SWERequest>
            <ReqID>
              <TotalServerTime>
                <WebServerTime>
                  <NetworkTime>
                    <SiebServerTime>
                      <DatabaseTime>
                        <DatabaseCalls>
                          +<SiebsrvrDetail>
```

ユーザーセッションの追跡分析に固有のタグについては、[表 11](#) を参照してください。また、パフォーマンス集計分析にも含まれるタグについては、[227 ページの表 9](#) を参照してください。

表 11 ユーザーセッション追跡のタグ

タグ	説明
UserID	ユーザーログイン名。たとえば、「SADMIN」です。
Session	このタグ内に含まれる特定のユーザーセッションについて取り込まれたパフォーマンスデータを示します。
SessionID	固有のユーザーセッション ID を 16 進形式で示します。このセッション ID の最初の要素はサーバー ID、第 2 の要素はプロセス ID、最後の要素はタスク ID です。次に例を示します。  !1.2b40.182b  サーバー ID = !1  プロセス ID = 2b40 (2b40 は、10 進形式の 11072 であり、オペレーティングシステムのプロセス ID 番号を示します)。  タスク ID = 182b (182b は、10 進形式で 6187 であり、タスク ID 番号を示します)。
UserActionID	このタグ内に含まれるデータは、特定ユーザーの個別アクションまたはリクエストを示します。
ID	特定のユーザーセッションについて、特定ユーザーのアクションまたはリクエストを順番に識別する番号。
SWERequest	特定のユーザーアクションまたはリクエストのパフォーマンスタイミングデータを示します。
ReqID	Siebel Web Server Extension (SWSE) プラグインリクエストに応じて増加する数値の ID 番号。
TotalServerTime	サーバー上での合計リクエスト時間 (Web サーバー、Siebel Server、およびネットワークの時間を含みます)。
WebServerTime	ある 1 つのリクエストのために Web サーバー上で費やされた合計時間。
NetworkTime	Web サーバーと Siebel Server 間で費やされた合計時間。またこの時間には、リクエストを処理先の Siebel Server タスクにルーティングする若干の Siebel インフラストラクチャの時間が含まれる場合があります。
SiebServerTime	Siebel Server 上で費やされた時間。
DatabaseTime	データベースとの通信時にネットワーク上で費やされた時間。
DatabaseCalls	Siebel Server データベースのコネクタ層に対する呼び出し回数。
+<SiebsrvrDetail>	ある 1 つのセッションの計測に関する各アーキテクチャエリアの応答時間および実行時間。これらのタグについては、226 ページの「パフォーマンス集計分析およびデータについて」および 232 ページの「呼び出しグラフの生成分析およびデータについて」を参照してください。

## Siebel ARM から CSV への変換データについて

CSV 形式は、解釈または集計が適用されていないカンマ区切り形式のファイルです。この CSV ファイルには、カラム見出しの下に整理されたデータが含まれます。この出力を表示するには、スプレッドシートなどのサードパーティ製ソフトウェアツールを使用します。

この形式の Oracle の Siebel ARM 出力の作成については、225 ページの「Siebel ARM データの CSV 変換の実行」を参照してください。

これらのカラム見出しのリストとその説明については、232 ページの「呼び出しグラフの生成分析およびデータについて」で呼び出しグラフの分析に関するタグの定義を参照してください。これらのカラムごとに情報をレビューおよび整理できます。CSV データの例については、図 4 を参照してください。

	B	C	D	E	F	G	H	I	J	K	L
	ThreadID	IsRoot	Type(level)	RootID	ParentSarmID	ParentTimeID	ParentProcID	AreaCodeSymbol	AreaDesc	SubAreaCodeSymbol	SubAreaDesc
1											
2	5300	N	Sarm(1)	1	1	1074205585	1848	Area_SWSE	Web Server Plugin	Sub_SWSE_SENDMSG	Send message to app se
3	5300	N	Sarm(1)	1	1	1074205585	1848	Area_SWSE	Web Server Plugin	Sub_SWSE_SENDMSG	Send message to app se
4	5300	N	Sarm(1)	1	1	1074205585	1848	Area_SWSE	Web Server Plugin	Sub_SWSE_SENDMSG	Send message to app se
5	5300	N	Sarm(1)	1	1	1074205585	1848	Area_SWSE	Web Server Plugin	Sub_SWSE_SENDMSG	Send message to app se
6	5300	N	Sarm(1)	1	1	1074205585	1848	Area_SWSE	Web Server Plugin	Sub_SWSE_SENDMSG	Send message to app se
7	5300	N	Sarm(1)	1	1	1074205585	1848	Area_SWSE	Web Server Plugin	Sub_SWSE_SENDMSG	Send message to app se
8	5300	N	Sarm(1)	1	1	1074205585	1848	Area_SWSE	Web Server Plugin	Sub_SWSE_SENDMSG	Send message to app se
9	5300	N	Sarm(1)	1	1	1074205585	1848	Area_SWSE	Web Server Plugin	Sub_SWSE_SENDMSG	Send message to app se
10	5300	N	Sarm(1)	1	1	1074205585	1848	Area_SWSE	Web Server Plugin	Sub_SWSE_SENDMSG	Send message to app se
11	5300	N	Sarm(1)	1	1	1074205585	1848	Area_SWSE	Web Server Plugin	Sub_SWSE_SENDMSG	Send message to app se
12	5300	N	Sarm(1)	1	1	1074205585	1848	Area_SWSE	Web Server Plugin	Sub_SWSE_SENDMSG	Send message to app se
13	5300	N	Sarm(1)	1	1	1074205585	1848	Area_SWSE	Web Server Plugin	Sub_SWSE_SENDMSG	Send message to app se
14	5300	N	Sarm(1)	1	1	1074205585	1848	Area_SWSE	Web Server Plugin	Sub_SWSE_SENDMSG	Send message to app se
15	5300	N	Sarm(1)	1	1	1074205585	1848	Area_SWSE	Web Server Plugin	Sub_SWSE_SENDMSG	Send message to app se
16	5300	N	Sarm(1)	1	1	1074205585	1848	Area_SWSE	Web Server Plugin	Sub_SWSE_SENDMSG	Send message to app se
17	5300	N	Sarm(1)	1	1	1074205585	1848	Area_SWSE	Web Server Plugin	Sub_SWSE_SENDMSG	Send message to app se
18	5300	N	Detail(2)	1	1	1074205585	1848	Area_SARM	SARM Framework	Sub_SARM_IO	Flush SARM Buffer To Di
19	5300	N	Sarm(1)	1	1	1074205585	1848	Area_SWSE	Web Server Plugin	Sub_SWSE_SENDMSG	Send message to app se
20	5300	N	Sarm(1)	1	1	1074205585	1848	Area_SWSE	Web Server Plugin	Sub_SWSE_SENDMSG	Send message to app se

図 4 CSV データの例

## Siebel ARM Query Tool を使った Siebel ARM ファイルの分析

ここでは、Siebel ARM Query Tool を使って Siebel ARM ファイルを分析する方法について説明します。このツールは、バイナリ Siebel ARM ファイルを分析のために判読可能なアウトプットに変換します。

次の項では、Siebel ARM Query Tool の使用方法について説明します。

- 238 ページの「Siebel ARM Query Tool について」
- 239 ページの「Siebel ARM Query Tool の一般的なコマンド」
- 240 ページの「Siebel ARM Query Tool の設定」
- 241 ページの「Siebel ARM Query Tool の入力の設定」
- 242 ページの「Siebel ARM Query Tool の出力の設定」
- 245 ページの「Siebel ARM Query Tool でのフィルタの使用」
- 254 ページの「Siebel ARM Query Tool による Siebel ARM データの集計」
- 256 ページの「Siebel ARM Query Tool を使用したヒストグラムの生成」
- 257 ページの「Siebel ARM Query Tool でのマクロの使用」

### Siebel ARM Query Tool について

Siebel ARM Query Tool は、Siebel Server によって生成されたバイナリ Siebel ARM データを処理する、パフォーマンス分析コマンドラインツールです。次の点で Siebel ARM Analyzer Tool と異なります。

#### ■ 有用性

Siebel ARM Query は、より複雑なクエリーを作成可能な、多くのコマンドラインオプションを提供します。

#### ■ パフォーマンス

Siebel ARM Query Tool は、Siebel ARM Analyzer Tool よりも高速にデータを処理します。

#### ■ メモリ管理

結果の統計情報の精度を上げるために、Siebel ARM Query Tool が使用するメモリ量を増やすことができます。デフォルトでは、Siebel ARM Query Tool は、約 20 MB のメモリを使用します。これは、約 1% の統計エラーに相当します。Siebel ARM Query Tool が使用するメモリ量を 500 MB に増やすと、統計エラーを 0.2% に減らすことができます。

割り当てるメモリの量は、Siebel ARM Query Tool のパフォーマンスに影響を与えません。割り当てる追加のメモリは、Siebel ARM Query Tool が統計の計算に使用するバッファに適用されます。

**備考：** 統計エラーを完全に排除することは不可能です。

Siebel ARM Query Tool が統計情報の精度に使用するメモリ量を変更するためのパラメータについては、[240 ページの「Siebel ARM Query Tool の設定」](#)を参照してください。

■ ソースデータの指定

Siebel ARM Query Tool の入力として、個々の Siebel ARM ファイルまたはディレクトリを指定できます。入力データを指定するパラメータについては、[241 ページの「Siebel ARM Query Tool の入力の設定」](#)を参照してください。

■ データのフィルタ機能

Siebel ARM Query Tool の分析に含めるか、または除外するデータをフィルタにより指定できます。

■ 集計データ

集計の順序を指定できます。これにはロールアップ計算も含まれます。集計については、[254 ページの「Siebel ARM Query Tool による Siebel ARM データの集計」](#)を参照してください。

■ 複数の出力形式

複数の出力タイプを同時に生成することを指定できます。サポートされる出力形式は、.TXT、.XML、および .CSV です。出力形式の指定方法については、[242 ページの「Siebel ARM Query Tool の出力の設定」](#)を参照してください。

■ マクロ言語

Siebel ARM Query Tool は、マクロの使用をサポートしています。詳しくは、[257 ページの「Siebel ARM Query Tool でのマクロの使用」](#)を参照してください。

## Siebel ARM Query Tool の一般的なコマンド

この項では、Siebel ARM Query Tool で使用する一般的なオプションについて説明します。オンラインヘルプを表示するコマンド、およびコマンドウィンドウに進捗情報を表示するコマンドも説明します。[表 12](#) では、これらのコマンドについて説明します。

表 12 Siebel ARM Query Tool の一般的なフラグ

フラグ	説明
-help	ヘルプを表示します。
-copyright	Siebel ARM Query Tool に関する著作権情報を表示します。
-tips	一般的な集計、レポート、および変換を実行するためのコマンドライン構文を表示します。
-macrosyntax	Siebel ARM Query Tool は、マクロ言語をサポートしています。このフラグは、マクロ言語の構文を表示します。
-planonly	クエリーを実行せずに、クエリーの実行計画を表示します。
-quiet	出力コンソールに結果のみを表示する場合に、このフラグを使用します。このフラグを指定すると、進捗情報は表示されません。次の例のようにして、進捗情報をファイルに保存することを指定できます。  <pre>&gt; sarmquery -output verbose=verbose.txt -input data.sarm -aggregate time=1</pre> この場合、進捗情報はファイル verbose.txt に保存されます。

