

HP Business Availability Center

Windows および Solaris オペレーティング・システム用

ソフトウェア・バージョン : 8.00

TransactionVision デプロイメント・ガイド

ドキュメント発行日 : 2009 年 1 月 (英語版)

ソフトウェア・リリース日 : 2009 年 1 月 (英語版)



利用条件

保証

HP の製品およびサービスの保証は、かかる製品およびサービスに付属する明示的な保証の声明において定められている保証に限ります。本ドキュメントの内容は、追加の保証を構成するものではありません。HP は、本ドキュメントに技術的な間違いまたは編集上の間違い、あるいは欠落があった場合でも責任を負わないものとします。

本ドキュメントに含まれる情報は、事前の予告なく変更されることがあります。

制限事項

本コンピュータ・ソフトウェアは、機密性があります。これらを所有、使用、または複製するには、HP からの有効なライセンスが必要です。FAR 12.211 および 12.212 に従って、商用コンピュータソフトウェア、コンピュータソフトウェアのドキュメント、および商用アイテムの技術データは、HP の標準商用ライセンス条件に基づいて米国政府にライセンスされています。

サードパーティ Web サイト

HP は、補足情報の検索に役立つ外部サードパーティ Web サイトへのリンクを提供します。サイトの内容と利用の可否は予告なしに変更される場合があります。HP は、サイトの内容または利用の可否について、いかなる表明も保証も行いません。

著作権

© 1992 - 2009 Hewlett-Packard Development Company, L.P.

商標

Adobe® および Acrobat® は、Adobe Systems Incorporated の商標です。

Intel®, Pentium® および Intel® Xeon™ は、米国およびその他の国における Intel Corporation またはその子会社の商標または登録商標です。

Java™ は、Sun Microsystems, Inc. の米国商標です。

Microsoft®, Windows®, Windows NT® および Windows XP® は、Microsoft Corporation の米国登録商標です。

Oracle® は、カリフォルニア州レッドウッド市の Oracle Corporation の米国登録商標です。

Unix® は、The Open Group の登録商標です。

SlickEdit® は、SlickEdit Inc. の登録商標です。

文書の更新

本書のタイトル・ページには、次の識別情報が含まれています。

- ソフトウェアのバージョンを示すソフトウェア・バージョン番号
- ドキュメントが更新されるたびに更新されるドキュメント発行日
- 本バージョンのソフトウェアをリリースした日付を示す、ソフトウェア・リリース日付

最新のアップデートまたはドキュメントの最新版を使用していることを確認するには、<http://h20230.www2.hp.com/selfsolve/manuals> を参照します。

このサイトでは、HP Passport に登録してサインインする必要があります。HP Passport ID の登録は、以下の Web サイトにアクセスしてください。

<http://h20229.www2.hp.com/passport-registration.html>

または、HP Passport のログイン・ページの [**New users - please register**] リンクをクリックしてください。

適切な製品サポート・サービスに登録すると、更新情報や最新情報も入手できます。詳細については HP の営業担当にお問い合わせください。

サポート

HP ソフトウェアのサポート Web サイトは、次の場所にあります。

<http://www.hp.com/go/hpsoftwaresupport>

HP ソフトウェアのオンライン・サポートは、インタラクティブな技術サポート・ツールにアクセスするための効率的な手段を提供します。サポート・サイトを利用することで、次のようなことができるメリットがあります。

- 関心のある内容の技術情報の検索
- サポート・ケースおよび機能強化要求の提出および追跡
- ソフトウェア・パッチのダウンロード
- サポート契約の管理
- HP サポートの連絡先の表示
- 利用可能なサービスに関する情報の確認
- ほかのソフトウェア顧客との議論の開始
- ソフトウェアのトレーニングに関する調査と登録

ほとんどのサポート・エリアは、HP Passport ユーザとしての登録およびサインインが必要です。また多くは、サポート契約も必要です。アクセス・レベルの詳細情報については、**http://h20230.www2.hp.com/new_access_levels.jsp** を参照してください。

HP Passport ID の登録は、次の場所で行います。

<http://h20229.www2.hp.com/passport-registration.html>

目次

本書について.....	13
本書の構成.....	14
対象読者.....	15
TransactionVision ドキュメント.....	15
その他のオンライン・リソース.....	17
ドキュメントのアップデート.....	17

第 I 部 : TRANSACTIONVISION の紹介

第 1 章 : TransactionVision の概要	21
TransactionVision について.....	21
アーキテクチャ図.....	23
デプロイメント環境における TransactionVision.....	25
インストール・パッケージ.....	27
下位互換性.....	27
以前のリリースからのアップグレード.....	29
第 2 章 : システム要件の確認	31
サポートされている Analyzer プラットフォーム.....	32
サポートされている TransactionVision UI/Job サーバ・プラットフォーム.....	33
サポートされているデータベース管理システム.....	33
サポートされているメッセージング・ミドルウェア・プロバイダ.....	34
サポートされている WebSphere MQ Sensor プラットフォーム.....	34
サポートされている WebSphere メッセージ・ブローカの構成.....	36
サポートされている Servlet Sensor プラットフォーム.....	37
サポートされている EJB Sensor プラットフォーム.....	38
サポートされている JMS Sensor プラットフォーム.....	39
サポートされている JDBC Sensor プラットフォーム.....	40
サポートされている CICS Sensor プラットフォーム.....	40
サポートされている BEA Tuxedo Sensor プラットフォーム.....	40
サポートされている NonStop TMF Sensor プラットフォーム.....	41
サポートされている .NET Agent プラットフォーム.....	41

サポートされているブラウザの構成	41
LDAP のサポート	41
Java のサポート	42
Flash Player のサポート	42
ローカリゼーションと I18N のサポート	42

第 II 部 : ANALYZER のインストールおよび設定

第 3 章 : TransactionVision Analyzer をインストールする準備	45
TransactionVision Analyzer について	45
デプロイメント環境における TransactionVision Analyzer	46
Analyzer のインストールと設定の概要	46
第 4 章 : Windows への Analyzer のインストール	47
Windows 上での Analyzer インストール・プログラムの起動	47
初めてのインストール	48
アップグレード・インストール	49
Analyzer のアンインストール	50
第 5 章 : UNIX プラットフォームへの Analyzer のインストール	53
インストール・ファイル	53
UNIX 上での Analyzer インストール・プログラムの起動	54
初めてのインストール	55
アップグレード・インストール	55
Analyzer のアンインストール	57
第 6 章 : データベースの設定	59
データベースの設定について	59
サポートされているデータベース	60
データベース・アクセスの設定	60
DB2 の変数の設定	62
Oracle の変数の設定	64
DBMS のパフォーマンスの調整	64
DBMS のディスク領域要件	68
Unicode データに対応するためのデータベース設定	68
第 7 章 : Analyzer の設定	71
Analyzer の設定について	71
TVisionSetupInfo による変更を受けるファイル	72
TVisionSetupInfo が必要とする情報	72
TVisionSetupInfo の実行	78
Analyzer の管理	84
Analyzer のその他の設定	85
イベント・データベースのサイズの縮小	97

第 8 章：Analyzer のログ記録の設定	101
ログ・ファイル	101
循環的なログ記録.....	102
Windows と UNIX のシステム・ログの使用	104
SMTP ログ記録の有効化.....	105
SNMP ログ記録の有効化	107
JMS ログ記録の有効化.....	108

第 III 部：UI/JOB サーバのインストールおよび設定

第 9 章：TransactionVision UI/Job サーバを インストールする準備	113
TransactionVision UI/Job サーバについて	113
デプロイメント環境における TransactionVision UI/Job Server	113
第 10 章：UNIX プラットフォームへの UI/Job サーバの インストール	115
インストール・ファイル	115
UNIX 上での UI/Job サーバ・インストール・プログラムの起動	115
初めてのインストール.....	117
アップグレード・インストール	117
UI/Job サーバのアンインストール	119
第 11 章：Windows への UI/Job サーバのインストール	121
Windows 上での UI/Job サーバ・インストール・プログラムの起動 ..	121
初めてのインストール.....	122
UI/Job サーバのアンインストール	123
第 12 章：UI/Job サーバの設定	125
UI/Job サーバの設定について	125
TVisionSetupInfo による変更を受けるファイル	126
TVisionSetupInfo が必要とする情報	126
TVisionSetupInfo の実行	127
UI/Job サーバの管理.....	132
第 13 章：UI/Job サーバのログ記録の設定	133
ログ・ファイル	133
循環的なログ記録.....	133
トレース・ログ	135
Windows と UNIX のシステム・ログの使用	135

第 IV 部 : SENSOR と AGENT のインストールおよび設定

第 14 章 : TransactionVision Sensor をインストールする準備	139
監視できるアプリケーション.....	140
使用可能な Agent と Sensor の種類.....	141
第 15 章 : Java Agent のインストールと設定	147
Java Agent のインストールと設定について.....	147
Windows での Java Agent のインストールと設定.....	149
UNIX での Java Agent のインストールと設定.....	163
Java Agent のサイレント・インストール.....	172
JRE Instrumenter の実行.....	174
アプリケーション・サーバの設定.....	183
メッセージング・システム・プロバイダの設定.....	184
カスタム・ユーザ・イベントの設定.....	186
第 16 章 : Windows での WebSphere MQ および User Event Sensor のインストール	187
Windows 上でのインストール・プログラムの起動.....	187
初めてのインストール.....	189
アップグレード・インストール.....	190
インストールの変更.....	192
Sensor のアンインストール.....	194
第 17 章 : UNIX プラットフォームでの WebSphere MQ および User Event Sensor のインストール	195
Sensor のインストール.....	196
Sensor のアンインストール.....	199
第 18 章 : i5/OS での Sensor のインストール	201
i5/OS 上でのインストール・プログラムの起動.....	201
第 19 章 : z/OS での Sensor のインストールと設定	203
z/OS 環境での Sensor について.....	204
基本コンポーネントのインストールの概要.....	204
基本コンポーネントのインストール手順.....	205
z/OS での SLD Sensor コンポーネントの設定 :	
CICS, WebSphere MQ Batch, および WebSphere MQ IMS.....	212
z/OS での SLM Sensor コンポーネントの設定 :	
WebSphere MQ CICS Bridge および WebSphere MQ IMS Bridge..	214
背景情報 : CICS 用の WebSphere MQ Sensor.....	216
SLMC を CICS 用に設定する.....	217
背景情報 : WebSphere MQ IMS Bridge Sensor.....	218

第 20 章 : CICS, WMQ Batch, および WMQ-IMS Bridge Sensor の設定	219
概要	220
共通の Sensor コンポーネント	220
CICS Sensor コマンド	222
Sensor の操作	229
バッファ・キューに関する注意点	231
TransactionVision Manager 起動プロシージャ	232
Sensor ドライバ起動プロシージャ	233
第 21 章 : BEA Tuxedo への Sensor のインストールと設定	235
インストールの準備	235
インストールの実行	236
Tuxedo Sensor の再バインド	237
Sensor のアンインストール	237
第 22 章 : .NET Agent のインストールと設定	243
.NET Agent Installer について	244
.NET Agent のインストール	245
.NET Agent の設定	250
IIS の再起動	258
.NET Agent のバージョンを確認する	258
.NET Agent のアンインストール	259
.NET Agents 対応の SSL 設定	259
第 23 章 : NonStop TMF への Sensor のインストールと設定	261
NonStop TMF Sensor について	262
インストールの準備	262
NonStop TMF Sensor のインストール	263
起動と終了	264
NonStop TMF Sensor の設定	265
第 24 章 : WebSphere MQ Sensor の設定	267
WebSphere MQ Sensor ライブラリの設定	267
WebSphere MQ API Exit Sensor の設定	278
WebSphere MQ Sensor と FASTPATH_BINDING	286
WebSphere MQ サンプルでの Sensor の使用	286
WebSphere MQ クライアント・アプリケーションの監視	287
WebSphere MQ-IMS Bridge Sensor の使用	291
WebSphere Business Integration Sensor の使用	297
第 25 章 : Proxy Sensor の設定	299
Proxy Sensor について	299
アプリケーションの要件	300
Proxy Sensor について	300

プロキシ定義ファイルの設定	300
ユーザ・インタフェースの設定	303
第 26 章： Agent および Sensor ロギングの設定	305
ログ・ファイル	305
循環ロギング	306
トレース・ログ	308
複数の Sensor インスタンスについての個別の ログ・ファイルの設定	308
Windows および UNIX システム・ログの使用	310

第 V 部：セキュリティ

第 27 章：セキュリティ	315
TransactionVision におけるセキュリティについて	315
TransactionVision 対応の SSL 設定	316
TransactionVision ユーザ権限の管理	318
標準ロール	321
TransactionVision 認証と Light Weight Single Sign On	322
TransactionVision 設定ファイルのセキュリティ保護	323
TransactionVision Analyzer のセキュリティ保護	325
TransactionVision データベースのセキュリティ保護	327

第 VI 部：付録

付録 A：ユーティリティ・リファレンス	331
CreateSqlScript	332
DB2RunStats	335
DB2Test	337
MigrateDB	338
nanny	339
OracleRunStats	342
OracleTest	344
PassGen	346
rebind_sensor	347
rebind_tux_sensor	349
runSupportSnapshot	350
ServicesManager	353
SetupModule	356
SQLServerTest	357
TVisionSetupInfo	358

付録 B : 設定ファイル	361
Analyzer.properties.....	362
CacheSize.properties	368
Database.properties	368
JobManager.properties.....	372
Sensor.properties	372
SensorConfiguration.xml	372
Setup.properties	374
StatisticsCache.properties.....	375
UI.properties	376
付録 C : データベースの移行	379
時間と領域の要件.....	379
使用されていない統合カラムの無効化	380
カスタマイズされたデータベース・スキーマの移行.....	380
データベースの移行 - 技術的な詳細	380
非統合環境での TransactionVision の最適化	382
付録 D : その他の z/OS 設定	385
RACF 認証.....	385
ファイアウォール設定.....	388
必要な MIPS	388
付録 E : uCMDB ディスカバリ・エージェント	389
uCMDB ディスカバリ・エージェントのインストールと設定	390
uCMDB ディスカバリ・エージェントのコンポーネントと操作.....	403
uCMDB ディスカバリ・エージェントのセキュリティ要件.....	405
uCMDB コマンドの要約.....	409
uCMDB メインフレーム・サービス・エージェントの コンソール・メッセージ	410
uCMDB z/OS のディスカバリ・エラー・メッセージ.....	418
索引	431

本書について

『TransactionVision デプロイメント・ガイド』へようこそ。本書では TransactionVision について紹介し、作業の開始にあたっての詳細、サーバとコンポーネントの設定とインストール、アップグレード・プロセスの詳細についてそれぞれ説明します。

本項の内容

- ▶ 本書の構成 (14 ページ)
- ▶ 対象読者 (15 ページ)
- ▶ TransactionVision ドキュメント (15 ページ)
- ▶ その他のオンライン・リソース (17 ページ)
- ▶ ドキュメントのアップデート (17 ページ)

本書の構成

本書は、次の各部で構成されています。

第 I 部 TransactionVision の紹介

TransactionVision について紹介し、その起動から実行までのスタートアップ・ロードマップと、デプロイメント計画のチェックリストを示します。また、TransactionVision プラットフォームとコンポーネントの概要について説明します。

第 II 部 Analyzer のインストールおよび設定

TransactionVision Analyzer のインストールおよび設定方法について説明します。

第 III 部 UI/Job サーバのインストールおよび設定

UI/Job サーバのインストールおよび設定方法について説明します。

第 IV 部 Sensor と Agent のインストールおよび設定

TransactionVision Sensor と Agent のインストールおよび設定方法について説明します。

第 V 部 セキュリティ

TransactionVision コンポーネントをセキュリティで保護する方法について説明します。

第 VI 部 付録

TransactionVision に関連するユーティリティ、設定ファイルおよびその他の情報を提供します。

対象読者

本書は、次の TransactionVision 利用者を対象としています。

- ▶ アプリケーション開発者または設定者
- ▶ システム管理者
- ▶ データベース管理者

本書の読者は、エンタープライズ・アプリケーションの開発に精通し、エンタープライズ・システムおよびデータベース管理の高度な技術を備えている必要があります。

TransactionVision ドキュメント

TransactionVision のドキュメントは、Business Availability Center の TransactionVision アプリケーションの使用法および Business Availability Center デプロイメント環境の TransactionVision 特有なコンポーネントのデプロイおよび管理に関する情報を提供します。

TransactionVision には次のドキュメントがあります。

- ▶ 『**TransactionVision デプロイメント・ガイド**』は、Business Availability Center デプロイメント環境の TransactionVision 特有なコンポーネントのインストールおよび設定方法について説明します。このガイドは、Business Availability Center 文書ライブラリから PDF 形式で使用できます。
- ▶ 『**Using TransactionVision Guide**』(英語版)は、トランザクションの追跡を行うように TransactionVision を設定する方法およびビジネス・トランザクションの表示方法とレポートのカスタマイズ方法について説明します。このガイドは、TransactionVision Portal または Business Availability Center オンライン文書ライブラリから PDF 形式で使用できます。
- ▶ 『**TransactionVision Planning Guide**』(英語版)は、新規インストールのサイズ設定および計画に関する重要な情報を提供します。このガイドは、HP ソフトウェア製品マニュアル・サイトからダウンロードできます。その他の TransactionVision ドキュメントは、Business Availability Center の次の領域から使用できます。

その他の TransactionVision ドキュメントは、Business Availability Center の次の領域から使用できます。

『**最初にお読みください**』：バージョンの制限事項と最新情報の一覧を示します。HP Business Availability Center DVD のルート・ディレクトリから、readme80.html for Web delivery versions, the HP Business Availability Center release download area をダブルクリックしてください。HP ソフトウェア・サポート Web サイトから最新の readme ファイルにアクセスすることもできます。