## Siebel ARM Query Tool の設定

この項では、Siebel ARM Query Tool の設定方法について説明します。Siebel ARM Query Tool の設定変更の一例は、Siebel ARM クエリーが使用するメモリ量を変更して統計情報の精度を向上させることです。表 13 では、Siebel ARM Query Tool を設定するために使用可能なフラグについて説明します。表 13 に説明のあるすべてのオプションの前には、オプション `-config` を付けます。

表 13 Siebel ARM Query Tool の設定オプション

フラグ	説明
<code>file=&lt;macro_file_name&gt;</code>	マクロを含むファイルを指定します。
<code>macro=macro_name(str)</code>	マクロを実行して、マクロへの最初の引数として文字列 <i>str</i> を渡します。このフラグを使用するには、フラグを使用するマクロを含むファイルを指定する必要があります。 <code>-config file=&lt;macro_file_name&gt;</code>
<code>gmt=0</code>	すべてのタイムスタンプ（コマンドライン上）を解析し、コマンドウィンドウにグリニッジ標準時（GMT）ですべての時刻を表示します。このフラグを指定しない場合、Siebel ARM Query Tool はローカルの時刻を使用します。
<code>gmt=[+-]HHMM</code>	すべての時刻を GMT からのオフセットとして解析およびレポートします。オフセットは、HHMM の表記で指定します。ここで、「HH」は時間（00-23）、および「MM」は分（00-59）です。 たとえば、太平洋標準時（PST）をレポートするには、次のフラグを使用します。 <code>&gt; sarmquery -config gmt=-0800</code>
<code>datalimit=&lt;limit&gt;</code>	このフラグを使用することにより、戻される最大レコード数を指定します。 <code>&lt;limit&gt;</code> に最大数を指定します。最大数を超える場合、エラーメッセージ（データ制限の超過）が表示され、Siebel ARM Query Tool は終了します。
<code>timelimit=&lt;seconds&gt;</code>	このフラグを使用することにより、Siebel ARM Query Tool が実行可能な最大秒数を指定します。最大秒数が経過すると、Siebel ARM Query Tool は終了します。 Siebel ARM Query Tool は、次の 2 つのフェーズで処理します。 <ul style="list-style-type: none"> <li>■ ファイル収集フェーズ ツールは必要なすべてのファイルを識別します。timelimit フラグに指定する値は、このフェーズに影響を与えません。このフェーズで Siebel ARM Query Tool が費やす時間には制限はありません。</li> <li>■ データ処理フェーズ timelimit フラグに指定する値は、このフェーズの長さに影響を与えます。</li> </ul>

表 13 Siebel ARM Query Tool の設定オプション

フラグ	説明
memlimit=<MB>	<p>このフラグを使用することにより、Siebel ARM Query Tool が集計の統計情報を計算するのに使用する内部バッファに割り当てることができるメモリ量を指定します。</p> <p>内部バッファで使用できるメモリ量が多いほど、返される統計の精度が向上します。このフラグのデフォルト値は、20 メガバイト (MB) です。5 MB ~ 500 MB の値を割り当てることができます。</p> <p>指定する値は、パフォーマンスには影響を与えません。Siebel ARM クエリーの統計の計算に使用するバッファにのみ適用されます。</p>

## Siebel ARM Query Tool の入力の設定

この項では、Siebel ARM Query Tool が使用する Siebel ARM ファイルの指定方法について説明します。Siebel ARM Query Tool は、バイナリ Siebel ARM ファイルを分析のために判読可能なアウトプットに変換します。表 14 に、使用可能な入力オプションを示します。

表 14 に説明のあるすべてのオプションの前には、オプション `-input` を付けます。

表 14 Siebel ARM Query Tool の入力フラグ

フラグ	説明
<sarmfile>	バイナリ Siebel ARM ファイル (.sarm) を指定します。
<directory>	Siebel ARM ファイルが格納されているディレクトリを指定します。Siebel ARM Query Tool は、指定されたディレクトリで見つかる、すべての Siebel ARM ファイルを処理します。
stdin	<p>標準入力から Siebel ARM ファイル名のリストを読み取るように、Siebel ARM クエリーに指示するリテラルキーワード。1 行に 1 つの Siebel ARM ファイルまたはディレクトリ名を指定できます。有効な入力の例は次のとおりです。</p> <pre>&gt; sarmquery . &gt; sarmquery -input d:¥sarmdata &gt; sarmquery d:¥sarmdata¥srvr1 d:¥sarmdata¥srvr2 sarmfile1.data &gt; dir /s /b *.sarm   sarmquery -input stdin</pre>

## Siebel ARM Query Tool の出力の設定

この項では、Siebel ARM Query Tool を使用する際に利用可能な出力オプションについて説明します。Siebel ARM Query Tool は、.TXT、.XML、および .CSV 形式のファイルに、一度に出力することができます。Siebel ARM Query Tool を使用して、次に例に示すように異なるファイルに異なる種類の情報を書き込むことができます。表 15 では、出力の設定に使用可能なフラグを説明しています。

```
> sarmquery -i d:¥sarmdata -sel subarea=infra_entry -sel resp=1000 -out header=hdr.csv
-out sarm=sarm.csv -aggregate area -out agg=agg.xml
```

このコマンドは以下を書き込みます。

- ファイル `hdr.csv` には、Siebel ARM ファイルのヘッダーメタデータに関する情報
- ファイル `sarm.csv` には、Siebel ARM データ
- ファイル `agg.xml` には、エリアに関する集計情報

ファイルに書き込む最大行数も指定できます。ファイルが指定された最大行数を含む場合、Siebel ARM Query Tool は、同じファイル名に *N* を付加した新しいファイルを作成します。ここで、*N* は数値を表します。次の例は、このオプションの使用について説明しています。

```
> sarmquery -output sarm=sarmdata.csv#20000
```

このコマンドは、Siebel ARM データをカンマ区切り値 (CSV) 形式で `sarmdata.csv` という名前のファイルに書き込みます。オプション値 `#20000` により、`sarmdata.csv` には最大 20000 行を書き込みます。`sarmdata.csv` の行数が 20000 行になった場合、Siebel ARM Query Tool は、`sarmdata_0002.csv` という名前の新しいファイルに書き込み、すべてのデータが出力されるまで、同様の処理を繰り返します。

**備考：** 表 15 のすべてのフラグの前には、コマンドオプション `-output` を付けます。

表 15 Siebel ARM Query Tool の出力フラグ

フラグ	説明
<code>fdr=&lt;filename&gt;</code>	すべての <code>fdr</code> ファイル ( <code>-input</code> を使用して指定) を CSV 形式に変換し、それを指定されたファイルに書き込みます。
<code>fdrhdr=&lt;filename&gt;</code>	すべての <code>fdr</code> ヘッダーを CSV に変換し、それを指定されたファイルに書き込みます。
<code>error=&lt;filename&gt;</code>	デフォルトでは、Siebel ARM Query Tool は出力を <code>stderr.txt</code> に書き込みます。このコマンドはエラーメッセージを指定のファイルにリダイレクトします。
<code>debug=&lt;filename&gt;</code>	デフォルトでは、Siebel ARM クエリーはデバッグメッセージを書き込みません。このコマンドにより、Siebel ARM Query Tool は、指定されたファイルへのデバッグメッセージの書き込みを有効にします。

表 15 Siebel ARM Query Tool の出力フラグ

フラグ	説明
dbglines=<filename>	このコマンドにより、Siebel ARM Query Tool は、より多くのデバッグ情報の書き込みを有効にします。
verbose=<filename>	デフォルトでは、Siebel ARM Query Tool は、ファイル stderr.txt に書き込みます。このコマンドは指定されたファイルに詳細な出力をリダイレクトします。
map=<filename>	Siebel ARM のすべてのエリアとサブエリアをリストするファイルを作成します。
header=<filename> appheader=<filename> webheader=<filename> cliheader=<filename>	<p>使用可能なフラグは次のとおりです。</p> <ul style="list-style-type: none"> <li>■ header すべてのヘッダー情報を指定のファイルに書き込みます。</li> <li>■ appheader Siebel Server によって生成された Siebel ARM ファイルのヘッダー情報を書き込みます。</li> <li>■ webheader Siebel Web Server によって生成された Siebel ARM ファイルのヘッダー情報を書き込みます。</li> <li>■ cliheader クライアントによって生成された Siebel ARM ファイルのヘッダー情報を書き込みます。</li> </ul> <p>次の例では、すべてのヘッダーに関する情報を all.csv に書き込み、各特定部分の情報を異なるファイルに書き込みます。</p> <pre>&gt; sarmquery -i dir -o header=all.csv -o appheader=app.csv -o webheader=web.csv -o cliheader=cli.csv</pre>

表 15 Siebel ARM Query Tool の出力フラグ

フラグ	説明
<p>sarm=&lt;filename&gt; appsarm=&lt;filename&gt; websarm=&lt;filename&gt; clisarm=&lt;filename&gt;</p>	<p>使用可能なフラグは次のとおりです。</p> <ul style="list-style-type: none"> <li>■ sarm すべての Siebel ARM データを指定のファイルに書き込みます。</li> <li>■ appsarm Siebel Server によって生成されたすべての Siebel ARM データを指定のファイルに書き込みます。</li> <li>■ websarm Siebel Web Server (SWSE) によって生成されたすべての Siebel ARM データを指定のファイルに書き込みます。</li> <li>■ clisarm クライアントによって生成されたすべての Siebel ARM データを書き込みます。</li> </ul> <p>次の例では、Siebel Server と SWSE によって生成された Siebel ARM データを別々のファイルに書き込みます。</p> <pre>&gt; sarmquery -output appsarm=app.csv -output websarm=web.csv</pre>
<p>agg=&lt;filename&gt; iagg=&lt;filename&gt; sagg=&lt;filename&gt;</p>	<p>使用可能なフラグは次のとおりです。</p> <ul style="list-style-type: none"> <li>■ agg 指定のファイルに集計レポートを書き込みます。</li> <li>■ sagg 指定のファイルに集計レポートのサブセットを書き込みます。このフラグは排他的な基準を指定します。</li> <li>■ iagg 指定のファイルに集計レポートのサブセットを書き込みます。このフラグは包括的な基準を指定します。</li> </ul> <p>このオプションを使用するには、フラグ <code>-aggregate</code> を使用して、Siebel ARM データを集計している必要があります。</p>

表 15 Siebel ARM Query Tool の出力フラグ

フラグ	説明
avginclresp=<filename> pctcount=<filename> pctinclresp=<filename> pctselftime=<filename>	<p>使用可能なフラグは次のとおりです。</p> <ul style="list-style-type: none"> <li>■ avginclresp 平均応答時間を指定のファイルに書き込みます。</li> <li>■ pctcount 指定の時間内に完了するサーバーリクエストの割合を指定のファイルに書き込みます。  次の例は、100 ミリ秒内、100 ミリ秒～ 500 ミリ秒間などに完了するサーバーリクエストの割合を書き込みます。   <pre>&gt; sarmquery -hist   resp=100,500,1000,2000,5000 - out   pctcount=stdout.txt</pre> </li> <li>■ pctselftime Siebel Server の内部で費やされた時間の割合を書き込みます。これは、SLA フィンガープリントとも呼ばれます。</li> </ul>

## Siebel ARM Query Tool でのフィルタの使用

この項では、Siebel ARM Query Tool ツールが取得するデータを制限するために、フィルタを使用する方法について説明します。

Siebel ARM ファイルには、1 つまたは複数のセグメントがあります。各セグメントには、ヘッダーおよび本文のセクションがあります。ヘッダーセクションにはメタデータが含まれます。このメタデータは、本文のセクションに含まれるデータについて説明するものです。たとえば、メタデータでは、本文セクションのデータが収集された時刻範囲と、本文セクションのデータ量について説明します。

選択フィルタを使用して、メタデータと実際のパフォーマンスデータをフィルタすることができます。

- 表 16 では、Siebel ARM ファイルのメタデータに使用可能なフィルタについて説明しています。
- 表 17 では、Siebel ARM ファイルのデータに使用可能なフィルタについて説明しています。

クエリーを式で表す場合は次の点に注意してください。

- 文字列フィルタ検索では大文字と小文字の区別がなく、ワイルドカードを受け入れます。たとえば、次のクエリーを使用できます。

```
> sarmquery -select user=JohnDoe
```

このクエリーは、ユーザー「johndoe」、「JOHNDOE」、および「JoHnDoE」に関連付けられた Siebel ARM レコードを取得します。

- 文字列フィルタは、先頭および末尾のワイルドカードも受け入れます。たとえば、次の各クエリーは、JohnDoe、場合によってはその他に関連付けられたレコードを取得します。

```
> sarmquery -select user="*doe"
> sarmquery -select user="john*"
> sarmquery -select user="*hndo*"
```

**備考：**ワイルドカードは先頭と末尾にのみ使用可能です。パターンの中央のワイルドカードは結果を取得しません。たとえば、パターン「jo\*ndoe」は、「johndoe」を取得しません。

- 選択フィルタを結合して、複数の条件に一致するレコードを取得することができます。たとえば、次の選択フィルタは、少なくとも 5 秒間実行されたすべてのスクリプト実行レコードを取得します。

```
> sarmquery -select area=script -select resp=5000
```

表 16 に説明のあるすべてのオプションの前には、オプション `-select` を付けます。

表 16 Siebel ARM Query Tool のヘッダーフィルタ

フラグ	説明
<code>component=&lt;name&gt;</code>	指定されたコンポーネントによって生成された Siebel ARM ファイルのヘッダーを選択します。
<code>fillfactor=&lt;percent&gt;</code> または <code>fillfactor=&lt;minpct&gt;, &lt;maxpct&gt;</code>	このフラグを使用することにより、パーセンテージ値を指定します。Siebel ARM Query Tool は、 <code>&lt;percent&gt;</code> パーセントが使用されているヘッダーを取得します。あるいは、値の範囲を取得する場合は、2 つの引数、最小パーセンテージ <code>&lt;minpct&gt;</code> と最大パーセンテージ <code>&lt;maxpct&gt;</code> を指定することができます。一般的に、ヘッダーには、一定数の Siebel ARM レコードを保持する容量があります。パーセンテージ値は、その容量に対する実際のレコード数の割合です。
<code>host=&lt;name&gt;</code>	指定されたホスト上で生成された Siebel ARM ファイルのヘッダーを選択します。
<code>procid=&lt;int&gt;</code>	ID (pid) が <code>&lt;int&gt;</code> であるプロセスによって生成された Siebel ARM ファイルのヘッダーを選択します。
<code>segcapacity=&lt;nrecs&gt;</code> または <code>segcapacity=&lt;min&gt;, &lt;max&gt;</code>	Siebel ARM レコードを保持する容量が <code>&lt;nrecs&gt;</code> によって指定されたレコード数に一致するヘッダーを選択します。 <code>&lt;nrecs&gt;</code> には、レコード数を指定します。あるいは、値の範囲を取得する場合は、2 つの引数、最小レコード数と最大レコード数を指定することができます。
<code>segduration=&lt;nsecs&gt;</code> または <code>segduration=&lt;min&gt;, &lt;max&gt;</code>	期間が指定された秒数に一致するヘッダーを選択します。ヘッダーの期間は、ファイルセグメントがフラッシュされるまでにメモリ内に存在した秒数です。あるいは、値の範囲を取得する場合は、2 つの引数、最小秒数と最大秒数を指定することができます。
<code>segid=&lt;min&gt;</code> または <code>segid=&lt;min&gt;, &lt;max&gt;</code>	少なくとも <code>&lt;min&gt;</code> の内部セグメント ID を持つヘッダーを取得する場合は、単一引数を指定します。あるいは、値の範囲を取得する場合は、2 つの引数を指定できます。

表 16 Siebel ARM Query Tool のヘッダーフィルタ

フラグ	説明
segsize=<size> または segsize=<min>, <max>	セグメントサイズは、バイト単位のヘッダーおよび本文のサイズです。セグメントサイズが <size> バイトであるすべてのヘッダーを取得する場合は、<size> の単一引数を指定します。範囲を取得する場合は、2 つの引数、<min> および <max> を指定します。
server=<name>	指定された Siebel <name> 上で生成された Siebel ARM ファイルのヘッダーを取得します。
sourcetype=<val>	指定されたタイプのサーバーまたはプロセスによって生成された Siebel ARM ファイルのヘッダーを取得します。ヘッダーを取得するには、<val> の代わりに次のパラメータのいずれかを指定します。 <ul style="list-style-type: none"> <li>■ app これは、Siebel Servers によって生成されたヘッダーを取得します。</li> <li>■ web これは、Siebel Web Server Extension (SWSE) によって生成されたヘッダーを取得します。</li> <li>■ cli これは、Siebel モバイル Web クライアントなどの他のクライアントプログラムによって生成されたヘッダーを取得します。</li> </ul>
threshold=<min> または threshold=<min>, <max>	しきい値を、ミリ秒単位で指定します。Siebel ARM フレームワークでは、合計応答時間がしきい値より小さいパフォーマンスレコードは破棄されます。Siebel ARM ファイルが生成された時点のしきい値設定は、ヘッダーに保存されます。 <p>しきい値が少なくとも &lt;min&gt; ミリ秒の Siebel ARM レコードを含むすべてのヘッダーを取得するには、単一引数を指定します。</p> <p>最小しきい値と最大しきい値の範囲の値を取得するには、2 つの引数を指定します。</p>

表 16 Siebel ARM Query Tool のヘッダーフィルタ

フラグ	説明
starttime=<start>	<p>開始時刻を指定して、指定された開始時刻の後に終了した Siebel ARM レコードを含むヘッダーを取得します。開始時刻は次の方法で指定できます。</p> <ul style="list-style-type: none"> <li>■ 形式 YYYY-MM-DD hh:mm:ss の文字列 次に例を示します。 &gt; sarmquery -select starttime="2006-02-13 17:05:00"</li> <li>■ 協定世界時 (UTC) に解釈される数値 次に例を示します。 sarmquery -select starttime=1108083900</li> <li>■ 現在の時刻からの秒数を示す負の数： たとえば、このコマンドは、299 秒以内に生成されたデータを含むヘッダーを取得します。 &gt; sarmquery -select starttime=-300</li> </ul> <p><b>備考：</b> 時間フィルタは、Siebel ARM レコードの生成終了時刻に対して比較されます。</p>
endtime=<end>	<p>終了時刻を指定して、指定された終了時刻よりも前に生成された Siebel ARM レコードを含むヘッダーを取得します。</p> <p><b>備考：</b> 時間フィルタは、Siebel ARM レコードの生成終了時刻に対して比較されます。</p> <ul style="list-style-type: none"> <li>■ 形式 YYYY-MM-DD hh:mm:ss の文字列 次に例を示します。 &gt; sarmquery -select endtime="2006-02-13 17:05:00"</li> <li>■ 協定世界時 (UTC) に解釈される数値 次に例を示します。 sarmquery -select endtime=1108083900</li> <li>■ 現在の時刻からの秒数を示す負の数： たとえば、このコマンドは、299 秒以内に生成されたデータを含むヘッダーを取得します。 &gt; sarmquery -select endtime=-300</li> <li>■ 開始時刻後の秒数を示す正の数： たとえば、このコマンドは、601 秒以上前に生成されたデータを含むヘッダーを取得します。 &gt; sarmquery -select starttime=-600</li> </ul>

表 17 では、Siebel ARM データを取得するクエリーを式で表す際に使用可能なフィルタについて説明しています。

表 17 に説明のあるすべてのオプションの前には、オプション `-select` を付けます。

表 17 レコード選択のための Siebel ARM Query Tool のフラグ

フラグ	説明
<code>clickid=&lt;ID&gt;</code>	Web サーバーの Siebel ARM ファイル ( <code>-select sourcetype=web</code> ) には、ユーザークリック ID (オペレーション ID と呼ばれる) が含まれます。このフィルタは、クリック ID が <code>&lt;ID&gt;</code> に等しいすべてのレコードを取得します。
<code>foreign</code>	このフラグを使用して、親レコードが異なるプロセスから生じるレコードを取得します。これは、バッチモードコンポーネントから生成された Siebel ARM データの場合によく該当します。
<code>orphan</code>	ルートレコードではなく、親レコードが入力 Siebel ARM ファイルで見つからなかったレコードを選択します。
<code>level=&lt;level&gt;</code> または <code>level=&lt;min&gt;,&lt;max&gt;</code>	アプリケーションプログラムインターフェイス (API) のレベルが少なくとも <code>&lt;level&gt;</code> に等しいレコードを取得するために、1 つの引数を指定します。最小レベルと最大レベルの範囲の値を取得するために、2 つの引数を指定します。API レベルは、次のように Siebel ARM レコードの重要性を示します。 <ul style="list-style-type: none"> <li>■ <code>&lt;level&gt; = 1</code> SARMLevel=1 に相当します。これは、高いレベルの情報を示します。</li> <li>■ <code>&lt;level&gt; = 2</code> SARMLevel=2 に相当します。これは、詳細パフォーマンス情報を示します。</li> <li>■ <code>&lt;level&gt; = 3</code> SARMLevel=3 に相当します。これは、デバッグ情報または内部パフォーマンス情報を示します。</li> </ul>
<code>parentid=&lt;ID&gt;</code>	親 ID が <code>&lt;ID&gt;</code> である Siebel ARM レコードを選択します。
<code>rootid=&lt;ID&gt;</code>	ルート ID (インタラクティブモードコンポーネントでは「リクエスト ID」とも呼ばれる) が <code>&lt;ID&gt;</code> である Siebel ARM レコードを選択します。

表 17 レコード選択のための Siebel ARM Query Tool のフラグ

フラグ	説明
sarmid=<sarmid> または sarmid=<procid>. <sarmid> または sarmid=<segid>. <procid>.<sarmid >	Siebel ARM レコードのプライマリ ID を取得します。 ■ <sarmid> は、プロセス/コンポーネント内のこのパフォーマンスレコードを一意に識別する番号です。 ■ <procid> は、プロセス ID (「-select procid=<procid>」と同じ) です。 ■ <segid> は、セグメント ID (「-select segid=<segid>,<segid>」と同じ) です。
sessionid=<sessid>	すべてのセッションベースのコンポーネントのパフォーマンスデータには、セッション ID として知られる属性が含まれます。同じセッション ID を持つすべてのプロセスから生成されるすべてのパフォーマンスデータは、単一ユーザーの単一セッションに属すると想定されます。 Siebel ARM Query Tool は、<sessid> を使用した大文字小文字の区別のない検索を実行します。ワイルドカードが使用可能です。
taskid=<taskid>	タスク <taskid> に関連付けられたすべてのレコードを選択します。
threadid=<threadid>	OSスレッドIDが<threadid>であるスレッドによって作成されたすべてのレコードを選択します。
user=<username>	ユーザー名 <username> によって作成されたすべてのレコードを選択します。 <username> は、通常、ログイン名です。検索は大文字小文字の区別がありません。またワイルドカードが使用可能です。

表 17 レコード選択のための Siebel ARM Query Tool のフラグ

フラグ	説明