「**新機能**」：当該バージョンの新機能と改良点の一覧を示します。HP Business Availability Center で、[ヘルプ] > [新機能] を選択します。

「**オンライン文書ライブラリ**」：文書ライブラリは、HP Business Availability Center の操作方法を説明するオンライン・ヘルプ・システムです。文書ライブラリへは Web ブラウザを利用してアクセスします。閲覧の際の注意事項については、『**HP Business Availability Center デプロイメント・ガイド**』（PDF 形式）の第 6 章「HP Business Availability Center サイトの表示」を参照してください。

文書ライブラリにアクセスするには、HP Business Availability Center で、[ヘルプ] > [文書ライブラリ] を選択します。コンテキスト・センシティブ・ヘルプは、HP Business Availability Center の各ページで [ヘルプ] > [このページのヘルプ] をクリックするか、各ウィンドウで [ヘルプ] ボタンをクリックすることで利用できます。文書ライブラリの使用方法の詳細については、『**Platform Administration**』（英語版）の「Working with the HP Business Availability Center Documentation Library」を参照してください。

その他のオンライン・リソース

トラブルシューティング & ナレッジベース : HP ソフトウェア・サポート Web サイトのトラブルシューティングのページにアクセスします。このサイトからはセルフ・ソルブ技術情報を検索できます。[ヘルプ] > [トラブルシューティング & ナレッジベース] を選択します。この Web サイトの URL は、<http://h20230.www2.hp.com/troubleshooting.jsp> です。

HP ソフトウェア・サポート : HP ソフトウェアのサポート Web サイトにアクセスします。このサイトでは、セルフ・ソルブ技術情報を閲覧できます。また、ユーザ・ディスカッション・フォーラムへの投稿や検索、サポート依頼の送信、パッチや更新されたドキュメントのダウンロードなども行えます。[ヘルプ] > [HP ソフトウェア サポート] を選択します。この Web サイトの URL は、www.hp.com/go/hpsoftwaresupport です。

ほとんどのサポート・ページでは、HP Passport ユーザとして登録してログインすることを求められます。また、多くはサポート契約が必要です。

アクセス・レベルの詳細については、http://h20230.www2.hp.com/new_access_levels.jsp を参照してください。

HP Passport ユーザ ID の登録は、次の場所で行います。

<http://h20229.www2.hp.com/passport-registration.html>

HP ソフトウェア Web サイトからは、HP ソフトウェア Web サイトにアクセスします。このサイトでは、HP ソフトウェア製品に関する最新情報を提供します。新しいソフトウェアのリリース、セミナー、展示会、カスタマー・サポートなどの情報も含まれています。[ヘルプ] > [HP ソフトウェア Web サイト] を選択します。この Web サイトの URL は、<http://welcome.hp.com/country/jp/ja/prodserv/software.html> です。

ドキュメントのアップデート

HP ソフトウェアの製品ドキュメントは、常に更新されています。

最新の更新情報の確認や、使用しているドキュメントが最新版であるかどうかの確認を行うには、HP ソフトウェアの製品ドキュメント Web サイトを参照してください (<http://h20230.www2.hp.com/selfsolve/manuals>)。

はじめに

第 I 部

TransactionVision の紹介

第1章

TransactionVision の概要

本章の内容

- ▶ TransactionVision について (21 ページ)
- ▶ アーキテクチャ図 (23 ページ)
- ▶ デプロイメント環境における TransactionVision (25 ページ)
- ▶ インストール・パッケージ (27 ページ)
- ▶ 下位互換性 (27 ページ)
- ▶ 以前のリリースからのアップグレード (29 ページ)

TransactionVision について

TransactionVision は、システムの全コンポーネント間の相互作用をグラフィックで表示するトランザクション追跡ソリューションです。

HP TransactionVision は、コンピュータ・ネットワークを通じて流れるトランザクションによって生成された個々の電子的イベントを、それらに干渉せずに記録します。さらに重要なことは、特許を取った TransactionVision の「Transaction Contractor アルゴリズム」が、それらのイベントを単一の一貫したビジネス・トランザクションにまとめることです。

HP TransactionVision の主な機能は次のとおりです。

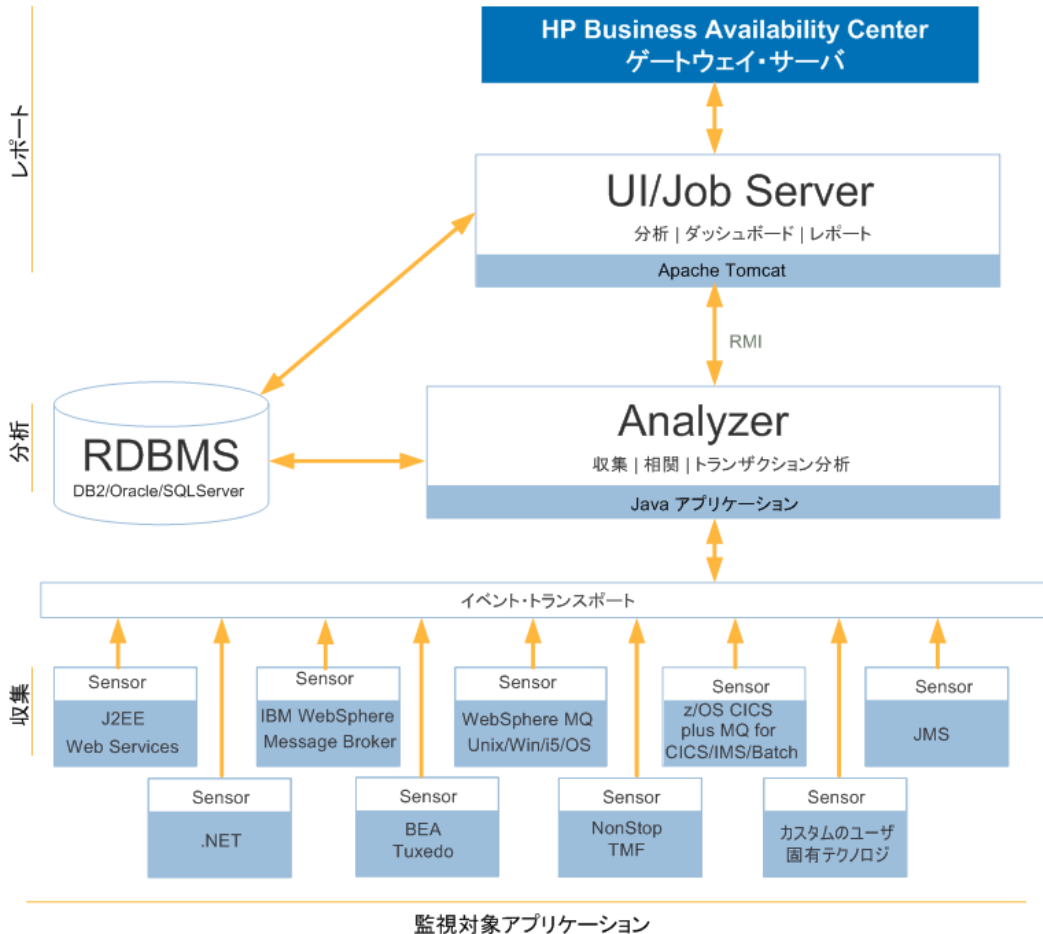
- ▶ 個々のビジネス・トランザクションのエンド・ツー・エンドの可視性
- ▶ 各処理手順全体にわたって各ビジネス・トランザクションを干渉することなく追跡する
- ▶ 自動的にアプリケーション・イベントをビジネス・トランザクションに関連させる

第 1 章・TransactionVision の概要

- ▶ ビジネス・トランザクションの問題の特定および解決にかかる平均時間を短縮する深い可視性。
- ▶ 各トランザクションについて技術データとビジネス・データの両方を収集し、各トランザクションのビジネス・コンテキスト（カスタマ ID、経済的価値など）を特定する。
- ▶ HP Business Availability Center および主要アプリケーション（End User Monitor, Diagnostics, Business Process Insight など）との包括的な統合。

アーキテクチャ図

次の図は、TransactionVision の主要なコンポーネントを示しています。



UI/Job サーバ

TransactionVision UI/Job サーバは、Apache Tomcat Servlet/JSP コンテナのインスタンス上で動作する Web アプリケーションです。このサーバは、通信リンクやデータ収集フィルタなどのデータ収集設定情報を提供するために、Analyzer と通信します。また、プロジェクト分析とレポート結果を表示するために、プロジェクト・データベースにも接続します。

UI/Job サーバのインストールと設定については、第3部を参照してください。

Analyzer

TransactionVision Analyzer は、メッセージング・ミドルウェアを介して TransactionVision Sensor と通信する、Windows 上のサービス（または UNIX 上のデーモン）です。Analyzer は構成メッセージを生成し、それらを指定された構成キューに入れることによって Sensor に配信します。構成メッセージは、Sensor がイベント・メッセージを入れるイベント・キューの名前やプロジェクトのデータ収集フィルタ定義などの Sensor 構成情報を指定します。

標準設定では、TransactionVision はメッセージング・ミドルウェア・プロバイダとして SonicMQ を使用します。TIBCO EMS と WebSphere MQ もサポートされています。

Analyzer はまた、Sensor がイベント・キューに入れたイベントを取得して分析のために処理し、Web ユーザ・インタフェースに表示します。Analyzer は、アンマーシヤル、相関、分析、およびデータ管理機能を実行します。

Analyzer のインストールと設定については、第 2 部を参照してください。

Sensor

TransactionVision Sensor は、分散トランザクションに関わるさまざまなアプリケーションからトランザクション・イベントを収集します。Sensor は、環境内の各コンピュータにインストールされる軽量ライブラリまたは出口プログラムです。

各 Sensor は、インストール先のシステム上での支援テクノロジーによる呼び出しを監視し、それらをフィルタ条件と比較します。呼び出しがフィルタ条件と一致すると、Sensor はその呼び出しに関する入口情報を収集し、その呼び出しを処理のために適切なライブラリに渡します。呼び出しが復帰すると、Sensor はその呼び出しに関する出口情報を収集します。その後、入口情報と出口情報を組み合わせて TransactionVision イベントを生成し、それを指定されたイベント・キューに入れることによって Analyzer に送ります。

.NET Sensor と Java Sensor は、それぞれ .NET Agent および Java Agent と呼ばれます。Agent は、HP Diagnostics Probe と TransactionVision Sensor の機能を 1 つのコンポーネントにまとめたものです。

Sensor と Agent のインストールと設定については、第 4 部を参照してください。

RDBMS (データベース)

TransactionVision は、データを保存するためにサードパーティの RDBMS を使用します。Analyzer は、Sensor が収集したイベントを取得して処理し、イベント関連のテーブルに保存します。スキーマを使ってイベント・データをプロジェクトごとにパーティショニングすれば、各プロジェクトによって収集されたイベント・データへのアクセスを制御できます。

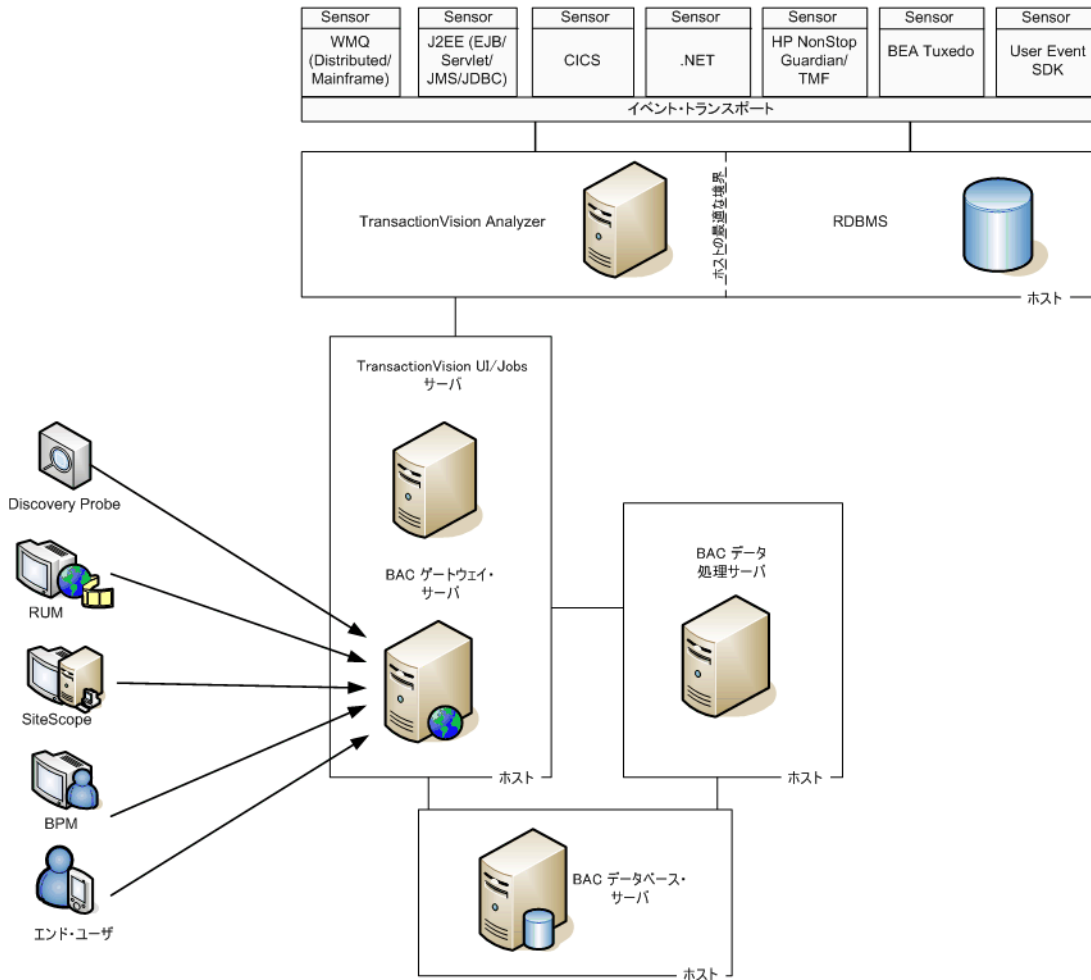
詳細については、59 ページ「データベースの設定」を参照してください。

デプロイメント環境における TransactionVision

TransactionVision は、Business Availability Center デプロイメント環境内で次のように動作します。

- ▶ TransactionVision Analyzer は、Business Availability Center ゲートウェイ・サーバがインストールされているホストとは別のホストにインストールされます。
- ▶ TransactionVision UI/Job サーバは、Business Availability Center ゲートウェイ・サーバ・ホストにインストールされます。
- ▶ TransactionVision が使用するデータベースは、Business Availability Center が使用するデータベースとは別のものであり、TransactionVision Analyzer が最適にアクセスできるホストにインストールされます。

次の図は、典型的な Business Availability Center デプロイメント環境内の TransactionVision コンポーネントを示しています。



複数の BAC ゲートウェイ・サーバがあるデプロイメント環境で（高可用性シナリオをサポートしている場合など）、一度に動作できる TransactionVision UI/Job サーバは 1 つだけです。ほかの BAC ゲートウェイ・サーバ上にミラーリングされた UI/Job サーバをセットアップできますが、その UI/Job サーバは一次 TransactionVision UI/Job サーバが停止するまでは起動できません。

Business Availability Center デプロイメント環境の詳細については、『**HP Business Availability Center デプロイメント・ガイド**』（PDF 形式）を参照してください。

インストール・パッケージ

TransactionVision Analyzer, UI/Job Server, および Sensor は、デプロイメント環境内の Business Availability Center コンポーネントとは別にインストールされます。

TransactionVision コンポーネントは、プラットフォームに固有のパッケージに収められています。本書に記載されているインストール手順の各セットでは、どのインストール・パッケージを使用するかと、それらのパッケージがどこにあるかを明記してあります。

パッケージをインストールする前に、インストール先のシステムが TransactionVision のシステム要件を満たしていることを確認してください。

下位互換性

次の表は、TransactionVision コンポーネントの下位互換性の要約です。

	8.00 Analyzer	8.00 UI/Job Server	8.00 Sensor/A gent	8.00 .NET Agent
7.50 Analyzer				✓
7.50 Web コンポーネント				✓
7.50 Sensor/Agent	✓	✓		
5.0.0 Analyzer				
5.0.0 Web コンポーネント				
5.00 Sensor/Agent	✓	✓		
5.0.0 SPC Analyzer				
5.0.0 SPC Web コンポーネント				
5.00 SPC Sensor/Agent	✓	✓		
5.0.0 SPE Analyzer				
5.0.0 SPE Web コンポーネント				

	8.00 Analyzer	8.00 UI/Job Server	8.00 Sensor/Agent	8.00 .NET Agent
5.00 SPE Sensor/Agent	✓	✓		
5.0.0 SPH Analyzer				
5.0.0 SPH Web コンポーネント				
5.00 SPH Sensor/Agent	✓	✓		

TransactionVision 8.00 Analyzer コンポーネントと UI/Job サーバ・コンポーネントは、以前のリリースからの Analyzer および Web アプリケーション・コンポーネントと互換ではありません。

.NET Agent 以外の TransactionVision 8.00 Sensor は、旧バージョンの TransactionVision Analyzer とともに使用することはできません。8.00 .NET Agent は、7.50 Analyzer でサポートされます。

リリース 5.0.0, 5.0.0 SPC, 5.0.0 SPE, 5.0.0 SPH, および 7.50 からのほかの Sensor および Agent は、TransactionVision 8.0.0 Analyzer および UI/Job サーバで使用できます。ただし、最新の更新と改良を活用できるように 8.0.0 Sensor をインストールすることを強くお勧めします。

本リリースの TransactionVision の新機能については、『**Business Availability Center 最初にお読みください**』を参照してください。

以前のリリースからのアップグレード

リリース 7.50 からの TransactionVision Web コンポーネントおよび Analyzer は、簡単に 8.0 に移行できます。Sensor は、移行可能な設定を含んでいません。

一般的なアップグレード・プロセスは次のとおりです。

- 1 Analyzer インストール・プログラムと UI/Job サーバ・インストール・プログラムを実行します。これらのインストール・プログラムはそれぞれ、以前にホスト上にインストールされたこれらのコンポーネントを検出し、古い設定ファイルのバックアップをとってそれらをリリース 8.00 に移行するためのオプションを提供します。
- 2 旧バージョンの TransactionVision Web ユーザ・インタフェースを実行しているホストの場合は、**TVisionSetupInfo -cleanweb** を実行して古い TransactionVision Web ユーザ・インタフェース・アプリケーションを削除します。

注：以前のリリースからの TransactionVision Web ユーザ・インタフェースは、製品コンポーネントではなくなりました。このユーザ・インタフェースの機能は、Business Availability Center と TransactionVision UI/Job サーバの両方が提供するようになりました。

8.00 では、TransactionVision UI/Job サーバ・コンポーネントが自己充足型の TomCat アプリケーション・サーバとともにインストールされるため、デプロイメント環境において WebSphere または WebLogic は必要なくなりました。

- 3 TransactionVision をインストールして設定した後、**MigrateDB** スクリプトを使ってプロジェクト・データベースを移行します。**MigrateDB** スクリプトを実行する前に、データベースのバックアップを作成しておいてください。
- 4 コンポーネントの移行が完了したら、**TVisionSetupInfo** スクリプトを実行します。
- 5 Windows プラットフォームの場合は **SupervisorStart** を実行し、UNIX プラットフォームの場合は **run_topaz** を実行します。

アップグレードの詳細な手順については、本書のインストールの章を参照してください。

第 1 章・TransactionVision の概要

TransactionVision のアップグレードまたは再インストールの詳細については、『最初にお読みください』の「TransactionVision Known Problems and Issues」の項を参照してください。

第 2 章

システム要件の確認

本章では、HP Business Availability Center プラットフォームの TransactionVision コンポーネントを実行するのに必要なシステム要件について説明します。

注： HP Business Availability Center パッケージに含まれている HP Business Availability Center の readme ファイルには、HP Business Availability Center の現在および以前のバージョンに関する追加のシステム要件が記載されています。

本章の内容

- ▶ サポートされている Analyzer プラットフォーム (32 ページ)
- ▶ サポートされている TransactionVision UI/Job サーバ・プラットフォーム (33 ページ)
- ▶ サポートされているデータベース管理システム (33 ページ)
- ▶ サポートされているメッセージング・ミドルウェア・プロバイダ (34 ページ)
- ▶ サポートされている WebSphere MQ Sensor プラットフォーム (34 ページ)
- ▶ サポートされている WebSphere メッセージ・ブローカの構成 (36 ページ)
- ▶ サポートされている Servlet Sensor プラットフォーム (37 ページ)
- ▶ サポートされている EJB Sensor プラットフォーム (38 ページ)
- ▶ サポートされている JMS Sensor プラットフォーム (39 ページ)
- ▶ サポートされている JDBC Sensor プラットフォーム (40 ページ)
- ▶ サポートされている CICS Sensor プラットフォーム (40 ページ)
- ▶ サポートされている BEA Tuxedo Sensor プラットフォーム (40 ページ)

- ▶ サポートされている NonStop TMF Sensor プラットフォーム (41 ページ)
- ▶ サポートされている .NET Agent プラットフォーム (41 ページ)
- ▶ サポートされているブラウザの構成 (41 ページ)
- ▶ LDAP のサポート (41 ページ)
- ▶ Java のサポート (42 ページ)
- ▶ Flash Player のサポート (42 ページ)
- ▶ ローカリゼーションと I18N のサポート (42 ページ)

注：Java Agent のシステム要件は、それぞれの Java テクノロジ (Servlet, EJB, JMS, および JDBC) ごとに説明されています。

サポートされている Analyzer プラットフォーム

動作環境	WebSphere MQ	TIBCO EMS	SonicMQ
Windows 2003 Server x86 64 ビット	6.0, 7.0	4.2.0, 4.4.2	7.5.2
AIX 5L 5.3, 6.1 POWER	6.0, 7.0	4.2.0, 4.4.2	7.5.2
Solaris 9, 10 SPARC	6.0, 7.0	4.2.0, 4.4.2	7.5.2
RedHat Enterprise Linux 5.0 WS/ES/AS x86 64 ビット	6.0, 7.0	4.2.0, 4.4.2	7.5.2

サポートされている TransactionVision UI/Job サーバ・プラットフォーム

プラットフォーム	動作環境
Microsoft Windows	Windows 2003 Server x86 64 ビット
Sun Solaris	Solaris 9, 10 SPARC

以前のリリースでは、TransactionVision UI/Job サーバ・コンポーネントは TransactionVision Web アプリケーションの一部でした。本リリースでは、TransactionVision ユーザ・インタフェースは Business Availability Center を通じて提供されます。TransactionVision ユーザ・インタフェースにアクセスするためには、Business Availability Center に加えて TransactionVision UI/Job サーバもインストールする必要があります。詳細については、25 ページ「デプロイメント環境における TransactionVision」を参照してください。

サポートされているデータベース管理システム

TransactionVision は以下のデータベースと関連プラットフォームをサポートしています。これらのデータベース・サーバ構成には、DB2 クライアントまたは Oracle oci またはシンクライアント・インタフェースを通じてリモートからアクセスできます。TransactionVision がサポートするのは JDBC 2.0 だけです。

DBMS クライアント / サーバ
DB2 9.1
Oracle 9.2
Oracle 10g
Oracle RAC 10g
Microsoft SQL Server 2005

サポートされているメッセージング・ミドルウェア・プロバイダ

Agent/Sensor	WebSphere MQ	TransactionV ision SonicMQ	SonicMQ	TIBCO EMS	WebLogic JMS	HTTP
Java Agent	✓	✓	✓	✓	✓	
.NET Agent	✓	✓	✓			
WebSphere MQ Agent	✓					
BEA Tuxedo Sensor						✓
NonStop TMF Sensor						✓

サポートされている WebSphere MQ Sensor プラットフォーム

プラットフォーム	動作環境	WebSphere MQ	WMQ API Exit Sensor を サポートする
Microsoft Windows	Windows 2003 Server x86 32 および 64 ビット	6.0, 7.0 (32 ビット) ^{1,6}	はい
Sun Solaris ⁴	Solaris 9, 10 SPARC	6.0, 7.0 ^{1, 6} (32 ビット) 6.0, 7.0 ¹ , (64 ビット)	はい
Hewlett-Packard HP-UX ⁴	HP-UX 11i v3 PA-RISC および Itanium	6.0, 7.0 ^{1, 6} (32 ビット) 6.0, 7.0 ¹ (ライブラリ・ エージェントの場合のみ 64 ビット)	はい
IBM AIX ⁴	AIX 5L 5.3, 6.1 POWER	6.0, 7.0 ^{1,6} (32 ビット) 6.0, 7.0 ^{1, 6} (64 ビット)	はい
RedHat Linux ⁴	Enterprise Linux 5.0 WS/ES/AS x86 32 および 64 ビット	6.0, 7.0 ^{1,6} (32 ビット) 6.0, 7.0 ^{1, 6} (64 ビット)	はい

プラットフォーム	動作環境	WebSphere MQ	WMQ API Exit Sensor をサポートする
IBM i5/OS ²	i5/OS V5R4 iSeries	6.0, 7.0 ^{1, 6}	はい
IBM z/OS ^{3, 5}	z/OS 1.7, 1.8, 1.9 zSeries z/OS 1.7, 1.8, 1.9 Batch z/OS 1.7, 1.8, 1.9 CICS TS 2.x, 3.x z/OS 1.7, 1.8, 1.9 RRS z/OS 1.7, 1.8, 1.9 IMS 7.x, 8.x, 9.x	6.0, 7.0 ¹	N/A

¹ TransactionVision 8.00 は、WMQ 7.0 に対して実行される WMQ アプリケーションを、それらが新しい 7.0 固有の API 機能を使用していない場合にかぎりサポートします。

² i5/OS システム上の Java アプリケーションの監視については、C Library Agent はサポートされません。代わりに API Exit Agent を使用してください。

³ BTTRACE と BTMQEXIT は z/OS ではサポートされません。

⁴ 重要：UNIX システム上でマルチスレッド・アプリケーションを WebSphere MQ とともに使用する場合は、それらのアプリケーションについてスレッド用に十分なスタック・サイズが設定されていることを確認してください。IBM は、マルチスレッド・アプリケーションが MQI 呼び出しを使用する場合には少なくとも 256 KB のスタック・サイズを使用することを推奨しています。TransactionVision WebSphere MQ Agent を使用する場合は、さらに大きなサイズが必要になる可能性があります。推奨スタック・サイズは最小で 512 KB です。詳細については、『**WebSphere MQ Application Programming Guide**』（英語版）の Chapter 7、「Connecting to and disconnecting from a queue manager」を参照してください。

⁵ WebSphere MQ CICS Agent を z/OS に対して使用する場合は、Hewlett-Packard TransactionVision の技術サポートに問い合わせ、その z/OS コンポーネントが最新の最も効率的なバージョンであることを確認してください。

⁶ WebSphere MQ 7.0 の場合は、WebSphere MQ API Exit Sensor を使用するために Fix Pack 7.0.0.1 が必要です。

サポートされている WebSphere メッセージ・ブローカの構成

TransactionVision には、WebSphere メッセージ・ブローカへの、そして同ブローカからの WebSphere MQ 6 API 呼び出しを監視する機能があります。次の表は、TransactionVision がサポートしている WebSphere メッセージ・ブローカ構成を示しています。

動作環境	WebSphere メッセージ・ブローカのバージョン
AIX 5L 5.3, 6.1 POWER	6.1
Windows 2003 Server x86 32 および 64 ビット	6.1

サポートされている Servlet Sensor プラットフォーム

プラットフォーム	動作環境	アプリケーション・サーバ
Microsoft Windows	Windows 2003 Server x86 32 および 64 ビット	IBM WebSphere アプリケーション・サーバ V5.1, 6.0, 6.1 FP9+ ^{1, 2} BEA WebLogic アプリケーション・サーバ 8.1.6, 9.2.3, 10
Sun Solaris	Solaris 9, 10 SPARC	IBM WebSphere アプリケーション・サーバ V5.1, 6.0, 6.1 FP9+ ^{1, 2} BEA WebLogic アプリケーション・サーバ 8.1.6, 9.2.3, 10
IBM AIX	AIX 5L 5.3, 6.1 POWER	IBM WebSphere アプリケーション・サーバ V5.1, 6.0, 6.1 FP9+ ^{1, 2} BEA WebLogic アプリケーション・サーバ 8.1.6, 9.2.3, 10
RedHat Linux	RedHat Enterprise Linux 5.0 WS/ES/AS x86 32 および 64 ビット	IBM WebSphere アプリケーション・サーバ V5.1, 6.0, 6.1 FP9+ ^{1, 2} BEA WebLogic アプリケーション・サーバ 8.1.6, 9.2.3, 10

¹ TransactionVision 8.00 は、IBM WebSphere アプリケーション・サーバのコミュニティ・エディションをサポートしていません。

² TransactionVision 8.00 は、WebSphere アプリケーション・サーバ 6.1 を使用している場合に以前のバージョンで見られたパフォーマンス上の問題を排除するために、FixPack 9 以上を必要とします。

サポートされている EJB Sensor プラットフォーム

プラットフォーム	動作環境	アプリケーション・サーバ
Microsoft Windows	Windows 2003 Server x86 32 および 64 ビット	IBM WebSphere アプリケーション・サーバ V5.1, 6.0, 6.1 FP9+ ^{1, 2} BEA WebLogic アプリケーション・サーバ 8.1.6, 9.2.3, 10
Sun Solaris	Solaris 9, 10 SPARC	IBM WebSphere アプリケーション・サーバ V5.1, 6.0, 6.1 FP9+ ^{1, 2} BEA WebLogic アプリケーション・サーバ 8.1.6, 9.2.3, 10
IBM AIX	AIX 5L 5.3, 6.1 POWER	IBM WebSphere アプリケーション・サーバ V5.1, 6.0, 6.1 FP9+ ^{1, 2} BEA WebLogic アプリケーション・サーバ 8.1.6, 9.2.3, 10
RedHat Linux	RedHat Enterprise Linux 5.0 WS/ES/AS x86 32 および 64 ビット	IBM WebSphere アプリケーション・サーバ V5.1, 6.0, 6.1 FP9+ ^{1, 2} BEA WebLogic アプリケーション・サーバ 8.1.6, 9.2.3, 10

¹ TransactionVision 8.00 は、IBM WebSphere アプリケーション・サーバのコミュニティ・エディションをサポートしていません。

² TransactionVision 8.00 は、WebSphere アプリケーション・サーバ 6.1 を使用している場合に以前のバージョンで見られたパフォーマンス上の問題を排除するために、FixPack 9 以上を必要とします。

サポートされている JMS Sensor プラットフォーム

プラットフォーム	動作環境	JMS サービス・プロバイダ
Microsoft Windows	Windows 2003 Server x86 32 および 64 ビット	WebSphere MQ 6.0 ¹ TIBCO EMS 4.2.0, 4.4.2 SonicMQ 7.5.2 WebLogic JMS 8.1.6, 9.2.3, 10
Sun Solaris	Solaris 9, 10 SPARC	WebSphere MQ 6.0 (32 ビットおよび 64 ビット) ¹ TIBCO EMS 4.2.0, 4.4.2 SonicMQ 7.5.2 WebLogic JMS 8.1.6, 9.2.3, 10
IBM AIX	AIX 5L 5.3, 6.1 POWER	WebSphere MQ 6.0 (32 ビットおよび 64 ビット) ¹ TIBCO EMS 4.2.0, 4.4.2 SonicMQ 7.5.2 WebLogic JMS 8.1.6, 9.2.3, 10
RedHat Linux	RedHat Enterprise Linux 5.0 WS/ES/AS x86 32 および 64 ビット	WebSphere MQ 6.0 (32 ビットおよび 64 ビット) ¹ TIBCO EMS 4.2.0, 4.4.2 SonicMQ 7.5.2 WebLogic JMS 8.1.6, 9.2.3, 10

¹ WebSphere JMS (WebSphere アプリケーション・サーバに組み込まれた JMS) はサポートされていません。

サポートされている JDBC Sensor プラットフォーム

プラットフォーム	データベースのバージョン	動作環境
Oracle	9.2, 10g, RAC 10g	Windows 2003 Server 32 および 64 ビット Solaris 9, 10 AIX 5L 5.3, 6.1 RedHat Enterprise Linux x86 5.0 32 および 64 ビット
DB2	9.1	Windows 2003 Server 32 および 64 ビット Solaris 9, 10 AIX 5L 5.3, 6.1 RedHat Enterprise Linux 5.0 x86 32 および 64 ビット

サポートされている CICS Sensor プラットフォーム

プラットフォーム	動作環境	WebSphere MQ
z/OS	z/OS 1.7, 1.8, 1.9 CICS TS 2.x, 3.x zSeries ¹	6.0, 7.0

¹ z/OS 上の CICS TS とともに TransactionVision を実行する場合は、PTF UK37779 を適用する必要があります（この PTF には実際に問題に対処する APAR PK66562 が含まれています）。UK37779 の前提条件である PTF UK37616 も必要です。

サポートされている BEA Tuxedo Sensor プラットフォーム

プラットフォーム	動作環境	BEA Tuxedo のバージョン
Sun Solaris	Solaris 9, 10 SPARC	8.1 (32 および 64 ビット)
IBM AIX	AIX 5L 5.3 POWER	8.1 (32 および 64 ビット)
HP HP-UX	HP-UX 11i v2, v3 PA-RISC	8.1, 9.1 (32 および 64 ビット)

サポートされている NonStop TMF Sensor プラットフォーム

プラットフォーム	動作環境	TMF
NonStop	Guardian G06.29.02	T8652G08^10JUN2006^TMFCOM^AGL
NonStop	Guardian G06.30	T8608G08^08JAN2007^TMP^AGS

サポートされている .NET Agent プラットフォーム

プラットフォーム	動作環境	.NET
Microsoft Windows	Windows 2003 Server x86 32 および 64 ビット	1.1, 2.0, 3.0 ¹

¹ TransactionVision 8.00 は、.NET 3.0 に対して実行される .NET アプリケーションを、それらが新しい 3.0 固有の API 機能を使用していない場合にかぎりサポートします。

サポートされているブラウザの構成

TransactionVision アプリケーションがサポートしているブラウザ構成は、Business Availability Center がサポートしているものと同じです。『**HP Business Availability Center Hardening Guide**』（英語版）PDF を参照してください。

LDAP のサポート

TransactionVision の LDAP サポートは、Business Availability Center によって管理されます。詳細については、『**HP Business Availability Center Hardening Guide**』（英語版）PDF を参照してください。

Java のサポート

Analyzer コンポーネントと UI/Job サーバ・コンポーネントを実行するために、TransactionVision には Java 1.5 Runtime Environment が含まれています。

JMS, JDBC, Servlet, および EJB Agent 用にサポートされる JVM は、WebSphere アプリケーション・サーバ 5.1/6.0/6.1 および WebLogic アプリケーション・サーバ 8.1.6/9.2.2/10 とともに配付されるバージョンと一致します。

[コンポーネント トポロジアナリシス], [収集されたトポロジ], および [インスタンス トポロジ] ビューを表示するアプレットは、JRE 1.6.0_x を使用します (最新バージョンを推奨します)。

Flash Player のサポート

TransactionVision の一部のレポートとトポロジは、Adobe Flash Player を必要とします。バージョン情報については、『**HP Business Availability Center Hardening Guide**』(英語版) PDF を参照してください。