area=<area> area=<areacode> subarea=<sub> subarea=<subcode> pararea=<parea> pararea=<pareacode> parsubarea=<psarea> parsubarea=<pscode>	<p>2つのフィルタ、エリアとサブエリアは、各 Siebel ARM レコードを識別します。</p> <p>エリアおよびサブエリアは、Siebel Server の論理セクションです。たとえば、Siebel Web Engine (SWE) エリアは Web ページを作成します。SWE に関連付けられた Siebel ARM レコードは、Web ページ作成のパフォーマンスを示します。同様に、データベースコネクタ (DBC) エリアは、エンタープライズデータベースへの接続を表します。DBC に関連付けられた Siebel ARM レコードは、データベースクエリーのパフォーマンスを示します。たとえば、次のコマンドを入力します。</p> <pre>&gt; sarmquery -select area=DBC</pre> <p>これは、データベースクエリーに関連付けられたすべての Siebel ARM レコードを取得します。</p> <p>Siebel ARM Query Tool により、エリア、サブエリア、および説明の完全なリストを取得できます。次のコマンドは、ファイル map.csv に完全なリストを保存します。</p> <pre>&gt; sarmquery -output map=map.csv</pre> <p>数値エリアまたはサブエリアコードがわかっている場合は、それらをこのコマンドで直接使用できます。</p> <p>または、記号の文字列形式を使用することもできます。文字列形式を使用する場合は、完全なテキストを使用する必要はありません。エリアまたはサブエリアを一意に識別できる場合には、部分的なテキストを使用することができます。</p> <p>フィルタ pararea および parsubarea は、親エリアおよびサブエリアがそれぞれ &lt;parea&gt; および &lt;psarea&gt; の Siebel ARM レコードを選択する点を除いて、area と subarea に類似しています。次に例を示します。</p> <pre>&gt; sarmquery -select pararea=swe</pre> <p>これは、親エリアが SWE であるすべての Siebel ARM レコードを取得します。</p>
children=0	このフラグを使用して、子レコードを持たない Siebel ARM レコードを取得します。
children=<count> または children=<min>,<max>	少なくとも <count> 個の子レコードを持つ Siebel ARM レコードを取得するには、1つの引数を指定します。<min> ~ <max> 個の子レコードを持つレコードを取得する場合は、2つの引数を指定します。
cputime=<ms> または cputime=<min>,<max>	CPU で少なくとも <ms> ミリ秒を費やした Siebel ARM レコードを取得するには、1つの引数を指定します。CPU で <min> ~ <max> ミリ秒を費やした Siebel ARM レコードを取得する場合は、2つの引数を指定します。

表 17 レコード選択のための Siebel ARM Query Tool のフラグ

フラグ	説明
depth=<depth> または depth=<min>,<max>	ルートノードから <depth> 段階未満の Siebel ARM レコードを取得する場合は、整数の引数を 1 つ指定します。レコードの範囲を取得する場合は、整数の引数を 2 つ指定します。
descendants=<count> または descendants=<min>,<max>	少なくとも <count> 個の子孫を持つ Siebel ARM レコードを取得する場合は、整数の引数を 1 つ指定します。値の範囲を取得する場合は、2 つの引数を指定します。
instance=<string> detail=<string> int1=<int>	これらのフィルタを使用することにより、次のような Siebel ARM レコードを取得できます。 <ul style="list-style-type: none"> <li>■ instance このフィルタは、インスタンスメタデータが &lt;string&gt; に等しい Siebel ARM レコードを取得します。インスタンスメタデータには通常、ビュー名、画面名、ユーザー名、ワークフロー名、スクリプト名などの名前が含まれます。</li> <li>■ detail このフィルタは、詳細メタデータが &lt;string&gt; に等しい Siebel ARM レコードを取得します。詳細メタデータには通常、識別情報が含まれます。たとえば、データベース行 ID またはビジネスコンポーネントメソッドの名前が含まれる場合があります。</li> <li>■ int1 このフィルタは、int1 フィルタが &lt;int&gt; に等しい Siebel ARM レコードを取得します。int1 メタデータには通常、カウンタ値、タスク ID、その他の特定されない情報が含まれます。</li> </ul> <p><b>備考：</b> 先行するメタデータフィールドが参照する値は、関連付けられたエリアまたはサブエリアによって異なります。</p>
memusage=<excl> または memusage=<min>,<max>	割り当てられたメモリまたは割り当て解除されたメモリが、<excl> バイトより大きいすべての Siebel ARM レコードを取得するには、1 つの引数を指定します。 指定された範囲内のメモリ割り当てイベントまたはメモリ割り当て解除イベントを記録する Siebel ARM レコードの範囲を取得する場合は、2 つの引数を指定します。次に例を示します。 <pre>&gt; sarmquery -select memusage=1000000</pre> これは、関連付けられたイベントが 1 MB（以上）のメモリ割り当てまたは 1 MB 以上のメモリ割り当て解除を記録しているすべての Siebel ARM レコードを取得します。

表 17 レコード選択のための Siebel ARM Query Tool のフラグ

フラグ	説明
pctcpu=<pct> または pctcpu=<min>,<max>	<pct> パーセントの CPU 消費を示す Siebel ARM レコードを取得するには、単一引数を指定します。<min> ~ <max> パーセント間の範囲を取得する場合は、2つの引数を指定します。
resptime=<ms> または resptime=<min>,<max>	消費された包括的なウォールクロックタイムが <ms> ミリ秒である Siebel ARM レコードを取得するには、1つの引数を指定します。<min> ~ <max> の範囲を取得する場合は、2つの引数を指定します。包括的なウォールクロックタイムとは、アーキテクチャのある特定の部分で費やされた時間であり、そのすべての子孫で費やされた時間も含まれます。たとえば、SWE エリアの応答時間は、ビジネスコンポーネントイベントやデータベースアクセスを評価する時間を含む、Web ページの構築に費やされる時間です。
selftime=<ms> または selftime=<min>,<max>	1つの引数の場合、消費された排他的な時間が <ms> ミリ秒である Siebel ARM レコードを選択します。2つの引数の場合、<min> および <max> により範囲を指定します。排他的な時間とは、子孫で費やされた時間を除く、エリア（そのエリア内だけ）で費やされた時間です。したがって、SCRIPT エリアのセルフタイムは、スクリプトエンジンで費やされた時間であり、データベース、オブジェクトマネージャ、ワークフローなどで費やされた時間は含まれません。
starttime=<start> または endtime=<end>	語義および構文は、フィルタがタイムスタンプに基づいて Siebel ARM レコードを選択する場合を除き、ヘッダーに対して説明された time フィルタと同様です。
tree=all tree=descendant tree=ancestor tree=parent	<p>これらは、単一レコードベースでなく、レコード単位で操作する選択フィルタの特殊セットです。ほとんどの場合、これらは、セット全体でパフォーマンスレコードを置き換えるのに使用されます。</p> <p>これらの演算子を理解するには、Siebel ARM レコードが親および子ノードとともに実行スレッドのツリーを形成していることを認識することが重要です。</p> <ul style="list-style-type: none"> <li>■ -select tree=all 選択したレコードをそのツリーのすべてのレコードで置き換えます。</li> <li>■ -select tree=parent 選択したレコードをその親で置き換えます。レコード自体は破棄されます。</li> <li>■ -select tree=ancestor 選択したレコードからツリーのルートに続くすべての Siebel ARM レコードで、選択したレコードを置き換えます。選択したレコードは、このセットにも含まれます。</li> <li>■ -select tree=children 選択したレコードをその直属の子で置き換えます。選択したレコードは含まれません。</li> </ul>

## Siebel ARM Query Tool による Siebel ARM データの集計

この項では、Siebel ARM Query Tool を使用する際に利用可能な集計オプションについて説明します。集計では、共通の属性を持つ Siebel ARM レコードをグループ化し、そのグループの統計情報を計算します。

複数の `-aggregate` オプションを指定できます。この場合、Siebel ARM Query Tool はサブ集計として解釈します。

Siebel ARM Query Tool は、次の項目の合計、最大、最小、平均、および関与したレコード数を計算できます。

- 包括的な応答時間
- 排他的な応答時間（セルフタイム）
- 包括的な CPU 時間
- 排他的な CPU 時間
- 包括的なメモリ使用量
- 排他的なメモリ使用量

**備考：** 集計は、選択フィルタによって取得された Siebel ARM レコードに対して計算されます。

表 18 に説明のあるすべてのオプションの前には、オプション `-aggregate` を付けます。

表 18 集計のフラグオプション

フラグ	説明
area	エリアに対して同じ値を持つ Siebel ARM レコードをグループ化します。
subarea	サブエリアに対して同じ値を持つ Siebel ARM レコードをグループ化します。サブエリア、または次の例のようにエリアに対するフィルタによってグループ化する場合は、エリアによってもグループ化することが一般的であることに注意してください。 > <code>sarmquery -aggregate area -aggregate subarea</code> > <code>sarmquery -select area=DBC -aggregate subarea</code>
instance	同じ値のインスタンスメタデータを持つ Siebel ARM レコードをグループ化します。 インスタンスメタデータには通常、スクリプト、ワークフロー、ビューなどの名前が含まれます。これは、インスタンスメタデータの値がエリア/サブエリアに依存していることを意味します。結果として、集計インスタンスの前には、次の例に示すように、エリア/サブエリアの集計またはエリア/サブエリアに対するフィルタを指定します。 > <code>sarmquery -aggregate area -aggregate instance</code> > <code>sarmquery -select area=script -aggregate instance</code>

表 18 集計のフラグオプション

フラグ	説明
server component host procid	<p>以下に対して同じ値を持つ Siebel ARM レコードを集計します。</p> <ul style="list-style-type: none"> <li>■ Server</li> <li>■ Component</li> <li>■ Host</li> <li>■ Procid</li> </ul>
user	<p>ユーザー名に対して同じ値を持つ Siebel ARM レコードを集計します。</p> <p><b>備考：</b> ユーザー名は、大文字と小文字が区別されます。</p>
sessionid	<p>セッション ID に対して同じ値を持つ Siebel ARM レコードを集計します。</p> <p><b>ヒント：</b> このフラグは、次の例のように user のフラグとともに使用される場合に役立ちます。</p> <pre>&gt; sarmquery -aggregate user -aggregate sessionid &gt; sarmquery -select user=andy -aggregate sessionid</pre>
clickid	<p>クリック ID に対して同じ値を持つ Siebel ARM レコードを集計します。</p> <p>通常、インタラクティブコンポーネントでは、ユーザーは複数のセッションを持ち、セッションには複数のリクエストがあります。クライアント (ブラウザ) 上の単一ユーザーアクションの結果、複数のリクエストが生じる場合があります。そのような場合は、これらのリクエストのそれぞれがクリック ID として知られる同じ固有の ID に関連付けられています。</p> <p>ユーザーアクションの結果、単一リクエストが生成される場合にも、この ID は生成されリクエストに関連付けられます。</p> <p><b>ヒント：</b> 次の例のように、user および sessionid とともにこのフラグを使用します。</p> <pre>&gt; sarmquery -aggregate user -aggregate sessionid -aggregate clickid &gt; sarmquery -select user=andy -aggregate sessionid -aggregate clickid</pre>

表 18 集計のフラグオプション

フラグ	説明
time=<interval>	<p>&lt;interval&gt; によって指定される間隔でタイムスタンプにより Siebel ARM レコードを集計します。ここで &lt;interval&gt; は分単位の値です。</p> <p>たとえば、5 と指定すると、12:00 ~ 12:05 のタイムスタンプを持つ Siebel ARM レコードは 1 つの集計を形成し、12:05 ~ 12:10 のタイムスタンプを持つ Siebel ARM レコードは別の集計を形成します。以下同様に続きます。</p> <p>次の例のコマンドは、15 分間隔のアプリケーションサーバーの平均応答時間をコマンドウィンドウにレポートします。</p> <pre>&gt; sarmquery -select source=app -select subarea=infra_entry -aggregate time=15</pre>

## Siebel ARM Query Tool を使用したヒストグラムの生成

ヒストグラムは、特殊なタイプの集計です。ヒストグラムを使用することにより、Siebel ARM データに対してクエリーを発行すると、Siebel ARM Query Tool が取得する結果を集計できます。ヒストグラムを使用しない場合には多くの値を返しすぎる可能性があります。

たとえば、次のクエリーを使用できます。

```
> sarmquery -aggregate resptime
```

これは、最初の行が 1 ミリ秒で完了したリクエストを返し、2 番目の行が 2 ミリ秒で完了したリクエスト数を返す場合に、数百万行を取得する可能性があります。

より効果的なクエリーでは、次の例のようにヒストグラムを生成します。

```
> sarmquery -histogram resptime=100,200,300,400,500
```

これは、6 つの行を返します。最初の行は、100 ミリ秒未満で完了するリクエスト数を集計します。2 番目の行は、100 ~ 200 ミリ秒間で完了するリクエスト数を集計します。以下同様にして続きます。

この例では、値 100, 200, 300 などは「bin エンドポイント」として知られています。