ローカリゼーションと I18N のサポート

TransactionVision 8.00 のユーザ・インタフェースは、英語、フランス語、日本語、韓国語、および簡体字中国語用にローカライズされました。

TransactionVision 8.00 は I18N に準拠しており、イベント・フィールド、ユーザ・データ、およびレポートの英語以外のロケールでの表示や、英語以外のプロジェクト名およびクエリ名をサポートしています。

TransactionVision の以下の内容はローカライズされていません。

- ▶ 製品名, アプリケーション名
- ▶ ユーザ定義オブジェクト (ジョブ, フィルタ, 通信リンクなど)
- ▶ TVisionSetupInfo および設定ファイル
- ▶ ほとんどの日付形式

第 II 部

Analyzer のインストールおよび設定

第 3 章

TransactionVision Analyzer をインストールする準備

本章の内容

- ▶ TransactionVision Analyzer について (45 ページ)
- ▶ デプロイメント環境における TransactionVision Analyzer (46 ページ)
- ▶ Analyzer のインストールと設定の概要 (46 ページ)

TransactionVision Analyzer について

TransactionVision Analyzer は、TransactionVision Sensor と通信し、Sensor が収集したイベント・データを処理して意味のある分析結果を生成します。Analyzer は、Windows 上では Windows サービスとして実行され、Unix 上ではデーモンとして実行されます。

TransactionVision は、指定されたポート（標準設定は 21100）で RMI（Remote Method Invocation）を使って Analyzer と通信します。TransactionVision UI/Job サーバとコマンドライン・ユーティリティも、自身の動作を制御するため、サービスのシャットダウンを開始するため、収集の開始または停止を許可するため、あるいはステータス情報を取得するために、RMI を使って Analyzer と通信します。

TransactionVision Analyzer がインストールされるホストは、静的 IP アドレスを持っている必要があります。Analyzer が TransactionVision プロジェクトに登録されると、Analyzer のホスト名が解決されて IP アドレスが導き出され、それが Analyzer の一意の識別子として使用されます。詳細については、『**Using TransactionVision**』（英語版）を参照してください。

デプロイメント環境における TransactionVision Analyzer

ほとんどのデプロイメント環境では、TransactionVision Analyzer は Business Availability Center ゲートウェイ・サーバおよび TransactionVision UI/Job サーバとは別のホストにインストールされます。

Analyzer のホストのシステム要件については、32 ページ「サポートされている Analyzer プラットフォーム」を参照してください。

Analyzer のインストールと設定の概要

Analyzer のインストールと設定の一般的な手順は次のとおりです。

- 1 Business Availability Center ゲートウェイ・サーバがインストールされているホストを見つけます。これは、Analyzer のインストール先となるホストです。
- 2 Analyzer のシステム要件を確認します。詳細については、32 ページ「サポートされている Analyzer プラットフォーム」を参照してください。
- 3 Analyzer をインストールします。任意で、以前のリリースから設定を移行します。

Windows オペレーティング・システム上で動作しているホストの場合は、47 ページ「Windows への Analyzer のインストール」を参照してください。

UNIX オペレーティング・システム上で動作しているホストの場合は、53 ページ「UNIX プラットフォームへの Analyzer のインストール」を参照してください。

- 4 (任意) TransactionVision に付属している SonicMQ を使用しない場合は、サポートされているメッセージング・ミドルウェア製品をインストールします。
- 5 データベースをセットアップします。
「データベースの設定」の章を参照してください。

- 6 **TVisionSetupInfo** ユーティリティを実行します。
詳細については、71 ページ「Analyzer の設定」を参照してください。

- 7 必要に応じて、任意の設定を行います。
詳細については、85 ページ「Analyzer のその他の設定」を参照してください。

- 8 Analyzer を起動します。
詳細については、84 ページ「Analyzer の管理」を参照してください。

第4章

Windows への Analyzer のインストール

本章の内容

- ▶ Windows 上での Analyzer インストール・プログラムの起動 (47 ページ)
- ▶ 初めてのインストール (48 ページ)
- ▶ アップグレード・インストール (49 ページ)
- ▶ Analyzer のアンインストール (50 ページ)

Windows 上での Analyzer インストール・プログラムの起動

Analyzer をインストールするには、次の手順を実行します。

- 1 インストール先のシステムに、Administrator として、または Administrator 権限を持つユーザとしてログインします。
- 2 Analyzer を以前のリリースからアップグレードする場合は、JAVA_HOME 環境変数が設定されていることを確認するか、パスに Java が含まれていることを確認してください。
- 3 当該コンピュータで現在実行されている Windows プログラムを、自動バックアップ・プログラムも含めてすべて終了します。ウイルス対策、スパイウェア防止、および脅威対策プログラムを終了する必要はありません。
- 4 Windows エクスプローラから **tvalzr_800_win.exe** をダブルクリックします。
[InstallShield Welcome] 画面が開きます。
- 5 [Next] をクリックし、[TransactionVision Setup Welcome] 画面が表示されるのを待ちます。

- 6 InstallShield の [Save Files] 画面が表示された場合は、[Next] をクリックして標準設定のフォルダ（たとえば C:\TEMP\Hewlett-Packard\TransactionVision）にインストール・ファイルを抽出するか、[Change] をクリックして希望するフォルダを選択して [Next] をクリックします。

そのホストに TransactionVision Analyzer を初めてインストールする場合は、48 ページ「初めてのインストール」に進んでください。インストール先のコンピュータに Analyzer の以前のバージョンがインストールされている場合は、49 ページ「アップグレード・インストール」に進んでください。

初めてのインストール

初めてのインストールの場合は、[Welcome] 画面が表示されます。

- 1 [Welcome] 画面で [Next] をクリックすると、TransactionVision のライセンス契約が表示されます。
- 2 [Yes] をクリックして、ライセンス契約に同意します。[User Information] 画面が表示されます。
- 3 自分の名前と会社名を入力し、[NextNext] をクリックします。[Choose Destination Location] 画面が開きます。
- 4 標準設定のインストール・フォルダ（C:\Program Files\Hewlett-Packard\TransactionVision）を使用する場合は、[Next] をクリックします。別のインストール・フォルダを選択する場合は、[Browse...] をクリックして希望するインストール・フォルダを選択し、[Next] をクリックします。
選択したパッケージが指定した場所にインストールされます。[Setup Complete] ページが表示されます。
- 5 [Finish] をクリックすると、インストールが完了します。

アップグレード・インストール

アップグレード・インストールの場合は、インストール・ウィザードでセットアップ保守メニューが表示されます。

- 1 以前のインストールとは異なる設定で **TransactionVision** をインストールする場合は、以前のインストールを削除するために **[Delete]** を選択して **[Next]** をクリックしてから、インストール手順を再び開始します。以前のリリースからアップグレードする場合は、**[Reinstall]** を選択して **[Next]** をクリックし、以前のインストールからの設定を使って **TransactionVision** をインストールします。**[Configuration File Migration]** ダイアログ・ボックスが表示されます。

- 2 以前のインストールからの設定情報を保持する場合は、**[Yes]** をクリックします。インストール・ウィザードは、既存の設定ファイルのバックアップ・コピーを作成し、新しいバージョンの **TransactionVision** をインストールして、MS-DOS ウィンドウを開き、既存の設定ファイルを新しいバージョンに移行します。この作業が完了すると、**[Setup Complete]** 画面が表示されます。

既存の設定ファイルを上書きする場合は、**[No]** をクリックします。インストールを続行する前に既存の設定ファイルのバックアップ・コピーを作成するかどうかを尋ねるメッセージ・ボックスが表示されます。

バックアップ・コピーを作成する場合は **[Yes]** をクリックし、設定ファイルをバックアップせずにインストールを続行する場合は **[No]** をクリックします。インストール・ウィザードが **TransactionVision** の新しいバージョンをインストールして既存の設定ファイルを上書きし、**[Setup Complete]** 画面を表示します。

- 3 **[Finish]** をクリックすると、インストールが完了します。
- 4 アップグレード・インストールの実行後、**MigrateDB.bat** スクリプトを使って既存のデータベース・スキーマを移行する必要があります。このユーティリティの詳細については、付録 A 「ユーティリティ・リファレンス」を参照してください。

大量のデータを移行する必要がある場合は、付録 C 「データベースの移行」に記載されている移行プロセスの詳細情報を参照してください。

Analyzer のアンインストール

Analyzer をアンインストールするには、次の手順を実行します。

- 1 [スタート] メニューから [**コントロール パネル**] を選択します。
- 2 [**アプリケーションの追加と削除**] をダブル・クリックします。
- 3 アンインストールする HP TransactionVision パッケージを選択し、[**変更と削除**] をクリックします。保守メニュー画面が表示されます。
- 4 [**Remove**] を選択して [**Next**] をクリックし、TransactionVision コンポーネントを削除します。
- 5 指定したパッケージをアンインストールすることを確認し、[**OK**] をクリックします。指定したパッケージがアンインストールされます。次の種類のファイルは削除されません。

▶ インストール後に追加されたファイル

▶ まだインストールされているパッケージに関連付けられた共有ファイル

共有ファイルがインストールされたパッケージと関連付けられていると判断された場合（たとえば、ほかの TransactionVision パッケージがすべてアンインストールされた場合）は、[Locked File Detected] 画面が表示されます。

▶ インストールされた共有ファイルをすべて残す場合は、[Don't display this message again] にチェック・マークを付けて [**No**] をクリックします。

▶ 現在のファイルは残すけれど、ほかの共有ファイルについてはこのメッセージが表示されるようにする場合は、[**No**] をクリックします。

▶ 共有ファイルを削除する場合は、[**Yes**] をクリックします。

Servlet Sensor のアンインストールを実行すると、まず始めにその Sensor が監視している WebSphere アプリケーション・サーバ内で Servlet Sensor がオフになります。\$WAS_HOME/classes の下にある測定対象クラスは、その親ディレクトリが空であれば、親ディレクトリと一緒に削除されます。

- 6 [Uninstall Completed] 画面が表示されます。[**Finish**] をクリックしてアンインストール手順を完了します。

注： TransactionVision の Web ユーザ・インタフェースをアンインストールしたら、その一時キャッシュと配布ディレクトリをクリーンアップする必要があります。WebSphere がこれを自動的に行うことはなく、古いファイルが残っていると新しくインストールしたバージョンが正しく動作しない可能性があります。

第 5 章

UNIX プラットフォームへの Analyzer のインストール

本章の内容

- ▶ インストール・ファイル (53 ページ)
- ▶ UNIX 上での Analyzer インストール・プログラムの起動 (54 ページ)
- ▶ 初めてのインストール (55 ページ)
- ▶ アップグレード・インストール (55 ページ)
- ▶ Analyzer のアンインストール (57 ページ)

インストール・ファイル

次の表は、TransactionVision Analyzer パッケージのインストール・ファイルの名前を配布プラットフォーム別に示しています。

プラットフォーム	ファイル名
AIX	tvalzr_800_aix_power.tar.gz
Linux	tvalzr_800_linux_x86.tar.gz
Solaris	tvalzr_800_sol_sparc.tar.gz

UNIX 上での Analyzer インストール・プログラムの起動

Solaris と Linux では、Analyzer は `/opt/HP/TransactionVision` ディレクトリにインストールされます。AIX では、Analyzer は `/usr/lpp/HP/TransactionVision` にインストールされます。

TransactionVision Analyzer を UNIX プラットフォームにインストールするには、次の手順を実行します。

- 1 TransactionVision のインストール・ファイルがあるディレクトリの場所（DVD デバイスまたはダウンロード・ディレクトリ）に移動します。
注：Solaris と HP-UX では、代わりに DVD デバイスからホストのハード・ドライブ上の一時ファイルにインストール・ファイルをコピーしなければなりません。
- 2 使用するプラットフォームに対応するパッケージを展開します。53 ページ「インストール・ファイル」を参照してください。次に例を示します。

```
gunzip tvalzr_800_sol_sparc.tar.gz
tar xvf tvalzr_800_sol_sparc.tar
```

- 3 スーパーユーザとしてログインします。

```
su
```

- 4 次のコマンドを入力して、インストール手順を開始します。

```
./tvinstall_800_unix.sh
```

パッケージが展開されると、使用可能なコンポーネントを示したメニューが表示されます。次に例を示します。

```
The following TransactionVision packages are available for installation:
```

1. TransactionVision Analyzer
2. TransactionVision UI/Job Server
3. TransactionVision WebSphere MQ Agent
4. TransactionVision User Event Agent

```
99. All of above
q. Quit install
```

```
Please specify your choices (separated by,) by number/letter:
```

インストール先のコンピュータに TransactionVision コンポーネントを初めてインストールする場合は、「初めてのインストール」に進んでください。そのコンピュータに TransactionVision の以前のバージョンがインストールされている場合は、55 ページ「アップグレード・インストール」に進んでください。

初めてのインストール

- 1 **1** を入力して **Return** キーを押します。

インストール・スクリプトがパッケージをインストールして、次のメッセージを表示します。

```
Installation of <TVANLZR> was successful.
```

TransactionVision コンポーネントのメニューが表示されます。

- 2 **q** と入力して **Return** キーを押し、インストール・プロセスを終了します。

このホストに追加のコンポーネントをインストールする場合は、それらのコンポーネントのインストール手順を参照してください。

アップグレード・インストール

- 1 **1** を入力して **Return** キーを押します。

インストール・スクリプトは、Analyzer の以前のバージョンがインストールされていると判断すると、次のメッセージを表示します。

```
There is an earlier version of TransactionVision installed on the system.  
The earlier version has to be uninstalled before installing the current package(s).  
Continue with the uninstallation? (y/n) [n]:
```

- 2 以前のバージョンをアンインストールするために、**Y** を入力して **Return** キーを押します。次のプロンプトが表示されます。

```
Installation has detected a previous installation of TransactionVision.  
Do you want to migrate existing TransactionVision configuration files? (y/n) [y] :
```

- 3 設定を新しいバージョンに移行するために、**Y**を入力して **Return** キーを押します。既存の Analyzer 設定ファイルのバックアップ・コピーが移行のために自動的に作成されます。その後、Analyzer の以前のバージョンがアンインストールされ、TransactionVision コンポーネントのメニューが表示されます。手順4に進みます。

Nを入力して **Return** キーを押すと、将来のために設定ファイルのバックアップ・コピーを作成するためのオプションが提供されます。

Although migration will not be performed at this time, you may optionally back up configuration files from your previous installation for reference purposes. Note that answering N will overwrite these files, causing any existing setup information to be lost. Do you wish to back up configuration files from the previous installation? (y/n) [y] :

設定ファイルのバックアップ・コピーを作成する場合は **Y**を入力して **Return** キーを押します。すると、バックアップの場所を指定するように求められます。

Enter the directory to which existing TransactionVision configuration files should be backed up [/opt/TVision/migrate_tv800_yyyymmdd_HHMMSS]:

標準設定の場所を使用する場合は **Return** キーを押し、それ以外の場所を使用する場合は希望するバックアップ・ディレクトリの場所を入力します。インストール・ユーティリティは次の作業を実行します。

- ▶ 現在の設定ファイルを指定されたディレクトリにコピーします。
- ▶ TransactionVision の旧バージョンをアンインストールします。
- ▶ TransactionVision コンポーネントのメニューを再び表示します。

- 4 **q** と入力して **Return** キーを押し、インストール手順を終了します。

- 5 **MigrateDB.sh** スクリプトを使ってデータベース・スキーマを移行します。このユーティリティの詳細については、付録A「ユーティリティ・リファレンス」を参照してください。

大量のデータを移行する方法については、付録C「データベースの移行」に記載されている移行プロセスの詳細情報を参照してください。

Analyzer のアンインストール

TransactionVision コンポーネントをアンインストールするには、次の手順を実行します。

- 1 スーパーユーザとしてログインします。

```
su
```

- 2 次のコマンドを入力します。

```
./tvinstall_800_unix.sh -u
```

次のメニューが表示されます（実際のオプションはインストールされている TransactionVision コンポーネントによって異なります）。

```
The following TransactionVision packages are available for installation:
```

1. TransactionVision Analyzer
2. TransactionVision UI/Job Server
3. TransactionVision WebSphere MQ Agent
4. TransactionVision User Event Agent

```
99. All of above
```

```
q. Quit install
```

```
Please specify your choices (separated by,) by number/letter:
```

注：実際のオプションと番号は、当該コンピュータ上で使用できるインストール・ファイルによって決まります。

- 3 **1** を入力して **Return** キーを押します。

すべての TransactionVision コンポーネントをアンインストールするには、**99** と入力して **Return** キーを押します。

指定したパッケージがアンインストールされ、再びメニューが表示されます。

- 4 **q** と入力して **Return** キーを押し、インストールを終了します。

第 6 章

データベースの設定

本章の内容

- ▶ データベースの設定について (59 ページ)
- ▶ サポートされているデータベース (60 ページ)
- ▶ データベース・アクセスの設定 (60 ページ)
- ▶ DB2 の変数の設定 (62 ページ)
- ▶ Oracle の変数の設定 (64 ページ)
- ▶ DBMS のパフォーマンスの調整 (64 ページ)
- ▶ DBMS のディスク領域要件 (68 ページ)
- ▶ Unicode データに対応するためのデータベース設定 (68 ページ)

データベースの設定について

TransactionVision Analyzer を設定する前に、DBMS 環境を TransactionVision とともに機能するように正しく設定することが重要です。この設定は、DB2、Oracle、および SQL Server データベースでそれぞれ異なります。

TVisionSetupInfo スクリプトはデータベースを作成せず、既存のデータベースへの接続に依存します。したがって、データベースがアプリケーションにとって許容できるレベルのパフォーマンスで動作するように、データベース変数またはチューニング・パラメータを調整する必要があります。データベースのパフォーマンスのボトルネックの特定と解決を助けるために、TransactionVision とともに複数のツールが提供されています。

サポートされているデータベース

TransactionVision は以下のデータベースと関連プラットフォームをサポートしています。

- ▶ DB2 9.1
- ▶ Oracle 9.2
- ▶ Oracle 10g
- ▶ Oracle RAC 10g
- ▶ Microsoft SQL Server 2005

データベース・アクセスの設定

TransactionVision は、JDBC を通じてデータベースに接続しています。TransactionVision Analyzer および UI/Job サーバを設定するために TVisionSetupInfo ユーティリティを実行すると、データベースの種類 (Oracle, DB2, SQL Server)、ホスト名、データベース名、データベース・ポート、ユーザ名、およびパスワードを入力するように求められます。この入力により、データベースへの JDBC 接続を確立するために使用される < TVISION_HOME > /config/datamgr/Database.properties の設定が変更されます。ユーザ・パスワードは、暗号化された形でこのファイルに保存されます。

TransactionVision はデータベースにアクセスするために、DataDirect 社のサードパーティ JDBC ドライバを使用します。オリジナル・ベンダの JDBC ドライバを使用する必要がある場合は、TransactionVision で使用する JDBC ドライバを手動で設定することができます。

TransactionVision で使用する JDBC ドライバを手動で設定するには、次の手順を実行します。

- 1 < TVISION_HOME > /config/datamgr/Database.properties 内の以下のプロパティを編集します。
 - ▶ プロパティ jdbc_driver を JDBC ドライバ・クラスに設定します。プロパティ・ファイルには、一般的なベンダのドライバのほとんどについて、いくつかの設定例が含まれています。

- ▶ プロパティ `jdbc_url` を完全な JDBC 接続 URL に設定します。プロパティ・ファイルには、一般的なベンダのドライバのほとんどについて、URL の例がいくつか含まれています。

たとえば、DB2 ユニバーサル JDBC ドライバを使用するように `TransactionVision` を設定するには、次のプロパティを設定します。

```
jdbc_driver=com.ibm.db2.jcc.DB2Driver
jdbc_url=jdbc:db2://host:port/dbname:retrieveMessagesFromServerOnGetMessage=true;
```

- 2 JDBC ドライバの jar ファイルを `TransactionVision` の CLASSPATH に追加します。それには、`TVisionSetupInfo` から指定を求められたときに、カスタム CLASSPATH に jar ファイルを追加します。

The Analyzer can optionally be customized by plugin beans in JAR files.

The location of these JAR files needs to be added to the Analyzer CLASSPATH.

Please specify a semicolon delimited list of any additional JAR files you wish to be added to the CLASSPATH. (for example, 'C:¥TVision¥myext.jar;C:¥TVision¥myutil.jar']): C:¥oracle¥10g¥jdbc¥lib¥ojdbc14.jar

これにより、JDBC ドライバの jar が `SetupEnv.bat/sh` 内の CLASSPATH 設定に追加されます。

DB2 の変数の設定

TransactionVision を使用する前に、以下の DB2 変数を表に示された値に設定してください。

変数	値	説明
APP_CTL_HEAP_SZ	1024	最大アプリケーション制御ヒープ・サイズ。 この値は、4 KB ブロックの数を示します。
MAXAPPLS	150	アクティブなアプリケーションの最大数。
APPLHEAPSZ	1024	標準のアプリケーション・ヒープ・サイズ。 この値は、4 KB ブロックの数を示します。
LOCKLIST	システム に基づい た概算値	ロック・リストに割り当てられたストレージの量 (4 KB 単位)。

さらに、バッファ・プールのサイズを増やす必要があります。

APP_CTL_HEAP_SZ, MAXAPPLS, および APPLHEAPSZ

これらの値を設定するには、以下のコマンドを使用します。最後の3つのコマンドは、アクティブなデータベース接続をすべてドロップした後、DB2 サーバを停止して再起動します。これらの手順は、ほかのデータベース・ユーザが影響を受けない適切な時間に実行してください。

```
db2 connect to tvision
db2 get db cfg for tvision
db2 update db cfg for tvision using APP_CTL_HEAP_SZ 1024
db2 update db cfg for tvision using MAXAPPLS 150
db2 update db cfg for tvision using APPLHEAPSZ 1024
db2 force application all
db2stop
db2start
```

LOCKLIST

ロック・リストに必要なページ数を決定するには、次の手順を実行します。

- 1 ロック・リストのサイズの下限を計算します。

$$(512 * 36 * \text{mapappls}) / 4096$$

512 はアプリケーションごとのロックの推定平均数で、36 は既存のロックを持つオブジェクトに対する各ロックに必要なバイト数です。

- 2 ロック・リストのサイズの上限を計算します。

$$(512 * 72 * \text{maxappls}) / 4096$$

72 は、オブジェクトに対する最初のロックに必要なバイト数です。

- 3 データに対する同時実行の量を見積もります。予測に基づいて、上記で計算した上限と下限の範囲内にある LOCKLIST の初期値を選択します。値を設定するには、次のコマンドを使用します。

```
db2 update db cfg for database_name using LOCKLIST n
```

n は LOCKLIST の値です。

- 4 データベース・システム・モニタを使って、このパラメータの値を調整します。詳細については、『**DB2 V7 Administration Guide**』（英語版）の Chapter 32, 「Configuring DB2 Capacity Management」を参照してください。

バッファ・プール

標準設定の DB2 バッファ・プールのサイズはわずか 250 ページです。パフォーマンスを向上させるために、これを 1024 に（DB2 用に十分なメモリが使用できる場合はさらに）増やします。次のコマンドは、標準設定のバッファ・プールのサイズを 250 ページから 1024 ページに変更します。

```
db2 ALTER BUFFERPOOL IBMDEFAULTBP SIZE 1024
```

Oracle の変数の設定

必要なデータベース接続の数が最小限の比較的単純な環境で TransactionVision を使用する予定の場合は、Oracle DBMS 用の特別な設定は不要です。ただし、多数のユーザが同時に Web ユーザ・インタフェースにアクセスする環境、複数の Analyzer が使用される環境、または Analyzer が標準より多くのスレッド数を組み込む環境では、Open Cursors データベース・パラメータの値を大きくする必要があります。この制限を超過すると、TransactionVision の UI または Analyzer のログに、関連するエラー・メッセージが表示されることがあります。Open Cursors の値を大きくするには、次のコマンドを実行します。

```
alter system set open_cursors = 600
```

この変更は、Oracle サーバの実行中に動的に行うことができます。RDBMS を再起動する必要はありません。

DBMS のパフォーマンスの調整

TransactionVision は、データ収集および分析プロセスのために DBMS を排他的に使用するので、TransactionVision の全体的パフォーマンスにとって DBMS のパフォーマンスが重要です。TransactionVision に関連するデータベース操作の大部分は、レコードの挿入とレコードの更新です。したがって、物理ディスクと I/O インタフェースの速度がパフォーマンスに大きな影響を与えます。

I/O スループットの最適化

DBMS の高パフォーマンスへの鍵は、操作のボトルネックである I/O スループットの限界を克服することです。通常、この限界の原因は物理ディスクと I/O インタフェースです。

デプロイの前に、実際の DBMS システムに、I/O とディスクの優れたサブシステムがアタッチされており、そのサブシステムが書き込み用に調整されていることを必ず確認してください。確認事項には、ディスクがパフォーマンスを最適化するように設定された RAID であること、そのディスクの書き込みキャッシュが有効になっていること、I/O インタフェースが高速である（できれば光ファイバ・インタフェースである）ことが含まれます。

I/O の高スループットを達成するには、次のように何らかの並列処理を使用する必要があります。

- ▶ それぞれのプロジェクトに対して別々の DBMS インスタンスを使用します。データをパーティショニングできる場合は、各ホストで別々の DBMS インスタンスを使用することで並列処理を実現できます。このセットアップには、TransactionVision の複数のインスタンスをセットアップする必要があります。
- ▶ テーブルスペース・コンテナとして RAID ディスクを使用します。RAID ディスクは、ハードウェア・レベルでパラレル I/O を提供します。
- ▶ 独立したテーブルスペース・コンテナとログ・ファイル・ディレクトリ。DB2 のログ・ファイル、Oracle のロールバック・セグメント、および SQL Server のトランザクション・ログには、コミットされていないデータベース操作が保持され、データベースの挿入 / 更新の際に高頻度で使用されます。したがって、これらは物理的に独立したディスク（できれば RAID ディスク）上に専用のコンテナを持つ必要があります。
- ▶ テーブルスペースをストライピングします。パラレル I/O をハードウェア・レベルで実現できない場合は、テーブルスペースを複数のディスクにまたがらせるように DBMS をセットアップするという方法があります。これにより、DBMS の領域管理レベルで同時処理を導入できます。

ほかにも多くのデータベース・パラメータが TransactionVision のパフォーマンスに影響を与える可能性があります。特に DB2 の場合は、62 ページ「DB2 の変数の設定」で説明したパラメータが最適化されているかどうかを 1 つ 1 つ検証する必要があります。

TransactionVision が使用するテーブルスペースがデータベースによって直接管理される場合の利点もいくつかあります（DMS または DMS RAW テーブルスペース）。

DBMS のテストとパフォーマンス・ボトルネックの診断

HP は、TransactionVision に関する DBMS のパフォーマンス（特にレコード挿入速度）をテストするために利用できる独立したツールとして、DB2Test、OracleTest、および SQLServerTest を提供しています。これらのツールは Java で作成されており、TransactionVision Analyzer がインストールされる場所で実行する必要があります。これらのツールの詳細については、付録 A 「ユーティリティ・リファレンス」を参照してください。

これらのツールは、TransactionVision が生成するデータベース更新操作をシミュレートします。DBMS のパフォーマンスを完全に把握するために、テストを数回実行してください。テストの結果は、TransactionVision の処理速度と直接の相関関係はなく、むしろ DBMS のパフォーマンスが所定の設定でどの程度良好であるかを示すものです。

初期化プロセスのオーバーヘッドを最小限に抑えるために、それぞれのテストが少なくとも数分は持続するようにしてください。テストは、TransactionVision Analyzer が実行時に使用するデータベースおよびテーブルのセットを対象に実行しなければなりません。

- ▶ 1つのスレッドに対してレコード・サイズ 1 KB で挿入テストを実行します。これにより、イベント挿入の純粋なパフォーマンスを測定できます。
- ▶ 複数のスレッドに対してレコード・サイズ 1 KB で挿入テストを実行します。これにより、複数のスレッドを使用することで挿入のパフォーマンスが向上するかどうかをテストできます。通常、スレッド・カウントは CPU の数の 2 倍に設定されます。
- ▶ 1つのスレッドに対してレコード・サイズ (7 KB + 平均メッセージ・サイズ) で挿入テストを実行します。これにより、分析済みイベントの挿入パフォーマンスを測定できます。
- ▶ 複数のスレッドに対してレコード・サイズ (7 KB + 平均メッセージ・サイズ) で挿入テストを実行します。これにより、複数のスレッドを使用することで挿入のパフォーマンスが向上するかどうかをテストできます。通常、スレッド・カウントは CPU の数の 2 倍から 4 倍に設定されます。
- ▶ Analyzer ホストと DBMS ホストが異なる場合は、上記のテストを Analyzer ホストで実行しなければなりません。ただし、通信 /DB クライアント設定に関連した問題があるかどうかをチェックするために、少なくとも 1 つのテストは DBMS ホストで実行する必要があります。

挿入の速度は、HP によってテストされた類似のシステムが達成した結果と同等でなければなりません。テスト中に、DBMS システムの以下のパラメータを監視する必要があります。

- ▶ 関係するすべての物理ディスク (テーブルスペースおよびログ・ファイル) のディスク I/O 使用率 (特に I/O ビジー状態のパーセンテージ)。
- ▶ 待機時間のパーセンテージを含む、CPU の使用率。

ディスクの使用率が 80 ~ 90% に達するか、I/O 待機時間が異常に長い場合、それはディスク I/O のボトルネックを意味しています。明確なボトルネックが見つからない場合は、DBMS の設定または O/S の設定に問題がある可能性があります。設定に問題が生じている場合は、データベース、DBMS、およびカーネルのパラメータについて HP に確認してください。

DBMS のパフォーマンスを分析するためのもう 1 つの有用なツールは、DB2 パフォーマンス・スナップショット・モニタです。

データベース統計情報の更新

各データベース製品は、テーブルと関連インデックスの物理的特性に関する統計情報を更新するためのツールを提供しています。この特性には、レコードの数、ページの数、平均レコード長が含まれます。データベース・クエリ・オプティマイザは、データへのアクセス・パスを決定するときこれらの統計情報を使用します。

TransactionVision Analyzer によってデータが継続的にデータベースに収集される場合のようにテーブルに多数の更新が加えられたときには、必ずデータベース統計情報が更新されなければなりません。

その結果、TransactionVision のビューおよびレポートによって実行されるクエリと、TransactionVision Analyzer によってイベントを相関させるために内部的に実行されるクエリにおいて、パフォーマンスが大幅に向上する可能性があります。統計情報を定期的に再生成するには、データベース製品が提供する機能 (DB2 の自動保守や Oracle の組込ジョブ・スケジューラなど) を使用することをお勧めします。代わりに、TransactionVision の CreateSqlScript ユーティリティを使って統計情報を生成するための SQL スクリプトを作成し、そのスクリプトを Windows のタスク・スケジューラを使ってタスクとして、または UNIX で cron ジョブとして、手動でスケジュールすることもできます。332 ページ「CreateSqlScript」を参照してください。

DBMS のディスク領域要件

調整する必要がある別の要素として、TransactionVision イベントに必要なディスク・ストレージ領域の量があります。これは、メッセージの平均サイズ、TransactionVision によって収集されるイベントの処理速度、および TransactionVision イベント・データをデータベース内に保持するべき期間によって決まります。

ディスク・ストレージ使用量を算出するための公式は次のとおりです。

(平均メッセージ・サイズ + 7 KB) x イベントの処理速度 x イベントの保持期間

たとえば、平均メッセージ・サイズが 2 KB で、トランザクションの処理速度が 5 トランザクション/秒とします (1 日 8 時間で週 5 日稼働するとして、720,000 トランザクション/週)。TransactionVision データをアーカイブまたは削除するまで 4 週間保存する必要がある場合、必要なストレージの合計は約 52 GB になります ((2 + 7) KB x 720,000 x 2 x 4 = 52 GB)。

Unicode データに対応するためのデータベース設定

TransactionVision は、ビューとレポートで Unicode データを表示できます。しかし、TransactionVision データベースは必要なコード/文字セットで作成する必要があり、TransactionVision 内で適切なデータベース・プロパティを設定しなければなりません。

データベースのコード/文字セット

TransactionVision データベースを作成するときには、次の表に示したプロパティを指定する必要があります。

データベース・プロバイダ	必要な設定
DB2	TransactionVision データベースをコード・セット UTF-8 で作成する必要があります。

データベース・プロバイダ	必要な設定
Oracle	<ul style="list-style-type: none"> ▶ TransactionVision データベースを文字セット AL32UTF8 または UTF8 で作成する必要があります。 ▶ 初期化パラメータ NLS_LENGTH_SEMANTICS は BYTE に設定しなければなりません。
SQL Server	データベースの作成時には特別な設定は必要ありません。

TransactionVision のデータベース・プロパティ

データベースの作成時にデータベース用の正しい文字セットを設定するだけでなく、Database.properties 内の unicode_db プロパティも設定する必要があります。属性 unicode=true が設定された文字ベースの XDM カラムはすべて、指定された数の文字をデータベースに保存できるようにするために、バイト・サイズの2倍で生成されます。1文字に2バイトより多いバイト数が必要な文字セットの場合は、Database.properties file ファイルで unicode_bytes_per_character を1文字あたりに必要なバイト数に設定します。Database.properties ファイルに対する変更の詳細については、付録 B 「設定ファイル」を参照してください。

第7章

Analyzer の設定

本章の内容

- ▶ Analyzer の設定について (71 ページ)
- ▶ TVisionSetupInfo による変更を受けるファイル (72 ページ)
- ▶ TVisionSetupInfo が必要とする情報 (72 ページ)
- ▶ TVisionSetupInfo の実行 (78 ページ)
- ▶ Analyzer の管理 (84 ページ)
- ▶ Analyzer のその他の設定 (85 ページ)
- ▶ イベント・データベースのサイズの縮小 (97 ページ)

Analyzer の設定について

TransactionVision は、Analyzer を設定するプロセスをガイドするために、**TVisionSetupInfo** ユーティリティを提供しています。このユーティリティは、セットアップ・プロセスに必要な情報を入力するように求め、必要な環境変数を設定します。

このユーティリティを実行した後、本章で説明しているように任意でいくつかの設定を実行できます。

すべての設定が完了した後、本章で説明しているように Analyzer を起動する必要があります。

TVisionSetupInfo による変更を受けるファイル

TVisionSetupInfo ユーティリティは、Analyzer を設定するときに以下のファイルに変更を加えます。

- ▶ TVISION_HOME/config/datamgr/Database.properties
- ▶ TVISION_HOME/config/setup/Setup.properties

さらに、**TVisionSetupInfo** は以下の操作を実行します。

- ▶ ソフトウェア・ツールのインストール・パスを **TVISION_HOME/config/setup/DefaultInstallPath.xml** に保存します。
- ▶ **TVISION_HOME/bin/SetupEnv.[sh|bat]** を生成します。このファイルは、JAVA_HOME, CLASSPATH, および TransactionVision が必要とするシステム・ライブラリ・パス環境変数を設定するために、TVisionSetup によって使用されます。

TVisionSetupInfo が必要とする情報

TVisionSetupInfo ユーティリティは、設定に必要な以下の情報を入力するように求めます。**TVisionSetupInfo** を実行する前に必要な情報を確認して、適切な情報を入力できるようにしてください。

ライセンス情報

TransactionVision のライセンスを入力するように求められます。

TVisionSetupInfo を実行した後でライセンスを適用する場合は、`< TVISION_HOME > /config/license/License.properties` を変更します。