表 19 では、ヒストグラムパラメータとともに使用可能なフラグオプションについて説明しています。表 19 に説明のあるすべてのオプションの前には、オプション `-histogram` を付けます。

表 19 ヒストグラムパラメータのフラグオプション

フラグ	説明
<code>resptime=&lt;val1&gt;,&lt;val2&gt; ... &lt;valN&gt;</code>	引数 <code>&lt;val1&gt;</code> および <code>&lt;val2&gt;</code> で bin エンドポイントを指定する場合に、包括的応答時間のヒストグラムを作成します。 <code>&lt;val1&gt;</code> および <code>&lt;val2&gt;</code> はミリ秒単位で指定します。
<code>selftime=&lt;val1&gt;,&lt;val2&gt;...&lt;valN&gt;</code>	<code>&lt;val1&gt;...&lt;valN&gt;</code> でエンドポイントを指定する場合に、セルフタイム（排他的応答時間）のヒストグラムを作成します。値はミリ秒単位で指定します。
<code>pctcpu=&lt;val1&gt;,&lt;val2&gt;...&lt;valN&gt;</code>	消費された CPU 時間のパーセンテージのヒストグラムを作成します。引数は 0 ~ 100 の整数を指定します。
<code>children=&lt;val1&gt;,&lt;val2&gt;...&lt;valN&gt;</code>	実行ツリーの幅のヒストグラムを作成します。
<code>depth=&lt;val1&gt;,&lt;val2&gt;...&lt;valN&gt;</code>	実行ツリーの深さのヒストグラムを作成します。

## Siebel ARM Query Tool でのマクロの使用

Siebel ARM Query Tool は、マクロの使用をサポートしています。マクロにより、Siebel ARM Query Tool の入力として使用可能な複雑なクエリーをファイルに保存して、再使用することができます。

マクロを作成するには、次の操作が必要です。

- .CFG ファイルにクエリーの構文を記述する（たとえば、`macro.cfg`）  
クエリーの各部分は、.CFG ファイルの別の行に表示される必要があります。
- .CFG ファイルのクエリーの最初の行の前にマクロの名前を定義する

たとえば、ユーザーの平均ログイン時間のデータを取得する次のクエリーは、コマンドラインからの 1 つのエントリとして入力できます。

```
> sarmquery -input d:¥sarmdata -select source=web -select subarea=swse_login -select tree=all -select depth=1 -aggregate user -output avginclresp=stdout.txt
```

ただし、.CFG ファイルでは、このクエリーはマクロ名 [Login] を割り当てられ、次のように記述されます。

```
[Login]
-input d:¥sarmdata
-select source=app
-select subarea=objmgr_sess_relogin
-select tree=all
-select depth=1
-aggregate user
-output avginclresp=stdout.txt
```

Siebel ARM Query Tool でマクロを使用するには、2つのパラメータを指定する必要があります。

### ■ マクロを含む .CFG ファイルの場所

**備考：**1つの .CFG ファイルには、複数のマクロを含めることができます。

### ■ 実行するマクロの名前

次の例では、Siebel ARM Query Tool が macro.cfg ファイルに保存される Login マクロを実行することをリクエストします。

```
> sarmquery -config file=macro.cfg -config macro=Login
```

Siebel ARM Query Tool では、コマンドラインでマクロの個々の部分が指定されたかのように、マクロの入力引数を読み取ります。これは、マクロに存在しない可能性のある追加の引数を指定できることを意味します。次の例では、Login マクロを実行し、さらにタイムスライスを指定しています。

```
> sarmquery -config file=macro.cfg -config macro=Login -select starttime="2004-06-10 10:00:00" -sel endtime="2004-06-11 09:59:59"
```

マクロ言語の構文およびその他のオプションについては、コマンドラインから次のコマンドを実行します。

```
> sarmquery -macrosyntax
```

# 索引

## 数字

- 1 時間あたりに処理された受信電子メール  
メッセージの数 81
- 1 時間あたりに処理された電話の着信数 71
- 1 時間あたりに処理された電話の発信数 71
- 1 秒あたりのトランザクション数 (TPS)、  
スループットの定義 21
- 1 秒あたりのユーザーコミュニケーションの  
アクション数 71
- 4GT 196

## A

### AIX

- IBM HTTP Server のチューニング 199
- Siebel Server のチューニング 201
- カーネル設定のチューニング 203

### All Mode ユーザープロパティ、パフォーマンスの ための設定 125

### AOM

- 「Siebel Application Object Manager (AOM)」を  
参照

### AOM あたりのユーザー、メモリコンシューマと しての 36

### AOM コンポーネント、チューニング 74

### AOM のスレッドプール

- 使用 45
- 設定 45
- 備考、オーバーヘッドの処理 45
- 備考、推奨事項 45

## C

### Cache Data プロパティ、パフォーマンス 188

### Call Center、パラメータの設定例 33

### ChannelCleanupTimer パラメータ 78

### Check No Match プロパティ、パフォーマンス 191

### CommConfigCache パラメータ (AOM) 75

### CommConfigManager パラメータ (AOM) 75

### CommConfigMgr

- サーバーコンポーネント 70, 75

### CommInboundProcessor

- サーバーコンポーネント 70, 80

### CommInboundRcvr

- サーバーコンポーネント 70, 80
- スレッドの設定 83

### CommLogDebug パラメータ (AOM) 77

### CommLogFile パラメータ (AOM) 77

### CommMaxLogKB パラメータ (AOM) 77

### CommMaxMsgQ パラメータ (AOM) 78

### CommOutboundMgr

- サーバーコンポーネント 70, 80

### CommReleaseLogHandle パラメータ (AOM) 77

### CommReqTimeout パラメータ (AOM) 78

### CommSessionMgr

- AOM マシン上での実行 72
- コンポーネントのチューニング、最善の方法 74
- サーバーコンポーネント 69
- ログパラメータ 77

### Communications Configuration Manager

- サーバーコンポーネント 70

### Communications Inbound Processor

- サーバーコンポーネント 70, 80

### Communications Inbound Receiver

- サーバーコンポーネント 70, 80

### Communications Outbound Manager

- サーバーコンポーネント 70, 80

### Communications Server

- 「Siebel Communications Server」を参照

### Communications Session Manager

- サーバーコンポーネント 69

### Configurator

- 「Siebel Configurator」を参照

### Configurator キャッシュ

- 製品クラスの変更によるキャッシュの更新 114
- 製品クラスの変更によるキャッシュの  
リフレッシュ 114
- 製品の変更によるキャッシュのリフレッシュ 113
- 属性定義の変更によるキャッシュの更新 115
- 属性定義の変更によるキャッシュの  
リフレッシュ 115

### Configurator のメモリキャッシュ

- キャッシュパラメータのサイズ決定 112

### CPU

- AOM コンポーネントのチューニングの  
ガイドライン 29
- 定義されたハードウェアリソース 28

### CSV 変換

- 実行 225
- データ、説明 237

### CTI ミドルウェア 70

**D****Database Extract**

スループットの向上 164

**DB2 データベース**

最適化のヒント 155

バージョン 8 のオプション 145

**DbXtract**

「Database Extract」を参照

**Driver:DriverLogFile パラメータ (Siebel CTI Connect ドライバ) 77****DSMaxCursorSize パラメータ (AOM) 35****DSPreFetchSize パラメータ (AOM) 35****E****EAI Object Manager**

EAI Siebel Adapter、並列実行 123

注意、2つのセッションの並列実行 123

ログの無効化 121

**EAI Siebel アダプタのパフォーマンス**

EAIによって生成されるSQLの分析 122

インテグレーションオブジェクトのサイズの  
最小化 122

スクリプトのレビュー 121

注意、2つのセッションの並列実行 123

ビジネスオブジェクトのキャッシュ 123

並列実行 123

ログの無効化 121

**EAIによって生成されるSQL 122****EIM テーブル**

FREELIST パラメータの設定 154

MS SQL テーブルの削除 151

Oracle db テーブルの修正 153

Oracle データベーステーブルの削除 153

アーカイブのログ作成の無効化 153

インデックス 144

オブジェクトの再構築 154

説明 128

断片化の解消 150

データの並列ロードの使用 151

テーブルのキャッシュ 154

テーブルの更新 155

適切な統計の作成 145

レコードの制御 147

**EIM テーブルの削除**

MS SQL 151

Oracle データベース 153

**EIM の使用計画**

EIM プロセスのテスト 133

Siebel アプリケーションへのマッピング 132

条件の定義 131

**EIM プロセス**

オペレーションごとの分離 136

実装手順 135

テスト 133

**EIM プロセスのテスト 133****Email Response**

「Siebel Email Response」を参照

**EnableCDA パラメータ (AOM) 35****EnableSIBusyCursor 66****EnableViewCache パラメータ、ビューのレイアウトの  
キャッシュを無効にする 64****Enterprise Application Integration**「Siebel EAI、パフォーマンスのチューニング」を  
参照**eProdCfgObjMgr サーバーコンポーネント 39, 100****eService、パラメータの設定例 33****EXEC、トリガーの無効化 149****F****File System Manager**

サーバーコンポーネント 80

**Force Active プロパティ**

FALSE に設定する 188

**FREELIST パラメータ、設定 154****FSMSrvr**

サーバーコンポーネント 80

**H****HP-UX**

Apache Web Server のチューニング 208

SWSE のチューニング 198

カーネル設定のチューニング 210

最大スレッド制限の設定 208

スケジューラのチューニング 211

**HTTP Inbound Transport、パフォーマンスの  
向上 121****I****IBM AIX、「AIX」を参照****IBM DB2**

パフォーマンスチューニング 159

ローディングプロセス 158

**IBM DB2 UDB for z/OS 158****IBM HTTP Server**

AIX のチューニング 199

静的なファイルキャッシュの指定 59

**IBM WebSphere MQ Transport、****パフォーマンスの向上**

キュー、テストとオプション 119

受信 WebSphere MQ メッセージの実行 119

送信メッセージとキャッシュ 120

パフォーマンス追跡情報の設定 120

**IFB ファイル**

- 最適化 135
- 最適化の確認 136
- パフォーマンスをテストする目的で使用 139

**Internet SMTP/POP3 Server コミュニケーション  
ドライバ 83****L****LogDebug パラメータ (CommSessionMgr) 77****LogDebug パラメータ (Siebel CTI Connect  
ドライバ) 77****LogFile パラメータ (CommSessionMgr) 77****M****MaxLogKB パラメータ (CommSessionMgr) 77****MaxLogKB パラメータ (Siebel CTI Connect  
ドライバ) 77****MaxMTServers パラメータ**

- 計算式 32
- 式の変数 32
- 設定 30
- 設定ガイドライン 31
- 設定例 33
- 設定、効果 31

**MaxTasks パラメータ**

- CommSessionMgr コンポーネント 74
- 計算式 32
- 式の変数 32
- 設定 30
- 設定ガイドライン 31
- 設定例 33
- 設定、効果 31

**MemProtection パラメータ (AOM) 35****Microsoft**

- IIS、静的なファイルキャッシュの指定 59
- Internet Explorer、推奨設定 57

**MinMTServers パラメータ**

- 計算式 32
- 式の変数 32
- 設定 30
- 設定ガイドライン 31
- 設定例 33
- 設定、効果 31

**MinTrxDBConns パラメータ、設定 43****MLOV クエリーとキャッシュのパフォーマンス 183****MS SQL Server**

- TempDB の使用 152
- 設定パラメータ 152