ログ・ファイルの場所

TransactionVision が Analyzer のログ・ファイルを保存することになるディレクトリのパス名を入力する必要があります。標準設定の値は `TVISION_HOME/logs` です。詳細については、第8章「Analyzer のログ記録の設定」を参照してください。

データベース情報

TransactionVision で使用するデータベースの種類（IBM DB2, Oracle, または SQL Server）を指定する必要があります。

注 : TVisionSetupInfo ユーティリティは、DB2 ユニバーサル JDBC ドライバ（タイプ 4）または Oracle シンクライアント・ドライバを自動的にセットアップします。DB2 ユニバーサル・ドライバ（タイプ 2）や Oracle oci ドライバなど、ほかのドライバを使用する方法については、87 ページ「Analyzer とともに DB2 タイプ 2 または Oracle oci JDBC ドライバを使用する設定」を参照してください。

次の表は、収集されるデータベース・パラメータを、サポートされているデータベースの種類別に示しています。各パラメータの詳細は、表の後に記載されています。

パラメータ	データベース		
	DB2	Oracle	SQL Server
ログ・ファイルの場所			
データベース・インスタンス名	✓		✓
データベース接続名	✓	✓	
データベース名	✓	✓	✓
データベース・ホスト	✓	✓	✓
データベース・ユーザ名	✓	✓	✓
データベース・ユーザ・パスワード	✓	✓	✓
データベースのリリスナ・ポート	✓	✓	✓
データベースのインストール・パス	✓	✓	
JDBC ドライバのインストール・パス	✓	✓	✓

DB2 データベース固有の情報

DB2 データベースの場合は、以下の情報を入力する必要があります。

- ▶ DB2INSTANCE 環境変数の値。標準設定の値は DB2 です。WebSphere 5 サーバを起動する前に、DB2INSTANCE 環境変数を設定する必要があります。この変数の設定に失敗すると、TransactionVision の起動が失敗し、WebSphere サーバの起動と初期化が失敗します。
- ▶ TransactionVision が使用するデータベース接続の名前。接続名とデータベース名を調べる方法については、第6章「データベースの設定」を参照してください。
- ▶ TransactionVision が接続するデータベースの名前。クライアント・データベース接続が使用されている場合は、この名前が `database_connection_name` と異なることがあります。接続名とデータベース名を調べる方法については、第6章「データベースの設定」を参照してください。
- ▶ データベースのホストの名前。標準設定の値は `local_host` です。
- ▶ データベースに接続するためのユーザ名。このフィールドが空の場合は、現在ログインしているユーザを使ってデータベース接続が確立されます。指定したユーザまたは現在ログインしているユーザが、データベースにアクセスする権限を持っていることを確認してください。
- ▶ 指定したデータベース・ユーザ名のパスワード。このフィールドが空の場合は、現在ログインしているユーザのパスワードを使ってデータベース接続が確立されます。TVisionSetupInfo は、DES 暗号化アルゴリズムをサポートするインストール済みの JCE プロバイダを検出できた場合には、Database.properties ファイル内の `jce_provider` の値を検出した JCE プロバイダのクラス名に設定し、パスワードを暗号化します。検出できなかった場合は、パスワード暗号化機能を無効にし、パスワードを暗号化されていないテキストとして保存します。
- ▶ データベースのポート番号。これは、データベース・サーバがリッスンしている TCP/IP ポートです。一般的な値は 50000 です。正しい値を調べるには、DB2 管理サーバの稼働中に、DB2 コントロール・センターから次のメニュー項目を選択します。

[すべてのシステム] > システム名 > [インスタンス] > インスタンス名

右クリックして、ショートカット・メニューから [Setup Communications] を選択します。[プロパティ] をクリックすると、そのデータベース・インスタンスが使用しているポート番号がわかります。

Oracle データベース固有の情報

Oracle データベースの場合は、以下の情報を入力する必要があります。

- ▶ TransactionVision が使用するデータベース接続の名前。標準設定の値は TVISION です。接続名とデータベース名を調べる方法については、第6章「データベースの設定」を参照してください。
- ▶ TransactionVision が接続するデータベースの名前。標準設定の値は TVISION です。クライアント・データベース接続が使用されている場合は、この名前が `database_connection_name` と異なることがあります。接続名とデータベース名を調べる方法については、第6章「データベースの設定」を参照してください。
- ▶ データベースのホストの名前。標準設定の値は `local_host` です。
- ▶ データベースに接続するためのユーザ名。このフィールドが空の場合は、現在ログインしているユーザを使ってデータベース接続が確立されます。指定したユーザまたは現在ログインしているユーザが、データベースにアクセスする権限を持っていることを確認してください。
- ▶ 指定したデータベース・ユーザ名のパスワード。このフィールドが空の場合は、現在ログインしているユーザのパスワードを使ってデータベース接続が確立されます。TVisionSetupInfo は、DES 暗号化アルゴリズムをサポートするインストール済みの JCE プロバイダを検出できた場合には、Database.properties ファイル内の `jce_provider` の値を検出した JCE プロバイダのクラス名に設定し、パスワードを暗号化します。検出できなかった場合は、パスワード暗号化機能を無効にし、パスワードを暗号化されていないテキストとして保存します。
- ▶ データベースのポート番号。これは、データベース・サーバがリッスンしている TCP/IP ポートです。一般的な値は 1521 です。正しい値を調べるには、サーバ側で `lsnrctl` コマンドを実行します。status コマンドを使用して、ポート番号を突き止めます。

SQL Server データベース固有の情報

SQL Server データベースの場合は、以下の情報を入力する必要があります。

- ▶ データベースのホストの名前。標準設定の値は `local_host` です。
- ▶ SQL Server インスタンスの名前。複数のサーバ・インスタンスを実行していないかぎり、これは通常は空でなければなりません。
- ▶ SQL Server 上の接続先のポート番号。

- ▶ TransactionVision が接続する SQL Server データベースの名前。標準設定の値は TVISION です。
- ▶ データベースに接続するためのユーザ名。このフィールドが空の場合は、現在ログインしているユーザを使ってデータベース接続が確立されます。指定したユーザまたは現在ログインしているユーザが、データベースにアクセスする権限を持っていることを確認してください。
- ▶ 指定したデータベース・ユーザ名のパスワード。このフィールドが空の場合は、現在ログインしているユーザのパスワードを使ってデータベース接続が確立されます。TVisionSetupInfo は、DES 暗号化アルゴリズムをサポートするインストール済みの JCE プロバイダを検出できた場合には、Database.properties ファイル内の jce_provider の値を検出した JCE プロバイダのクラス名に設定し、パスワードを暗号化します。検出できなかった場合は、パスワード暗号化機能を無効にし、パスワードを暗号化されていないテキストとして保存します。

LW-SSO (Lightweight Single Sign-on) の設定

Business Availability Center からのアクセスを認証するために必要な LW-SSO 設定を指定します。

メッセージ・ミドルウェア・プロバイダ情報

TransactionVision は、Sensor と Analyzer の間での通信用に SonicMQ を提供しており、Analyzer がインストールされるときに Progress SonicMQ ソフトウェアがインストールされます。この付属バージョンの SonicMQ を使用する場合は、メッセージ・システム・プロバイダ情報を入力する必要はありません。

注：BEA Tuxedo または Non-Stop TMF Sensor 用の、あるいは RUM との統合用のメッセージング・システム・プロバイダは、組み込みの SonicMQ でなければなりません。

TIBCO EMS, WebSphere MQ, および SonicMQ を Analyzer とは別にインストールすることも可能です。その場合は、そのうちの1つをメッセージング・ミドルウェア・プロバイダとして指定できます。これらのメッセージ・システム・プロバイダのうちの1つを使用する場合は、インストール・パスと以下に示す追加情報を入力するように求められます。

WebSphere MQ のセットアップの検証

WebSphere MQ をメッセージング・ミドルウェア・プロバイダとして使用する場合は、Analyzer が指定されたキュー・マネージャと通信できることを検証するように求められます。サーバまたはクライアントの接続をテストする必要があるかもしれません。

以下の情報が必要です。

- ▶ WebSphere MQ キュー・マネージャのホスト名
- ▶ WebSphere MQ キュー・マネージャのチャンネル名
- ▶ WebSphere MQ キュー・マネージャのポート番号
- ▶ WebSphere MQ キュー・マネージャの CCSID（わからない場合は空白のままにしてください）

TIBCO のセットアップの検証

TIBCO をメッセージング・ミドルウェア・プロバイダとして使用する場合は、Analyzer が指定されたホストと通信できることを検証するように求められます。以下の情報が必要です。

- ▶ TIBCO EMS サーバのホスト名
- ▶ TIBCO EMS サーバのポート番号
- ▶ ユーザ名
- ▶ パスワード

Agent と Sensor の種類

ユーザ・イベント、.NET、NonStop TMF、および Tuxedo イベントは常に収集されますが、ほかのテクノロジーからのイベント収集は有効化/無効化できます。

環境内で使用し、それについてイベントを有効にする Sensor の種類を指定します。次のオプションがあります。

- ▶ CICS
- ▶ EJB
- ▶ Servlet
- ▶ TIBCO EMS, SonicMQ JMS, または WebLogic JMS

- ▶ WebSphere MQ
- ▶ WebSphere MQ IMS Bridge
- ▶ WebSphere MQ JMS
- ▶ JDBC
- ▶ 上記のすべて

Analyzer をカスタマイズするためのプラグイン・ビーン

Analyzer の CLASSPATH に追加する必要がある jar ファイルの場所。これらの jar ファイルには、Analyzer をカスタマイズするプラグイン・ビーンが収められています。

Business Availability Center への登録

TransactionVision コンポーネントが使用する Business Availability Center ゲートウェイ・サーバの場所を指定する必要があります。

TVisionSetupInfo の実行

- 1 インストール先のシステムに、root, Administrator, または Administrator 権限を持つユーザとしてログインします。
- 2 TVisionSetupInfo.[sh|bat] と入力して TVisionSetupInfo スクリプトを実行します。

オペレーティング・システム	スクリプト・コマンド
AIX, Linux, Solaris	< TVISION_HOME > /bin/TVisionSetupInfo.sh
Windows	< TVISION_HOME > %bin%TVisionSetupInfo.bat

注： TVISION_HOME 環境変数はインストール時に、インストール・ディレクトリの絶対パスに設定されます。たとえば、TVISION_HOME は Solaris では /opt/HP/TransactionVision, AIX では /usr/lpp/HP/TransactionVision のようになります。

TVisionSetupInfo は、以下を設定するために必要な情報を入力するように求めます。

- ▶ ログ・ファイル
- ▶ データベースのプロパティ
- ▶ サードパーティのメッセージング・ミドルウェア・プロバイダの情報
- ▶ ターゲット環境内の Agent と Sensor
- ▶ Business Availability Center への登録

TVisionSetupInfo からの入力要求に対して情報を入力するときに、大括弧で囲まれた標準設定の値を受け入れるには、**Enter** キーを押します。受け入れない場合は、正しい値を入力して **Enter** キーを押します。空の値を指定するには、スペースバーを押してから **Enter** キーを押します。以降の各項では、入力の例が斜体で示されています。

TVisionSetupInfo は、各設定セクションでの設定が完了するたびに、どのファイルが更新されたかを知らせるメッセージを表示します。

例

Windows での **TVisionSetupInfo** セッションの例を次に示します。

This program collects configuration information in order to set up the TransactionVision environment. This includes:

- Location of software that TransactionVision depends upon such as the messaging middleware and the relational database system
- Parameters required to connect to the messaging middleware
- Parameters to connect to the database system
- Setup parameters for installed TransactionVision components

You will be prompted to input required configuration parameters.

If a default value is provided in [], pressing <Enter> will set the parameter to this default value. Pressing <Space><Enter> will set the parameter to an empty value.

Please specify name of the directory where you want to store your log files

[C:/PROGRA~1/HP/TRANSA~1\logs]:

TransactionVision Info(FileUpdated): File

"C:\PROGRA~1\HP\TRANSA~1\config\setup\Setup.properties" has been successfully updated

Modifying *.Logging.xml files to use log file directory

C:/PROGRA~1/HP/TRANSA~1\logsTransactionVision Info(FileUpdated): File

"C:\PROGRA~1\HP\TRANSA~1\config\logging\bpitveventimporter_loggingconfig.properties" has been successfully updated

Please provide your TransactionVision license key: PreSalesTraining@hp.com-020EF5-815082021D0482C

TransactionVision Info(FileUpdated): File

"C:\PROGRA~1\HP\TRANSA~1\config\license\License.properties" has been successfully updated

Database Settings

Retrieving database configuration parameters...

Type of Database? (DB2/Oracle/SQLServer) [Oracle]:

Name of the host the database is running on [your_database_host]: ros89891duh.rose.hp.com

Name of database TransactionVision connects to.

This is the Oracle database SID [your_database_name]: orcl

Enter the port number that the database listener is on [1521]:

Database user name []:system

User password []: rpctest

TransactionVision Info(FileUpdated): File

"C:\PROGRA~1\HP\TRANSA~1\config\datamgr\Database.properties" has been successfully updated

LW-SSO (Lightweight Single Sign-On) Settings

Enter the LW-SSO init string, used for the initialization of the enc algorithm. This value needs to be the same in all applications integ with LW-SSO. Unless you have modified the default value in other app such as BAC, you can just accept the default. [Xy6stqZ]:

Enter the LW-SSO application domain, used for LW-SSO cookie creation. This is the full or partial domain part of the fully-qualified hostn of the machine running the TransactionVision UI. (for example, 'mydomain.com') [hpqcorp.net]:

Enter a comma delimited list of LW-SSO protected domains. Applicable if other applications integrated with LW-SSO are running in different domains. It is not necessary to include the TransactionVision UI doma (for example, if BAC is in domain bac.com and Diagnostics is in domadiag.com, enter

'bac.com,diag.com'): TransactionVision Info(FileUpdated): File

"C:\PROGRA~1\HP\TRANSA~1\cproperties" has been successfully updated

Generating script file for environment setup...

TransactionVision Info(NewFileCreated): File "C:\PROGRA~1\HP\TRANSA~nv.bat" has been successfully created

Initialize and verify database setup for TransactionVision

Do you wish to initialize the TransactionVision Database and add communication links for SonicMQ on this Analyzer ? [y]:

Default TVISION commlink has been added

Default HTTP commlink has been added

Verifying database initialization for TransactionVision...

TransactionVision Info(DBInitialized): Database has been properly initialized

Register TransactionVision with Business Availability Center

Do you wish to register TransactionVision with Business Availability Center at this time? [n]: y

Note: If your BAC server has Lightweight Single Sign On enabled, the hostnames of the TransactionVision UI and BAC servers must be a fully qualified domain name.

Please enter the hostname of the BAC server [] : ovresx1-vm11.rose.hp.com

Please enter the port of the BAC server [80] :

Do you wish to configure TransactionVision to access BAC using HTTPS? [n] :

TransactionVision has been configured to use the BAC server at: http://ovresx1-vm11.rose.hp.com:80

Do you wish to register the TransactionVision UI with BAC? [y] :

Note: If your BAC server has Lightweight Single Sign On enabled, the hostnames of the TransactionVision UI and BAC servers must be a fully qualified domain name.

Enter TransactionVision UI host [ROS89891DUH.americas.hpqcorp.net] :

Do you wish to configure BAC to access TransactionVision using HTTPS? [n] :

Enter TransactionVision UI port [21000] :

Do you wish to configure BAC to access TransactionVision using HTTPS? [n] :

Enter TransactionVision UI port [21000] :

Do you wish to register a TransactionVision Analyzer for RUM Data Publishing with BAC? [y] :

Enter the TransactionVision Analyzer host for RUM Data Publishing [ROS89891DUH.americas.hpqcorp.net] :

Enter RUM data publishing port [21113] :

User name for Basic Authentication to TransactionVision Analyzer
used by Real User Monitor :

Password for Basic Authentication to TransactionVision Analyzer used by Real User
Monitor :

The BAC server at `http://ovresx1-vm11.rose.hp.com:80` will be configured with:

TransactionVision Web Server:
`http://ROS89891DUH.americas.hpqcorp.net:21000`
TransactionVision URL for RUM:
`http://ROS89891DUH.americas.hpqcorp.net:21113/tv_rum`
TransactionVision User for RUM:
TransactionVision Password for RUM:

Please review the above settings and verify that they are correct.

Proceed with registration to BAC? [y] :

Registration with BAC was successful.

TransactionVision Info(TVisionSetupInfoSuccess): TVisionSetupInfo has completed
successfully.

All program output and user input has been logged to
"`C:/PROGRA~1/HP/TRANSA~1/logs/setup.log`".

Please start HP Business Availability Center by running
`%TVISION_HOME%\bin\SupervisorStart.bat`

完了情報

設定が完了すると、TVisionSetupInfo は次のメッセージを表示します。

TransactionVision Info(TVisionSetupInfoSuccess): TVisionSetupInfo has completed
successfully.