- データの並列ロードの使用 151
- テーブルの削除 151
- テーブルの断片化の解消 150

**MT サーバー**

- 「マルチスレッドプロセス」を参照

**N****NUM\_IPTABLE\_LOAD\_CUTOFF**

- 拡張パラメータ 148

**O****Object Manager、Solaris 用のインスタンスの  
チューニング 207****Oracle データベースサーバー**

- FREELIST パラメータの設定 154
- Oracle オプティマイザーモード 153
- アーカイブのログ作成の無効化 153
- オブジェクトの再構築 154
- 過剰なテーブルの断片化の回避 153
- テーブルの更新 155
- テーブルの削除 153

**P****Push Keep Alive コミュニケーションドライバ 78****R****RAID**

- パフォーマンスチューニング 130

**RDBMS**

- 「データベース接続」を参照

**ReleaseLogHandle パラメータ  
(CommSessionMgr) 77****ReleaseLogHandle パラメータ (Siebel CTI  
Connect ドライバ) 77****REMOVE、トリガーの無効化 149****S****S\_ESCL\_ACTN\_REQ テーブル 89****S\_ESCL\_REQ テーブル 89****S\_ESCL\_STATE テーブル 89****S\_ORG\_EXT、パフォーマンスの向上 144****Search Specification パラメータ**

- 使用の最小化 93
- 推奨されるフィールドのインデックス作成 93

**Search Specification プロパティ、データベース  
インデックスの管理 185****SearchSpec パラメータ**

- 使用の最小化 93
- 推奨されるフィールドのインデックス作成 93

**Server Request Broker**

- サーバーコンポーネントと例 69
- チューニング 49

**Service:ServiceLogFile パラメータ (Siebel CTI  
Connect ドライバ) 77****SessPerSisnConn パラメータ、説明 47****Siebel Application Object Manager (AOM)**

- CPU およびメモリ使用量のチューニング 29
- Server Request Broker のチューニング 49

- Siebel Configurator の実行 101
- Siebel Configurator の要素 100
- Siebel Configurator、専用展開用の設定 102
- Siebel アプリケーションの展開 27
- SISNAPI 接続プールの設定 47
- インフラストラクチャ 24
- 同じマシン上での CommSessionMgr の実行 72
- 活動のチューニング 25
- キャッシュのチューニング 34
- 共有された接続プール、設定例 42
- 共有接続の割り当て 42
- 共有接続プールの設定 41
- コンテキスト 15
- コンポーネントのチューニングの最善の方法 74
- サーバーコンポーネント、定義 69
- サーバーリソースの節約 75
- スレッドプールの使用 45
- スレッドプールの設定 45
- 説明と例 24
- 専用接続プールのシナリオ 44
- 専用接続プールの例 44
- 専用データベース接続プール 43
- 専用データベース接続プールの設定 43
- 専用データベース接続プールの割り当て 44
- データベース接続、仮定 37
- 展開、トポロジに関する注意事項 28
- 同時ユーザーおよびパフォーマンス 26
- ハードウェアリソースおよびパフォーマンス 28
- パフォーマンスドライバ 26
- パラメータ値、計算式 32
- パラメータ値、式の変数 32
- パラメータの設定、効果 31
- パラメータ、設定 30
- パラメータ、設定ガイドライン 31
- パラメータ、設定例 33
- パラメータ、相互の関係 30
- プールされていないデータベース接続 37
- プールデータベース接続 38
- 待ち時間およびパフォーマンス 27
- メモリコンシューマ 36
- モジュール、通信 25
- ログパラメータ 77
- Siebel Application Response Measurement**
- 「Siebel ARM」を参照
- Siebel ARM**
- ARM から CSV への変換データ、説明 237
- ARM データの CSV 変換、実行 225
- ARM ファイルの変換、最善の方法 218
- Siebel Server の Siebel ARM パラメータ 214
- データ、説明 225
- トランザクションの監視に使用 56
- パフォーマンス集計データ、データの説明 226
- パフォーマンス集計分析のタグ 227
- パラメータおよび変数 214
- ファイル、変換 220
- プロセスの有効化と設定 217
- ユーザーセッション追跡データのタグ 235
- ユーザーセッション追跡データ、説明 235
- 呼び出しグラフの生成データ、説明 232
- 呼び出しグラフの生成分析のタグ 232
- Siebel ARM Analyzer Tool**
- ARM から CSV への変換データ、説明 237
- ARM データの CSV 変換、実行 225
- 実行 220
- 説明 221
- データ、説明 225
- パフォーマンス集計データのタグ 227
- パフォーマンス集計データ、データの説明 226
- パフォーマンス集計分析の実行 222
- ユーザーセッション追跡データ、説明 235
- ユーザーセッション追跡のタグ 235
- ユーザーセッション追跡、実行 224
- 呼び出しグラフの生成データ、説明 232
- 呼び出しグラフの生成分析のタグ 232
- 呼び出しグラフの生成、実行 223
- Siebel ARM Query Tool**
- Siebel ARM データの集計 254
- コマンドラインパラメータ 239
- 出力の設定 242
- 説明 238
- ツールの設定 240
- 入力の設定 241
- ヒストグラムの生成 256
- フィルタの使用 245
- マクロの使用 257
- Siebel Business Applications**
- AIX 用の Siebel Server のチューニング 201
- AIX 用のカーネル設定のチューニング 203
- HP-UX スケジューラのチューニング 211
- HP-UX 用の HP Apache Web Server のチューニング 208
- HP-UX 用のカーネル設定のチューニング 210
- Solaris 用 Siebel Web Server Extension のチューニング 198
- Solaris 用チューニング 204
- Solaris 用の Object Manager インスタンスのチューニング 207
- Solaris 用の Siebel Server のパフォーマンスの最大化 206
- Solaris 用のカーネル設定のチューニング 205
- 最大スレッド制限の設定 208
- ワークフローのパフォーマンスの設定 94
- Siebel Call Center、パラメータの設定例 33**
- Siebel Collaboration 70**

**Siebel Communications Server**

- アーキテクチャとインフラストラクチャのチューニング 16
- コミュニケーションサポート活動 68
- コンポーネントトポロジ 72

**Siebel Configurator**

- AOM コンポーネントの実行 101
- AOM、専用展開用の設定 102
- Configurator キャッシュのサイズ決定 112
- カスタマイズ可能製品とクラス 105
- キャッシュ、タイプ 106
- キャッシュ、デフォルトの動作 107
- コンポーネント 100
- 最善の方法 103
- 専用サーバーの実行 102
- チューニング 104
- チューニング、説明 17
- データベース接続プール 39
- トポロジに関する注意事項 101
- パフォーマンスに影響する要因 100
- メモリキャッシュのパラメータ 109

**Siebel CTI Connect**

- Siebel 製品モジュール 70
- 画面ポップアップのパフォーマンスの向上 79
- ドライバのログパラメータ 77

**Siebel CTI Connect ドライバのログパラメータ 77****Siebel Developer Web クライアント、**

**SQL スプールの指定 175**

**Siebel EAI、チューニング**

- All Mode Sort ユーザープロパティ、設定 125
- EAI Siebel アダプタのパフォーマンス 121
- HTTP Inbound Transport のパフォーマンス 121
- IBM WebSphere MQ Transport のパフォーマンス 119
- Workflow Process Manager のパフォーマンスの向上 124
- 仮想ビジネスコンポーネントのパフォーマンス 123
- 最善の方法 118
- 説明 117
- チューニング、説明 17
- ディスクの確認 125
- データベースクエリーの最適化 125
- ビジネスコンポーネントの作成 125
- メッセージの最適化 125
- ログのオフ 125

**Siebel Email Response**

- 1 時間あたりに処理された受信電子メールの数 81
- CommInboundRcvr スレッド、設定 83
- Siebel Smart Answer、パフォーマンス 85
- Siebel Smart Answer、モジュール 81
- Siebel 割当マネージャ 81
- インフラ 80

- 活動の作成とパフォーマンス 84
- 顧客データの量 81
- サードパーティの電子メールサーバー 81
- 重要なサーバーコンポーネント 80
- チューニングを行うための最善の方法 83
- 電子メール処理ディレクトリ、管理 83
- トポロジ 82
- ログ、設定 84

**Siebel Enterprise Application Integration**

「Siebel EAI、チューニング」を参照

**Siebel eService、パラメータの設定例 33****Siebel File System**

Siebel Configurator コンポーネント 100

**Siebel Internet Session アプリケーション  
プログラミングインターフェイス**

「SISNAPI 接続プール」を参照

**Siebel Product Configuration Object  
Manager 39, 100****Siebel Remote**

- Database Extract のスループットの向上 164
- サーバーコンポーネント 163
- サーバーコンポーネント、チューニング 164
- 説明 163
- 同期 169
- モバイル Web クライアントのチューニング 167
- ルーティングモデル 169

**Siebel Server**

- AIX のチューニング 201
- Communications Configuration Manager 70
- Communications Inbound Processor 70
- Communications Inbound Receiver 70
- Communications Outbound Manager 70
- Communications Session Manager 69
- Siebel ARM パラメータ 214
- Siebel 製品モジュール 70
- Solaris のパフォーマンスの最大化 206
- Solaris 用のカーネル設定のチューニング 205
- サードパーティ製品のモジュール 70
- 通信コンポーネント 69
- ワークフローエージェント、複数のサーバー上で実行 91
- ワークフローポリシーグループ、管理のための作成 90

**Siebel Server、監視 161****Siebel Smart Answer 70**

Siebel Email Response、パフォーマンス 85  
モジュール 81

**Siebel Tools、チューニングの説明 17****Siebel Web Engine (SWE)**

ユーザーセッション追跡、実行 224

**Siebel Web Server Extension**

Solaris 用チューニング 198  
通信 25

**Siebel Web クライアント、チューニング**

- IBM HTTP Server、静的なファイル  
キャッシュ 59
- Microsoft IIS、静的なファイルキャッシュ 59
- Siebel クライアント、定義 52
- Sun Java System Web Server、静的なファイル  
キャッシュ 60
- Web サーバーおよびネットワーク容量 55
- 永続的なビューのレイアウトのキャッシュ 62
- クライアントハードウェアリソース 56
- 現在のビューレイアウトの取得 64
- 最善の方法 54
- システムコンポーネントのチューニング 56
- 静的なファイルキャッシュ 58
- 設定ガイドライン 57
- 説明 16
- パフォーマンスのテスト 55
- ビューのレイアウトのキャッシュ 60
- ビューのレイアウトのキャッシュサイズの設定 62
- ビューのレイアウトのキャッシュを無効にする 64
- ビュー、およびレイアウトのキャッシュ 65
- 複数の Siebel モジュールのサポート 53
- ブラウザキャッシュの管理 57
- メッセージバー、パフォーマンスの管理 65
- メモリ、キャッシュされたビューのプレロード 63
- ローカルマシンのリソース 53

**Siebel アプリケーション、マッピングの  
ガイドライン 132****Siebel クライアント、定義 52****Siebel スクリプト**

- 宣言的な手段、使用 179
- パフォーマンスのガイドライン 180

**Siebel 製品モジュール**

- Siebel Collaboration 70
- Siebel CTI Connect 70
- Siebel Smart Answer 70

**Siebel データベース、通信 25****Siebel モジュール、複数のモジュールのサポート 53****Siebel ユーザーリクエストフロー**

- ステップ 19
- プロセスフロー 19

**Siebel ワークフロー、チューニング**

- Search Specification パラメータ、使用の  
最小化 93
- Siebel Server の負荷、ワークフローポリシー  
グループの作成 90
- Workflow Process Manager のチューニング 96
- アーキテクチャとインフラストラクチャ 16
- エスカレーションアクションリクエストテー  
ブル 89
- エスカレーション状態テーブル 89
- エスカレーションリクエストテーブル 89
- 親ビジネスコンポーネントと子ビジネス  
コンポーネント、監視 94

**コンポーネント 87**

- 設定 94
- 説明 87
- パフォーマンス追跡情報 124
- ポリシー頻度分析ビュー、使用 88
- メモリのオーバーヘッドの監視 94
- ワークフローアクションエージェント、アクション  
間隔の設定 92
- ワークフローエージェント、複数のサーバー上で  
実行 91
- ワークフローポリシー、監視 88
- ワークフローポリシー、ログとファイルの使用 88
- ワークフローモニターエージェントとワークフロー  
アクションエージェント 91
- ワークフローモニターエージェント、アクション  
間隔の設定 92
- ワークポリシーグループ、スリープ間隔の設定 91

**Siebel 割当マネージャ 81****SISNAPI 接続プール、設定 47****Smart Answer**

「Siebel Smart Answer」を参照

**Solaris**

- Object Manager のインスタンスの  
チューニング 207
- Siebel Business Applications の  
チューニング 204
- Solaris 用の Siebel Server のパフォーマンスの  
最大化 206
- Sun Java System Web Server の  
チューニング 204
- SWSE のチューニング 198
- カーネル設定のチューニング 205

**Solaris、Siebel Server のパフォーマンスの  
最大化 206****Sort Specification プロパティ、データベース  
インデックスの管理 184****SQL**

- SQL 文の接続プール 42
- 時間のかかる文 142
- 備考、サポート 128

**SQLPROFILE、使用 142****SQL カーソルキャッシュ 34****SQL データキャッシュ 34****SQL、パフォーマンスが低いクエリー 94****SQL、パフォーマンスのための分析**

- Siebel Developer Web クライアント、SQL  
スプールの指定 175
- SQL クエリープランの例 178
- SQL クエリープラン、トラブルシューティングに  
使用 176
- SQL 追跡ファイル、トラブルシューティングに  
使用 176
- 説明 175
- データベースに対する SQL クエリー 178

**SRBroker (Server Request Broker)**

サーバーコンポーネントと例 69  
チューニング 49

**Sun Java System Web Server**

Solaris 用チューニング 204  
静的なファイルキャッシュの指定 60

**Sun Solaris. 「Solaris」を参照****T****TempDB、使用 152****Transaction Router**

スループットの向上 167  
チューニング 165

**TrickleSync 169****U****UNIX、パフォーマンスチューニング**

AIX 用の IBM HTTP Server のチューニング 199  
AIX 用の Siebel Server Extension の

チューニング 201

AIX 用のカーネル設定のチューニング 203

HP-UX スケジューラのチューニング 211

HP-UX 用 Web サーバーのチューニング 208

HP-UX 用のカーネル設定のチューニング 210

Siebel Server のパフォーマンスの最大化 206

Solaris 用 Siebel Web Server Extension 198

Solaris 用の Business Applications の  
チューニング 204

Solaris 用の Object Manager インスタンスの  
チューニング 207

Solaris 用のカーネル設定のチューニング 205  
最大スレッド制限の設定 208

**UPDATE STATISTICS パラメータ**

タスクの並列実行 149

**USE ESSENTIAL INDEX HINTS 139****USE INDEX HINTS パラメータ**

使用 139

**USING SYNONYMS パラメータ**

使用 148

**V****ViewPreloadSize パラメータ、設定 63****vmtune 値、変更 203****W****WebSphere MQ Transport、パフォーマンスの向上**

キュー、テストとオプション 119  
受信 WebSphere MQ メッセージの実行 119  
送信メッセージとキャッシュ 120  
パフォーマンス追跡情報の設定 120

**WebTemplateVersion パラメータ、および永続的なレイアウトのキャッシュ 62****Web クライアント**

「Siebel Web クライアント」を参照

**Web サーバー**

AIX 用の IBM HTTP Server のチューニング 199

**Windows**

IIS、静的なファイルキャッシュの指定 59

Internet Explorer、推奨設定 57

**Workflow Process Manager**

チューニング 96

パフォーマンスの問題 124

ビジネスサービスのキャッシュ 96

メモリのオーバーヘッドの監視に使用 95

ワークフロー、パフォーマンス追跡情報 124

**あ****アーカイブのログ作成、無効化 153****アーキテクチャ**

アーキテクチャとインフラストラクチャの  
チューニング 16

一般的なフロー、ステップ 19

汎用、グラフィック 15

ユーザーリクエストフロー、プロセスフロー 19

**アーキテクチャ計画**

データベースのサイズ設定のガイドライン 129

データベースレイアウト 130

要件 129

**アクション間隔、ワークフローアクションエージェントの設定 92****アクション間隔、ワークフローモニターエージェントの設定 92****アプリケーションの設定、ネットワーク容量 55****アプレット**

アプレットトグル、パフォーマンス 193

グリッドレイアウト、パフォーマンス 192

メモリコンシューマとしての 36

**い****移動とメモリ 36****インデックス**

EIM テーブル 144

FREELIST パラメータの設定 154

アーカイブのログ作成の無効化 153

オブジェクトの再構築 154

テーブルのキャッシュ 154

テーブルの更新 155

パフォーマンスのための削除 146

**インポート**

備考、EIM の使用 128

## え

## 永続的なビューのレイアウトのキャッシュとプレロード

- 備考、設定 57
- 備考、無効化 58
- ロジック 62

## エージェント、同時コミュニケーションユーザー 71

## エスカレーションアクションリクエストテーブル 89

## エスカレーション状態テーブル 89

## エスカレーションリクエストテーブル 89

## お

## 応答時間、定義 21

## オブジェクト、再構築 154

## 親ビジネスコンポーネント、条件の監視 94

## か

## カーネル設定

- AIX のチューニング 203
- HP-UX 用のチューニング 210
- Solaris 用チューニング 205

## 仮想ビジネスコンポーネントのパフォーマンス、

## 向上 123

## 仮想メモリ管理 203

## 活動レコード

- Siebel Email Response とパフォーマンス 84
- セッションコミュニケーションと  
パフォーマンス 80

## 画面ポップアップのパフォーマンス

- Siebel CTI、向上 79
- 向上 79

## カラム

- 標準カラムの再使用 185
- 標準カラムの再使用の例 186

## き

## キャッシュ

- AOM キャッシュのチューニング 34
- Siebel Configurator のデフォルトの動作 107
- Siebel Configurator、サポートされるタイプ 106
- SQL カーソルキャッシュ 34
- SQL データキャッシュ 34
- WebSphere MQ Transport、送信  
パフォーマンスの向上 120
- Workflow Process Manager による 96
- 永続的なビューのレイアウトのキャッシュ 62
- キャッシュされたビューのメモリへの  
プレロード 63
- 現在のビューレイアウトの取得 64
- ビューのレイアウトのキャッシュ 60
- ビューのレイアウトのキャッシュサイズの設定 62
- ビューのレイアウトのキャッシュを無効にする 64

- ビュー、キャッシュ 65
- ブラウザキャッシュの管理 57
- メモリ内でビューのレイアウトを  
キャッシュする 61

## 行

- 単一バッチの推奨 146

## 共有データベース接続

- プールデータベース接続 38
- プールの設定 41

## く

## クライアント、ハードウェアリソースの提供 56

## グラフィック形式、ログデータの表示 88

## グリッドレイアウト、パフォーマンス 192

## け

## 計算フィールド、ガイドライン 189

## こ

## 顧客設定、最善の方法

- Cascade Delete、設定 174
- Siebel Developer Web クライアント内での  
SQL スプールの指定 175
- Siebel スクリプトのパフォーマンスの  
ガイドライン 180
- Siebel スクリプト、宣言的な手段 179
- SQL クエリープランの例 178
- SQL クエリープラン、トラブルシューティングに  
使用 176
- SQL 追跡ファイル、トラブルシューティングに  
使用 176
- アクセスが可能なビュー、制限 172
- 返されるレコード数、制限 173
- 拡張テーブル、制限 173
- 画面タブ、制限 172
- 結合、制限 173
- スクロールを減らす 174
- チューニングされた PDQ を用意する 174
- データベースに対する SQL クエリー 178
- 内部結合、使用 174
- パフォーマンスのための SQL の分析 175
- ビジネスコンポーネントまたはアプレット  
フィールド、制限 173
- ビジネスサービスのキャッシュ 174
- 必須フィールドの数、制限 173
- ビュー内のビジネスコンポーネント 173
- フィールドの Link Specification プロパティ、  
制限 173
- 不要なボタンの削除 174
- プライマリ ID、制限 173
- ユーザーインターフェイスの設定 172, 173

顧客データの量 81  
 顧客データ、量とパフォーマンス 71  
 子ビジネスコンポーネント、条件の監視 94  
**コミュニケーション**  
 「Siebel Communications Server」を参照  
**コミュニケーションの設定**  
 セッションコミュニケーション、ログの設定 76  
 パフォーマンス 76  
**コンポーネント** 163

## さ

**サードパーティ製品**  
 製品モジュール 70  
 電子メールサーバー 81  
 パフォーマンス 72  
**最初のログイン、およびネットワークに関する**  
 注意事項 55  
**最善の方法**  
 AOM コンポーネント、チューニング 74  
 AOM サーバリソースの節約 75  
 CommSessionMgr コンポーネント、  
 チューニング 74  
 Siebel ARM ファイル、変換 218  
 Siebel Configurator のチューニング 103  
 Siebel EAI のチューニング 118  
 Siebel Email Response のチューニング 83  
 Siebel Web クライアントのチューニング 54  
 顧客設定 172  
 コミュニケーション設定、パフォーマンスの  
 向上 76  
 セッションコミュニケーションのチューニング 73  
 データオブジェクト層 183  
 ビジネスオブジェクト層 188  
 ユーザーインターフェイスオブジェクト層 192  
**最適化**  
 EIM の実装手順 135  
 IBM DB2 UDB 155  
 MS SQL Server 150  
 Oracle データベース 153  
 一般的なガイドライン 134  
**削除**  
 備考、EIM の使用 128

## し

**システムコンポーネントのチューニング** 56  
**実行パラメータ、推奨** 161  
**従業員アプリケーション、およびメッセージバー** 65  
**受信コミュニケーション、説明** 68  
**条件の定義、ガイドライン** 131

## す

**スクリプト**  
 宣言的な手段、使用 179  
 パフォーマンスのガイドライン 180  
**スクリプト、メモリコンシューマとしての** 36  
**スケーラビリティ、説明と例** 14  
**スケジューラ、HP-UX のチューニング** 211  
**スリープ間隔、ワークフローポリシーグループの**  
**設定** 91  
**スループット、定義** 21  
**スレッド**  
 タスクと同じ意味で使用 24  
 定義 21

## せ

**静的なファイルキャッシュ**  
 IBM HTTP Server 上の指定 59  
 Microsoft IIS 上の指定 59  
 Sun Java System Web Server 上の指定 60  
 指定の説明 58  
**セッションコミュニケーション**  
 Communications Configuration Manager 70  
 Communications Inbound Processor 70  
 Communications Inbound Receiver 70  
 Communications Outbound Manager 70  
 Communications Session Manager 69  
 Siebel CTI Connect、画面ポップアップの  
 パフォーマンスの向上 79  
 Siebel eMail Response 80  
 Siebel 製品モジュール 70  
 活動の作成、パフォーマンスへの影響 80  
 画面ポップアップのパフォーマンス、向上 79  
 コミュニケーション設定、パフォーマンスの  
 向上 76  
 サードパーティ製品のモジュール 70  
 最善の方法 73  
 説明 68  
 トポロジ 72  
 パフォーマンスに影響する要因 71  
 ログの設定 76  
**セッション接続、可用性の向上** 78  
**セッションタイムアウトとメモリ** 36  
**接続プール**  
 SISNAPI 接続プールの設定 47  
 SQL 文用 42  
 共有された接続プール、設定例 42  
 専用接続プールのシナリオ 44  
 専用データベース接続 38  
 専用データベース接続プール 43  
 専用データベース接続プールの設定 43

- 専用データベース接続プールの例 44
- 専用データベース接続プールの割り当て 44
- 多重化 41
- 割り当て 42

**設定**

- Siebel Web クライアントのガイドライン 57

**設定パラメータ、MS SQL Server 152****専用サーバー**

- AOM、Siebel Configurator の展開用の設定 102