Business Availability Center の再起動

TVisionSetupInfo の実行後に、Business Availability Center を再起動する必要がある場合があります。再起動が必要な場合は、**TVisionSetupInfo** スクリプトが使用するべき適切なコマンドを次のように表示します。

Windows の場合

```
Please start HP Business Availability Center by running  
%TVISION_HOME%\bin\%SupervisorStart.bat
```

UNIX の場合

```
Please start HP Business Availability Center by running  
$TVISION_HOME/bin/run_topaz start
```

Analyzer の管理

Analyzer サービスは、HP Business Availability Center サービスによって管理されます。『**HP Business Availability Center デプロイメント・ガイド**』（PDF 形式）を参照してください。

Sensor に対して収集停止メッセージを送信せずに Analyzer を終了するには（Sensor がイベントの収集を続けられるようにするために）、次のコマンドを使用します。

```
ServicesManager -exit -keepcollect
```

ただし、Analyzer から送信される新しい構成メッセージがなくなるため、Sensor がイベントの収集を続けるのは、最後の構成メッセージが期限切れになるまでです。構成メッセージの期限は、TransactionVision の Analyzer 設定ページで変更できます。

Analyzer のその他の設定

必要に応じて、以下の構成を実行できます。

- ▶ SonicMQ Broker のヒープ設定の変更
- ▶ メッセージの期限の設定
- ▶ Analyzer のスレッド・カウントの設定
- ▶ エラー・ログ
- ▶ Analyzer とともに DB2 タイプ 2 または Oracle oci JDBC ドライバを使用する設定
- ▶ マルチスレッド・サブレット /JMS イベント
- ▶ 不要な Analyzer ビーンの有効化
- ▶ ローカル・トランザクション・マッチング
- ▶ Analyzer の失敗モード
- ▶ JDBC Sensor のデータベース解決
- ▶ .NET Agent に関する最適化

SonicMQ Broker のヒープ設定の変更

- 1 次のスクリプトを使用して SonicMQ 管理コンソールを起動します。
< TVISION_HOME > ¥Sonic¥MQ7.5¥bin¥startmc.bat (Windows)
< TVISION_HOME > /Sonic/MQ7.5/bin/startmc.sh (UNIX)
- 2 [SonicMQ Management Console] ウィンドウの上部にある [**Configure**] タブをクリックします。
- 3 [Containers] という名前が付いた 2 番目の [+] ボタンをクリックして、SonicMQ で設定されているコンテナのリストを展開します。
- 4 ホスト名と同じ名前のコンテナを選択し、右クリックしてドロップダウン・メニューを表示して、[**Properties**] を選択します。
- 5 [Edit Container Properties] ウィンドウで、[Environment] という名前のタブをクリックします。
- 6 [Java VM Options] という名前のボックスで、標準設定の -Xms32m -Xmx256m を変更して、ヒープ・サイズの設定を増やします。
- 7 管理コンソールを終了します。

- 8 次のコマンドを使って nanny をシャットダウンします。

< TVISION_HOME > %bin%SupervisorStop.bat (Windows)

< TVISION_HOME > /bin/run_topaz stop (UNIX)

- 9 次のコマンドを使って nanny を再起動します。

< TVISION_HOME > %bin%SupervisorStart.bat (Windows)

< TVISION_HOME > /bin/run_topaz start (UNIX)

SonicMQ Broker ヒープ・サイズの設定の詳細については、『**Progress SonicMQ Performance Tuning Guide**』を参照してください。

メッセージの期限の設定

構成メッセージの期限は、「孤児」構成メッセージによって生成されるイベントの量が大きな運用上の問題を引き起こさずにイベント・キューに収まるようにするために、十分小さな値に設定しなければなりません。詳細については、『**Using TransactionVision**』（英語版）を参照してください。

Analyzer のスレッド・カウントの設定

Analyzer のスレッド・カウントは、DBMS 挿入テストの結果と一致するように設定しなければなりません。詳細については、『**Using TransactionVision**』（英語版）を参照してください。

エラー・ログ

稼働中の Analyzer は、エラーを TVISION_HOME/logs/analyzer.log に記録します。ただし、Analyzer のサービス部分を初期化しているときにエラーが発生した場合、そのエラーは analyzer_startup.log ファイルに記録されます。

Analyzer とともに DB2 タイプ 2 または Oracle oci JDBC ドライバを使用する設定

TVisionSetupInfo ユーティリティは、DB2 ユニバーサル JDBC ドライバ（タイプ 4）または Oracle シンクライアント・ドライバをセットアップします。DB2 ユニバーサル・ドライバ（タイプ 2）や Oracle oci ドライバなど、ほかのドライバを使用するには、次の手順を実行します。

- 1 データベース・ホスト名やデータベース名などの必要なパラメータを入力するために、TVisionSetupInfo を少なくとも 1 回実行します。
- 2 DB2 ユニバーサル・ドライバ（タイプ 2）を使用するには、プロパティ・ファイル < TVISION_HOME > /config/datamgr/Database.properties にプロパティ jdbc_url を次の形式で追加します。

jdbc:db2: <データベース接続名>

<データベース接続名> は、Analyzer が使用するデータベースの別名です。データベース接続名を調べる方法については、60 ページ「データベース・アクセスの設定」を参照してください。

- 3 Oracle oci ドライバ（タイプ 2）を使用するには、プロパティ・ファイル < TVISION_HOME > /config/datamgr/ Database.properties にプロパティ jdbc_url を次の形式で追加します。

jdbc:oracle:oci: <ユーザ> / <パスワード> @ <データベース接続名>

<データベース接続名> は、Analyzer が使用する Oracle ネット・サービス名です。<ユーザ> と <パスワード> は、Oracle 接続に必要なユーザ名とパスワードです。データベース接続名を調べる方法については、60 ページ「データベース・アクセスの設定」を参照してください。

- 4 いったん jdbc_url プロパティを変更すると、Analyzer もアプリケーション・サーバもそのプロパティ値を使ってデータベースに接続するようになります。WebSphere や WebLogic などのアプリケーション・サーバは、32 ビット JVM を使用するため、ライブラリ・パス環境変数に変更を加える必要はありません。

マルチスレッド・サーブレット / JMS イベント

標準設定では、サーブレットが JMS 呼び出しを行うためにスレッドを生成する場合、サーブレットは追跡情報を子スレッドに渡します。その結果、サーブレット・イベントと JMS イベントは両方とも同じビジネス・トランザクションに属することになります。ただし、これらのイベントを別々のトランザクションに分ける必要がある場合もあります。たとえば、サーブレットと同じトランザクションの一部にたくない長時間実行されているスレッドを、サーブレットが生成することもできます。

標準設定の動作を変更して、TransactionVision が子スレッドをそれを生成するサーブレットと同じビジネス・トランザクションの一部として認識しないようにするには、次の手順を実行します。

- 1 < TVISION_HOME > /config/services/Analyzer.properties ファイルを開きます。
- 2 次の例のように、スレッドを生成するプログラムの名前を `separate_child_thread_txns` プロパティに追加します。

```
separate_child_thread_txns=program1, program2
```

複数のプログラム名を入力する場合は、カンマで区切ります。
- 3 Analyzer.properties を閉じて保存します。
- 4 Analyzer を再起動します。

不要な Analyzer ビーンの無効化

TVisionSetupInfo の実行時に Analyzer を JMS イベント収集用に設定すると、JMSPubSubRelationBean が標準設定で有効にされます。このビーンは、イベントの分析で JMS パブリッシュおよびサブスクライブ・イベントを関連させるために必要です。

次の両方に当てはまる場合には、このビーンを無効にすることで、パブリッシュ / サブスクライブ・メカニズムの処理に関連したオーバーヘッドをなくせます。

- ▶ PubSub JMS イベントを Sensor から収集しない
- ▶ Analyzer の最大限のパフォーマンスを必要とする

JMSPubSubRelationBean を無効にするには、

< **TVISION_HOME** > /config/Beans.xml を編集します。

```
<!-- this bean is used to correlate Publish/Subscribe events -->
<Module type="Bean"
class="com.bristol.tvision.services.analysis.eventanalysis.JMSPubSubRel
ationshipBean"/>
```

を、次のように変更します。

```
<!-- this bean is used to correlate Publish/Subscribe events -->
<!--Module type="Bean"
class="com.bristol.tvision.services.analysis.eventanalysis.JMSPubSubRel
ationshipBean"/-->
```

ローカル・トランザクション・マッチング

標準設定では、Analyzer は WebSphere MQ イベントをローカル・トランザクションにグループ分けするために、厳密なローカル・トランザクション・マッチングを使用します。ただし、厳密なローカル・トランザクション・マッチングを無効にして、代わりに標準設定の MQ ローカル・トランザクション・ビーンを使用する必要が生じる場合があります。このビーンは、作業単位に従ってイベントをローカル・トランザクションにグループ分けします。

特定の WebSphere MQ イベントについて厳密なローカル・トランザクション・マッチングを無効にするには、TransactionVision の MQStrictLocalTxnExclude.xml ファイルで条件を定義する必要があります。イベントがこのファイルで指定された条件のどれかに一致すると、Analyzer はそれを独立したローカル・トランザクションに入れずに、標準設定の MQ ローカル・トランザクション・ビーンを代わりに使用します。

条件は、プログラム名、キュー・マネージャ、およびオブジェクト名の組み合わせで構成されます。次の例は、次の条件のどちらかに一致するイベントについて、厳密なローカル・トランザクション・マッチングを無効にします。

- ▶ プログラム名が amqcrsta、キュー・マネージャが QM1、オブジェクトが TEST.Q1

- ▶ プログラム名が `amqcrsta`、キュー・マネージャが `ALT_QM`、オブジェクトが `TEST.Q`

```
<Criteria>
  <Match id="0">
    <value xpath="/Event/StdHeader/ProgramName">amqcrsta</value>
    <value xpath="/Event/Technology/MQSeries/MQObject/@queueManager">
QM1</value>
    <value xpath="/Event/Technology/MQSeries/MQObject/@objectName">
TEST.Q1</value>
  </Match>

  <Match id="1">
    <value xpath="/Event/StdHeader/ProgramName">amqcrsta</value>
    <value xpath="/Event/Technology/MQSeries/MQObject/@queueManager">
ALT_QM</value>
    <value xpath="/Event/Technology/MQSeries/MQObject/@objectName">
TEST.Q1</value>
  </Match>
</Criteria>
```

Analyzer の失敗モード

Analyzer は、2つのモード、標準モードと失敗モードで動作できます。失敗モードでは、失敗したビジネス・トランザクション（または SLA に違反したトランザクション）のイベント・データだけを保存します。成功したビジネス・トランザクションについては、対応するビジネス・トランザクション行だけが保存されます。失敗モードを使用すると、プロジェクトのデータ・ストレージに関する要件が緩和されます。失敗モードを有効にする方法については、『**Using TransactionVision**』（英語版）を参照してください。

JDBC Sensor のデータベース解決

JDBC イベントはすべて、現在のデータベース接続 URL を報告します。

DB2 の場合、URL の構文は次のいずれかになります。

- ▶ jdbc:db2: < db2 データベース別名 > (タイプ 2 のドライバの場合)
- ▶ jdbc:db2:// < ホスト > : < ポート > / < db2 データベース別名 > (タイプ 4 のドライバの場合)

Oracle の場合、URL の構文は次のいずれかになります。

- ▶ jdbc:oracle:thin:@// < ホスト > : < ポート > / < サービス名 > (タイプ 4 のドライバの場合)
- ▶ jdbc:oracle:thin: < ホスト > : < ポート > : < SID > (タイプ 4 のドライバの場合)
- ▶ jdbc:oracle:oci:@ < TNSName > (タイプ 2 のドライバの場合)
- ▶ jdbc:oracle:oci:@ (ローカルの標準設定の接続の場合)

JDBC Sensor は、対応するイベントがどのデータベースに関連付けられているかを識別するために、接続 URL を使用します。多くの場合、URL にはデータベースを識別するために必要な情報がすべて含まれています。ホスト、ポート、およびデータベース名が含まれるタイプ 4 スタイルの URL の場合は特にそうです。その場合、このファイルに変更を加える必要はありません。URL がこの情報を含んでいない可能性もあります。その場合、TransactionVision Analyzer はデータベース名を識別するために、**JDBCSystemModelDefinition.xml** で定義されたルールに従います。このファイルを設定する必要があるかどうかは、アプリケーションがどのようにしてデータベースに接続するかと、使用されるデータベース別名がデータベースの実際の名前に解決されることを望むかどうかによって決まります。

TVISION_HOME/config/services/JDBCSystemModelDefinition.xml

このファイルには、JDBC 接続 URL をデータベース・レベルのオブジェクトに関連付けるためのデータが含まれています。このファイルには次の4つのセクションがあります。

- ▶ 各ベンダ用の標準設定のデータベース・インスタンス名（オプション）
- ▶ 監視対象の環境内にあるデータベース・インスタンス・オブジェクトのリスト。これらのオブジェクトは、標準ローカル・データベースのセクションとマッピングのセクションで参照できます。
- ▶ 標準のローカル・データベース接続のリスト。これは Oracle にのみ適用されます。
- ▶ JDBC URL 属性と、対応するデータベースとの間のマッピング。

標準設定のデータベース・インスタンス

このオプションのセクションでは、DB ベンダおよびプラットフォームに対して使用する標準のデータベース・インスタンス文字列を定義します。

- ▶ `<defdb2instwin>db2</defdb2instwin>` これは、Microsoft Windows プラットフォーム上の DB2 について標準のデータベース・インスタンス名を定義します。
- ▶ `<defdb2inst>db2inst1</defdb2inst>` これは、ほかのすべてのプラットフォーム上の DB2 について標準の名前を定義します。
- ▶ `<deforacleinst>1521</deforacleinst>` これは、Oracle インスタンスの標準のポート番号を定義します。

これらのエントリが存在しない場合、TransactionVision は上記の標準設定値を想定します。

データベース・インスタンス・オブジェクト

2 番目のセクションは、環境内の種類が異なるデータベース・インスタンスを定義します。

```
<dbinstance vendor="oracle" host=" myhost " name="1544">
  <protocol name="tcp">
    <param name="host" value=" myhost "/>
    <param name="port" value="1544"/>
  </protocol>
</dbinstance>
```

各データベース・インスタンス要素は、vendor (db2 または oracle), host, instance name の各属性を持っていなければなりません。インスタンスは、1 つ以上の protocol セクションを持つことができます。

DB2 の場合、インスタンス名は DB2 インスタンスに設定されます。Oracle の場合、インスタンス名はリスナ・プロセスのポート番号に設定されます。

標準設定のローカル・データベース接続

3 番目のセクションは、所定のホストとデータベース・インスタンスについて、標準のデータベースを定義します。標準ローカル・データベース定義の中で参照されるインスタンスには、データベース・インスタンス定義（上記を参照）が存在していなければなりません。

```
<localdefdb vendor="oracle" host="myhost" inst="1544"
  name="mydb"/>
```

標準のローカル・データベース設定の使用例のケースは、JDBC プログラムが jdbc:oracle:oci:@ のような URL を使って接続を確立した場合です。このケースでは、TransactionVision は使用されているデータベース名を識別できません。したがって、TransactionVision は **JDBCSystemModelDefinition.xml** ファイルの中を見て、当該イベントが発生したホストについて標準のデータベースとして何が定義されているかを調べます。これが上の例の myhost 上で動作していたとすると、このイベントは mydb という名前の Oracle データベースと関連付けられます。

JDBC URL のマッピング

4 番目のセクションは、JDBC URL 内のデータベース別名と実際のデータベース名とのマッピングを定義します。

マッピング要素は次のようになっています。

```
<mapping vendor="vendor">
  <src host="host name" dbalias="db alias"/>
  <dst host="host name" inst="instance name" dbname="db name"/>
</mapping>
```

- ▶ vendor 属性の値は、"db2" または "oracle" に設定できます。
- ▶ <src> 要素内の dbalias 属性は、Oracle では TNS 名に、DB2 ではデータベース別名に設定されます。
- ▶ <src> 要素の host は、監視対象のアプリケーションが実行されているホストです。
- ▶ インスタンス属性は、DB2 マッピング・エントリでのみ必要です。
- ▶ <dst> 要素には、一致するものがマッピングされる定義が含まれており、その host 属性と inst 属性は上記で定義された <dbinstance> エントリと一致しなければなりません。

たとえば、次のようなマッピング・エントリがあるとします。

```
<mapping vendor="oracle">
  <src host="apphost" dbalias="DBNAME"/>
  <dst host="myhost" inst="1544" dbname="mydb"/>
</mapping>
```

この状況で、apphost 上で動作している監視対象のアプリケーションが、jdbc:oracle:oci:@DBNAME のような JDBC URL 接続を使用した場合、Analyzer はこのデータベースを myhost 上で動作している Oracle の mydb データベースとして解決することができます。

データベース名の解決方法

JDBCSystemModelDefinition.xml 内でルールが定義された後、Analyzer がイベントを処理するときに、報告すべきデータベース名を決定するには、一連のルールに従います。このデータベース名解決は次の順序で行われます。

DB2 タイプ 2 ドライバの場合のデータベース解決

- 1 データベース・マッピング・リストの中を検索し、ソースの属性がイベントのホストおよびデータベース別名と一致するエントリーがあるかどうかを調べます。
- 2 一致するエントリーが見つかったら、`mapping destination` セクションからの情報を使ってデータベース・オブジェクトを作成します。
- 3 そのようなエントリーが見つからなかった場合は、イベントのホスト、ローカル・データベース・インスタンス名（イベント内にこのデータがない場合は標準設定値）、およびデータベース別名を使って、データベース・オブジェクトを作成します。

DB2 タイプ 4 ドライバの場合のデータベース解決

- 1 DB2 タイプ 2 ドライバの場合の解決手順の手順 1 および 2 を実行します。
- 2 (1) が失敗した場合は、以下の手順を実行します。
- 3 URL を解析して、サーバ名とリッスン・ポート番号を取得します。
- 4 データベース・インスタンスのリストで、ホスト名とポート番号が一致するデータベース・インスタンスを見つけます。
- 5 (2) で一致するエントリーが見つかった場合は、見つかったデータベース・インスタンスにイベントが関連付けられます。データベース名は、イベント URL が報告した名前に設定されます。
- 6 (2) でそのようなエントリーが見つからず、イベント・ホストが URL ホストと同じである場合は、イベントからのローカル・データベース・インスタンス・データをデータベース・インスタンスとして使用します。あるいは、イベント内にそのデータが存在しない場合は、標準設定値を使用します。上記の手順で判明したホスト、データベース・インスタンス、および URL データベース名を使って、データベース・オブジェクトを作成します。
- 7 (2) でそのようなエントリーが見つからず、イベント・ホストと URL ホストが異なる場合は、URL ホスト、標準設定のデータベース・インスタンス名、および URL データベース名を使って、データベース・オブジェクトを作成します。