- CommSessionMgr の実行 72
- Siebel Configurator の実行 102

**専用データベース接続 38****そ****送信コミュニケーション、説明 68****た****高い対話性を持つアプリケーション**

- 設定 57
- ビューのレイアウトのキャッシュ 60

**タスク**

- スレッドと同じ意味で使用 24
- 定義 21

**断片化**

- MS SQL Server の修正 150
- Oracle db テーブルの修正 153

**ち****遅延、定義 21****つ****ツール**

- 「Siebel Tools」を参照

**て****定義済みクエリー (PDQ)、メモリコンシューマとしての 36****データオブジェクト層、最善の方法**

- データベースインデックス、ソートと検索 183
- 標準カラムの再使用 185
- 標準カラムの再使用の例 186
- マルチリンガル LOV クエリーとキャッシュのパフォーマンス 183

**データ管理に関する推奨事項 160****データのエクスポート**

- 備考、EIM の使用 128

**データの並列ロード、テーブルの使用 151****データのマージ**

- 備考、EIM の使用 128

**データベース**

- 計画のガイドライン 129
- レイアウト 130

**データベースインデックス**

- Search Specification プロパティ 185
- Sort Specification プロパティ 184
- ソートと検索 183

**データベースクライアントライブラリ、メモリコンシューマとしての 36****データベース接続**

- AOM の仮定 37
- SISNAPI 接続プールの設定 47
- 共有された接続プール、設定例 42
- 共有接続の割り当て 42
- 共有データベース接続プールの設定 41
- 専用接続プールのシナリオ 44
- 専用データベース接続プール 43
- 専用データベース接続プールの設定 43
- 専用データベース接続プールの例 44
- 専用データベース接続プールの割り当て 44
- プールされていないデータベース接続 37
- プールデータベース接続 38

**データベース認証、データベース接続 42****データ、顧客データ量とパフォーマンス 71****テーブル**

- キャッシュ 154
- 更新 155

**電子メール処理ディレクトリ、管理 83****と****同時コミュニケーションユーザー、およびパフォーマンス 71****同時ユーザー、定義 21****トラブルシューティング**

- EIM テーブルのインデックス 144
- FREELIST パラメータの設定 154
- IBM DB2 UDB の最適化 155
- IBM DB、ローディングプロセス 158
- MS SQL Server 150
- MS SQL テーブルの削除 151
- MS SQL の設定パラメータ 152
- NUM\_IPTABLE\_LOAD\_CUTOFF 148
- Oracle データベース 153
- Oracle データベーステーブルの削除 153
- Siebel Server の監視 161
- SQLPROFILE、使用 142
- SQL クエリープランの例 178
- SQL クエリープラン、トラブルシューティングに使用 176
- SQL 追跡ファイル、トラブルシューティングに使用 176
- SQL の最適化 138
- TempDB の使用 152
- USE ESSENTIAL INDEX HINTS 139
- USE INDEX HINTS 139
- USING SYNONYMS パラメータ 148

アーカイブのログ作成の無効化 153  
 インデックスの削除 146  
 オブジェクトの再構築 154  
 推奨される実行パラメータ 161  
 説明、IBM DB2 UDB for z/OS 158  
 タスクの並列実行 149  
 データ管理に関する推奨事項 160  
 データの並列ロードの使用 151  
 テーブルでの制御 147  
 テーブルのキャッシュ 154  
 テーブルの更新 155  
 適切な統計の作成 145  
 トリガーの無効化 149  
 バッチ、サイズの制御 146  
**トリガー、無効化** 149

**ね**

**ネットワークアプリケーションのデータスループットを最大にする** 196

**ネットワーク容量**  
 アプリケーションの設定 55  
 最初のログイン 55  
 ビューのレイアウトのキャッシュ 55

**は**

**ハードウェアリソース、パフォーマンス** 28  
**ハートビートメッセージ、Push Keep Alive**  
**コミュニケーションドライバの使用** 78  
**バインド変数、潜在的な問題** 177  
**バッチ**  
 サイズの制御 146  
**パフォーマンス**  
 EIM テーブルのインデックス 144  
 EIM の実装手順 135  
 EIM の使用計画 131  
 FREELIST パラメータの設定 154  
 IBM DB2 UDB の最適化 155  
 IBM DB2、パフォーマンスチューニング 159  
 IBM DB2、ローディングプロセス 158  
 MS SQL Server 150  
 MS SQL テーブルの削除 151  
 MS SQL の設定パラメータ 152  
 NUM\_IPTABLE\_LOAD\_CUTOFF 148  
 Oracle データベース 153  
 Oracle データベーステーブルの削除 153  
 Siebel Application Object Manager (AOM)、  
 コンテキスト 15  
 Siebel Communications Server、  
 チューニングの説明 16  
 Siebel Configurator、チューニングの説明 17  
 Siebel CTI、画面ポップアップのパフォーマンスの  
 向上 79  
 Siebel EAI、チューニングの説明 17

Siebel Server の監視 161  
 Siebel Tools、チューニングの説明 17  
 Siebel Web クライアントのテスト 55  
 Siebel Web クライアント、チューニングの  
 説明 16  
 Siebel ワークフロー、チューニングの説明 16  
 SQLPROFILE、使用 142  
 SQL の最適化 138  
 TempDB の使用 152  
 USE ESSENTIAL INDEX HINTS 139  
 USE INDEX HINTS 139  
 USING SYNONYMS パラメータ 148  
 アーカイブのログ作成の無効化 153  
 アーキテクチャ計画の要件 129  
 インデックスの削除 146  
 オブジェクトの再構築 154  
 画面ポップアップのパフォーマンス、向上 79  
 コミュニケーション設定、向上 76  
 サードパーティ製品 72  
 最適化のガイドライン 134  
 推奨される実行パラメータ 161  
 セッションコミュニケーション 71  
 セッションコミュニケーション、最善の方法 73  
 説明と例 14  
 説明、IBM DB2 UDB for z/OS 158  
 タスクの並列実行 149  
 データ管理に関する推奨事項 160  
 データの並列ロードの使用 151  
 データベースのサイズ設定のガイドライン 129  
 データベースレイアウト 130  
 テーブルのキャッシュ 154  
 テーブルの更新 155  
 テーブルのレコードの制御 147  
 適切な統計の作成 145  
 トリガーの無効化 149  
 バッチ、サイズの制御 146  
 用語 21  
 ワークフローアクションエージェント、アクション  
 間隔の設定 92  
 ワークフローエージェント、複数のサーバー上で  
 実行 91  
 ワークフローポリシーグループ、Siebel Server の  
 負荷の管理 90  
 ワークフローポリシーグループ、スリープ間隔の  
 設定 91  
 ワークフローモニターエージェントとワークフロー  
 アクションエージェント 91  
 ワークフローモニターエージェント、アクション  
 間隔の設定 92  
**パフォーマンス集計分析**  
 実行 222  
 タグ 227  
 データ、説明 226

**パフォーマンス追跡情報**

- WebSphere MQ Transport 120
- ワークフロー 124

**パフォーマンスドライバ、Application Object Manager の展開 26****パフォーマンスのテスト**

- Siebel Web クライアント 55

**パラメータ**

- Siebel ARM 214
- Siebel Server の Siebel ARM パラメータ 214

**ひ****ビジネスオブジェクト層、最善の方法**

- Cache Data プロパティ、パフォーマンス 188
- Check No Match プロパティ、パフォーマンス 191
- Force Active プロパティを FALSE に設定する 188
- 計算フィールドのガイドライン 189
- プライマリ ID フィールド、パフォーマンス 190
- プロパティ、ピックアップのパフォーマンスを向上するために使用 190

**ビジネスオブジェクト、EAI Siebel Adapter によるキャッシュ 123****ビジネスコンポーネント**

- Cache Data プロパティ、パフォーマンス 188
- 条件の監視 94

**ビジネスサービス、Workflow Process Manager による呼び出し 96****ピックアップ、パフォーマンスの向上 190****非同期ワークフローモード、利点と欠点 95****ビューのキャッシュ**

- 「ビューのレイアウトのキャッシュ」を参照

**ビューのレイアウトのキャッシュ**

- 永続的なビューのレイアウトのキャッシュ 62
- キャッシュサイズの設定 62
- 現在のビューレイアウトの取得 64
- 説明 60
- ネットワーク容量 55
- ビューのレイアウトのキャッシュを無効にする 64
- ビュー、キャッシュ 65
- メモリ内でビューのレイアウトをキャッシュする 61
- メモリ、キャッシュされたビューのプレロード 63

**ビュー、およびレイアウトのキャッシュ 65****標準カラム**

- 再使用 185
- 再使用の例 186

**標準的な対話性を持つクライアント、および最高のパフォーマンス 56****ふ****ファイル**

- Siebel ARM ファイル、変換 220
- プールされていないデータベース接続 37
- プールデータベース接続 38
- 物理データベースレイアウト 130
- プライマリ ID フィールド、パフォーマンス 190
- ブラウザキャッシュ
  - 管理 57
  - 動作 58
  - ビューのレイアウトのキャッシュ 60
- プロセス、定義と例 21
- プロパティ、ピックアップのパフォーマンスの向上 190

**へ****並列、タスクの実行 149****ベーステーブル**

- データを直接ロード 128

**変換**

- ARM ファイル、最善の方法 218
- Siebel ARM ファイル 220

**変数**

- Siebel ARM 214

**ほ****ポリシー頻度分析ビュー 88****ポリシー、実行されたすべてのログの表示 88****ま****待ち時間およびパフォーマンス 27****待ち時間、定義と例 21****マッピングのガイドライン 132****マルチスレッドサーバー**

- 「マルチスレッドプロセス」を参照

**マルチスレッドプロセス**

- スレッド 24
- 定義 21

**マルチリンガル LOV クエリーとキャッシュのパフォーマンス 183****め****メッセージバー、パフォーマンスの管理 65****メモリ**

- AOM コンポーネントのチューニングのガイドライン 29
- AOM メモリコンシューマ 36
- Configurator キャッシュメモリのサイズ決定 112

Configurator のメモリキャッシュ、設定のための  
パラメータ 109  
Workflow Process Manager でのワークフローの  
実行 95  
キャッシュされたビューのプレロード 63  
定義されたハードウェアリソース 28  
ビューのレイアウトのキャッシュの使用 61  
ワークフローのローカルでの実行 94

**も****モバイル Web クライアント**

Siebel Remote の展開のチューニング 167  
キャッシュパラメータのサイズ設定 112

**ゆ****ユーザーインターフェイスオブジェクト層、最善の方法**

アプレットトグル、パフォーマンス 193  
グリッドレイアウト、パフォーマンス 192

**ユーザーセッション追跡分析**

実行 224  
タグの説明 235  
データ、説明 235

**ユーザーリクエストフロー**

ステップ 19  
プロセスフロー 19

**よ****用語 21****呼び出し参照**

呼び出しグラフの生成データ、説明 232  
呼び出しグラフの生成分析  
タグ 232  
呼び出しグラフの生成、実行 223

**り****リソース**

サーバーリソース、節約 75  
ローカルマシンのリソース 53

**略称**

USING SYNONYMS パラメータ 148

**れ****レイアウトのキャッシュ**

「ビューのレイアウトのキャッシュ」を参照

**レコード**

テーブルでの制御 147

**レポジトリ**

ロードの向上 148

**ろ****ログ**

ワークフローエージェント追跡ログ、使用 88

**ログパラメータ**

AOM 77  
CommSessionMgr 77  
Siebel CTI Connect ドライバ 77

**ログ、Siebel Email Response の設定 84**

論理データベースレイアウト 130

**わ****ワークフロー**

「Siebel ワークフロー」を参照

**ワークフローアクションエージェント**

アクション間隔の設定 92  
定義 91

**ワークフローエージェント**

追跡ログ、使用 88  
複数のサーバー上で実行 91  
プロセス、監視 88

**ワークフロープロセス**

説明 87

**ワークフロープロセス、チューニング**

Search Specification パラメータ、使用の  
最小化 93  
Siebel Business Applications の設定 94  
Workflow Process Manager 96  
親ビジネスコンポーネントと子ビジネスコン  
ポーネント、監視 94  
メモリのオーバーヘッドの監視 94

**ワークフローポリシー**

監視 88  
説明 87  
ログとファイル 88

**ワークフローポリシーグループ**

Siebel Server の負荷を管理するための使用 90  
最適なスリープ間隔、設定 91  
ワークフローモニターエージェントとワークフロー  
アクションエージェント 91

**ワークフローモニターエージェント**

アクション間隔の設定 92  
定義 91  
ポリシー、実行されたすべてのポリシーの表示 88