Oracle タイプ 2 ドライバの場合のデータベース解決

- 1 URL にデータベース名 (TNS 名) が含まれていない場合は、データベース・マッピング・リストをチェックし、ソースのホストおよびデータベース別名がイベントのホストおよびデータベース別名と一致するエントリがあるかどうかを調べます。
- 2 一致するエントリが見つかったら、`mapping host` セクションのデータ (ホスト、インスタンス、データベース名) を使って、そのイベントのためのデータベース・オブジェクトを作成します。
- 3 一致するエントリが見つからなかった場合は、(a) JDBC イベント・ホスト、(b) 標準設定の Oracle データベース・インスタンス (ポート)、および (c) イベント・データベース名を使って、データベース・オブジェクトを作成します。
- 4 URL にデータベース名が含まれていない場合は、`default local database` セクション内に一致するホスト名を持つエントリがあるかどうかをチェックします。一致するエントリが複数見つかった場合は、最初に一致したものを使用します。一致したエントリからのホスト、データベース・インスタンス、およびデータベース名のデータを使って、データベース・オブジェクトを作成します。
- 5 一致するエントリが見つからなかった場合は、(a) JDBC イベント・ホスト、(b) Oracle データベース・インスタンスの標準設定値、および (c) データベース名として文字列 "default" を使って、データベース・オブジェクトを作成します。

Oracle タイプ 4 ドライバの場合のデータベース解決

- 1 このケースでは、解決は必要ありません。URL が、データベース・ホスト、リスナ・ポート番号、および SID/ サービス名を提供します。
- 2 ホストとポート番号によって識別された Oracle データベース・インスタンスの詳細を設定データが提供しているかどうかをチェックします。そのようなレコードがあれば、その情報を使用します。
- 3 データベース・インスタンスおよび SID/ サービス名の情報を使って、システム・モデル・テーブル内にデータベース・オブジェクトを作成します。

.NET Agent に関する最適化

.NET Agent を含むデプロイメント環境で Analyzer を使用する場合は、TransactionVision に付属する SonicMQ メッセージング・ミドルウェアの代わりに WebSphere MQ をメッセージング・ミドルウェア・プロバイダとして使用することによって、イベントの転送時間と処理時間を短縮できます。

さらに、.NET アプリケーションに対するヒットが散発的な場合は、イベント・キューのサイズを大きくすることを考慮してください。イベント・キューのサイズを大きくする方法については、メッセージング・ミドルウェア・プロバイダのドキュメントを参照してください。

イベント・データベースのサイズの縮小

TransactionVision は、XML イベント圧縮ビーンを提供しています。各イベントについてデータベース・サイズを縮小するには、このビーンを使用します。

注：XML イベント圧縮ビーンを有効にすると、ユーザ・バッファ・データに対するクエリができなくなります。

TVisionSetupInfo を実行した後、< TVISION_HOME > /config/services/Beans.xml ファイルを開き、次の部分を見つけて変更します。

元のセグメント

```
<Module name="DBWriteEventCtx" type="Context">

  <!-- This context contains beans that write the XML event (or part of it) to the
  database. -->
  <!-- Each registered bean in the chain is called. -->

  <Module
  class="com.bristol.tvision.services.analysis.dbwrite.DBWriteEventDefaultBea
  n" type="Bean"/>

  <!-- Replace the default bean with this one if you want ZIP compression for
  the XML event -->
  <!--Module type="Bean"
  class="com.bristol.tvision.services.analysis.dbwrite.DBWriteEventCompress
  edBean" /-->
</Module>
```

変更後のセグメント

```
<Module name="DBWriteEventCtx" type="Context">

    <!-- This context contains beans that write the XML event (or part of it) to the
    database. -->
    <!-- Each registered bean in the chain is called. -->

<!--<Module
class="com.bristol.tvision.services.analysis.dbwrite.DBWriteEventDefaultBea
n" type="Bean"/-->

    <!-- Replace the default bean with this one if you want ZIP compression for
    the XML event -->
    Module type="Bean"
class="com.bristol.tvision.services.analysis.dbwrite.DBWriteEventCompress
edBean" />
</Module>
```

注：一度に有効にするビーンは1つだけにしてください。さもないと、DBWriteEventDefaultBean と DBWriteEventCompressedBean が両方とも有効になったときに、例外が発生します。

圧縮ビーンを使用するときには、イベント・データを BLOB 形式で保存するよう
に DatabaseDef.xml を更新する必要があります。EVENT テーブルと
EVENT_OVERFLOW テーブルの定義を更新して、event_data カラムのタイプを
CLOB から BLOB に変更してください。

この変更を行った後、その変更を有効にするために、Analyzer と UI/Job サーバを再起動して新しいプロジェクトを作成する必要があります。

次の例は、イベント・データを BLOB 形式で保存するように変更する前の DatabaseDef.xml を示しています。

```
<Table name="EVENT" volatile="true">
[...]
  <Column name="event_data" type="CLOB" size="1M"/>
[...]
</Table>
<Table name="EVENT_OVERFLOW" volatile="true">
[...]
  <Column name="event_data" type="CLOB" size="1M" notNull="true"/>
[...]
</Table>
```


第 8 章

Analyzer のログ記録の設定

本章の内容

- ▶ ログ・ファイル（101 ページ）
- ▶ 循環的なログ記録（102 ページ）
- ▶ Windows と UNIX のシステム・ログの使用（104 ページ）
- ▶ SMTP ログ記録の有効化（105 ページ）
- ▶ SNMP ログ記録の有効化（107 ページ）
- ▶ JMS ログ記録の有効化（108 ページ）

ログ・ファイル

標準設定では、すべての TransactionVision コンポーネントがエラー・メッセージと警告メッセージを適切なログ・ファイルに記録します。ログ・ファイルの場所は、TVisionSetupInfo または SensorSetup を実行したときに指定され、Setup.properties ファイルに保存されます。

Analyzer は、エラー・メッセージを analyzer.log ファイルに保存します。Windows 上では、Analyzer はほかに次の 3 つのログ・ファイルも使用します。

- ▶ **analyzer_startup.log** には、Analyzer の Windows サービス部分の実行に関する情報が収められます。通常、これには Analyzer の起動に使用されるオプションに関する情報が含まれます。Analyzer のサービス部分の初期化中にエラーが発生した場合、そのエラーはこのファイルに保存されます。

- ▶ **analyzer_stderr.log** と **analyzer_stdout.log** は、それぞれ Analyzer プロセスの標準出力と標準エラーのログ・ファイルです。コンソールまたは標準エラー・ファイルに出力するカスタム分析ビーンがある場合、その出力はこれらのファイルに記録されます。Analyzer の起動や実行に問題があり、標準の **analyzer.log** ファイルにそのエラーを示す情報が何も含まれていない場合は、これらのファイルも調べる必要があります。

循環的なログ記録

標準設定では、Analyzer は循環的なログ記録の形式を使用します。ログ・ファイルが設定された最大サイズに達すると、名前を変更されてバックアップ・ファイルとなり、新しい空のログ・ファイルが作成されます。標準設定では、最大ログ・サイズは 10 MB で、バックアップ・ログ・ファイルは 1 つです。

標準設定を使用した場合、たとえば Analyzer ログ・ファイル **analyzer.log** のサイズが 10 MB に達すると、その名前が **analyzer.log.1** に変更され、新しい **analyzer.log** ファイルが作成されます。設定を変更してバックアップ・ファイルの数を 2 個にすると、**analyzer.log** が 10 MB に達したときに以下のことが起こります。

- ▶ **analyzer.log.2** が存在する場合は、それが削除されます。
- ▶ **analyzer.log.1** の名前が **analyzer.log.2** に変更されます。
- ▶ **analyzer.log** の名前が **analyzer.log.1** に変更されます。
- ▶ 新しい **analyzer.log** が作成されます。

循環的なログ記録を望まない場合は、直線的なログ記録を使用するように設定を変更できます。その場合は、単一のログ・ファイルが生成されます。

< TVISION_HOME > /config/logging/*.Logging.xml ファイルは、使用されるログ記録の種類、最大ログ・ファイル・サイズ、およびバックアップ・ログ・ファイルの数を、各コンポーネントごとに指定します。たとえば、Sensor.Logging.xml は、Servlet Sensor と JMS Sensor に関する設定を指定します。このファイルには、次の例のようなエントリが含まれます。

```
<appender class="tvision.org.apache.log4j.RollingFileAppender"
name="SENSOR_LOGFILE">
  <param name="File" value="c:/Program Files/Hewlett-
Packard/TransactionVision/logs/sensor.log"/>
  <param name="Append" value="true"/>
  <param name="MaxBackupIndex" value="2"/>
  <param name="MaxFileSize" value="10MB"/>
  <layout class="tvision.org.apache.log4j.PatternLayout">
  <param name="ConversionPattern" value="%d [%t] %-5p %c %x - %m%n"/>
  </layout>
</appender>
```

ログ・ファイルの最大サイズ

ログ・ファイルの最大サイズを変更するには、MaxFileSize パラメータの値を希望するサイズに変更します。指定する値の最後には、メガバイトとキロバイトを区別するために、MB か KB を付けます。

バックアップ・ログ・ファイルの最大数

バックアップ・ファイルの最大数を変更するには、MaxBackupIndex パラメータの値を希望するバックアップ・ファイル数に変更します。

循環的なログ記録から直線的なログ記録への変更

循環的なログ記録ではなく直線的なログ記録を使用するには、次の手順を実行します。

- 1 appender class の値を RollingFileAppender から FileAppender に変更します。たとえば、上記の例では、最初の行を次のように変更します。

```
<appender class="tvision.org.apache.log4j.FileAppender"
name="SENSOR_LOGFILE">
```

- 2 MaxBackupIndex パラメータと MaxFileSize パラメータのエントリを削除します。

Windows と UNIX のシステム・ログの使用

UNIX プラットフォームと Windows プラットフォームでは、出力をシステム・イベント・ログ（Windows の場合はイベント・ログ，UNIX の場合は syslog）に記録するように、TransactionVision を設定することができます。そのために必要なログ設定ファイルの例は、`TVISION_HOME/config/logging/system/*/Sensor.Logging.xml` の中にあります。

Windows でも UNIX でも、専用のイベント・アペンダを定義する必要があります。

Windows イベント・アペンダ

次の例は、イベント・ログを使用するための Windows イベント・アペンダを設定する方法を示しています。

```
<appender name="NT_EVENT_LOG" class="tvision.org.apache.log4j.nt
.NTEventLogAppender">
  <layout class="tvision.org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%d [%t] - %m%n"/>
  </layout>
</appender>
```

その後、`NT_EVENT_LOG` を選択したカテゴリ定義の中で参照できます。次に例を示します。

```
<category additivity="false" class="com.bristol.tvision.util.log.XCategory"
name="sensorLog">
  <priority class="com.bristol.tvision.util.log.XPriority" value="info"/>
  <appender-ref ref="NT_EVENT_LOG"/>
</category>
```

Windows ではさらに、特別な DLL をパスに追加する必要があります。この DLL は `NTEventLogAppender.dll` で、`config¥logging¥system¥bin` ディレクトリの中にあります。次に例を示します。

```
set path=%TVISION_HOME%4config¥logging¥system¥bin;%PATH%
```


UNIX イベント・アペンダ

次の例は、syslog を使用するための UNIX イベント・アペンダを示しています。

```
<appender name="SYSLOG" class="tvision.org.apache.log4j.net.SyslogAppender">
  <param name="SyslogHost" value="localhost"/>
  <param name="Facility" value="local0"/>
  <layout class="tvision.org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="[%t] %-5p %c %x
  - %m%n"/>
  </layout>
</appender>
```

SyslogHost パラメータと Facility パラメータを環境に合わせて適切に指定します。

SMTP ログ記録の有効化

SMTPAppender は、指定されたしきい値のエラー・ログ・イベントがアペンダに到達すると、SMTP サーバに電子メールを送信します。SMTPAppender を有効にするには、以下の内容を Analyzer.logging.xml ファイルに追加します。

```
<appender name="EMAIL"
class="tvision.org.apache.log4j.net.SMTPAuthenticateAppender">
  <param name="SMTPHost" value="smtp.myserver.net"/>
  <param name="To" value="analyzer_log4j@myserver.net"/>
  <param name="From" value="administrator@myserver.net"/>
  <param name="UserName" value="smtp_user"/>
  <param name="Password" value=""/>
  <param name="Authenticate" value="true"/>
  <param name="BufferSize" value="1"/>
  <param name="Threshold" value="info"/>
  <layout class="tvision.org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%d [%t] %-5p %c %x - %m%n"/>
  </layout>
</appender>
```

Threshold パラメータは、SMTPAppender にアペンドできるログ・レベルを指定します。

カスタマイズしたトリガ・イベント・エバリュエータを定義するには、EvaluatorClass パラメータを追加します。

```
<param name="EvaluatorClass" class="tvision.org.apache.  
log4j.spi.TriggeringEventEvaluator"/>
```

このインタフェースは、いつ電子メールを送信するべきかを決定するために次の関数を提供します。

```
public boolean isTriggeringEvent(LoggingEvent event) {  
    long l = 0;  
    synchronized(lock) {  
        l = (msgCount ++);  
    }  
    return (((l + 1)%msgPkgSize) == 0); // fire email on every msgPkgSize events.  
}
```

SNMP ログ記録の有効化

JoeSNMPTrapSender は、指定されたエラー・レベルが発生すると電子メールを送信します。JoeSNMPTrapSender を有効にするには、以下の定義を Analyzer.logging.xml ファイルに追加します。

```
<!-- SNMP TRAP appender !-->
<appender name="TRAP_LOG"
class="tvision.org.apache.log4j.ext.SNMPTrapAppender">
  <param name="ImplementationClassName"
value="tvision.org.apache.log4j.ext.JoeSNMPTrapSender"/>
  <param name="ManagementHost" value="127.0.0.1"/>
  <param name="ManagementHostTrapListenPort" value="162"/>
  <param name="EnterpriseOID" value="1.3.6.1.4.1.24.0"/>
  <param name="LocalIPAddress" value="127.0.0.1"/>
  <param name="LocalTrapSendPort" value="161"/>
  <param name="GenericTrapType" value="6"/>
  <param name="SpecificTrapType" value="12345678"/>
  <param name="CommunityString" value="public"/>
  <param name="ForwardStackTraceWithTrap" value="true"/>
  <param name="Threshold" value="DEBUG"/>
  <param name="ApplicationTrapOID" value="1.3.6.1.4.1.24.12.10.22.64"/>
  <layout class="tvision.org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%d,%p,[%t],[%c],%m%n"/>
  </layout>
</appender>
```

注 : JoeSNMPTrapSender が必要とする joesnmp.jar を CLASSPATH に追加する必要があります。この jar ファイルは、<http://sourceforge.net/projects/joesnmp> にある JoeSNMP プロジェクトからダウンロードできます。

JMS ログ記録の有効化

TransactionVision は、JMS メッセージング・プロバイダを通じてログ・メッセージを送信するためのメカニズムを提供しています。これは、**Analyzer.logging.xml** ファイルで設定される `com.bristol.tvision.appender.JMSAppender` アペンダを通じて行われます。

JMS アペンダを設定するには、次の 2 つの方法があります。通常は JNDI 設定を使って JMS 接続を設定しますが、直接 WMQ JMS 設定を使用することもできます。

次のどちらを使って設定するかを選択します。

▶ JNDI

または

▶ 直接 WMQ JMS

注：両方の方式が指定された場合は WMQ JMS 設定が優先され、JNDI 設定は無視されます。

BPI JMS 接続を手動で設定する場合や、JMS キューを通じてログを発行するために独立したログ記録機能を設定する場合は、以下の `JMSAppender` オプションを使用します。

- ▶ **ConnectionRetryDelay** は、接続が失敗したときに再試行するまでの時間です。
- ▶ **ConnectionRetryTimeout** は、応答のない接続がタイム・アウトになるまでの時間です。
- ▶ **QueueName** は、JNDI オブジェクトの名前（JNDI が使用されている場合）、またはキューの実際の名前（WMQ JMS の場合）です。
- ▶ **Username** と **Password** は、JMS プロバイダに対する認証が必要な場合のオプションの設定です。

JNDI 設定の場合

- ▶ **InitialContextFactoryName** は、JNDI コンテキスト・ファクトリのクラス名です。この値は、どの JMS ベンダを使用するかによって決まります（詳細については対応するドキュメントを参照してください）。たとえば、`com.sun.jndi.fscontext.RefFSContextFactory` や `com.tibco.tibjms.naming.TibjmsInitialContextFactory` などの値があります。
- ▶ **ProviderURL** は、JNDI リポジトリに接続するための URL であり、どの JMS ベンダを使用するかによって決まります。`RefFSContextFactory` の URL は、`file:/C:/jndi` に似たものになります。TIBCO の場合は、`tibjmsnaming://host:7222` のような形式の URL を使用することになります。
- ▶ **QueueConnectionFactoryName** は、キュー接続ファクトリ JNDI オブジェクトの名前です。

WMQ JMS 設定

WMQ JMS 固有の設定は、WMQ JMS に接続するために使用するキュー・マネージャ名、ホスト、ポート、およびチャンネルに対応しています。

TargetMQClient は、MQ が JMS メッセージの中で RFH2 ヘッダを使用するかどうかをオン / オフします。

```
<appender class="com.bristol.tvision.appender.JMSAppender"
name="JMS_APPENDER">
  <!--connection retry interval in milliseconds -->
  <param name="ConnectionRetryDelay" value="0"/>
  <param name="ConnectionRetryTimeout" value="0"/>
  <param name="QueueName" value=""/>
  <param name="UserName" value=""/>
  <param name="Password" value=""/>
  <!-- enable the following to provide JNDI context parameters for
    JMS connection -->
  <!--<param name="InitialContextFactoryName" value="" />
  <param name="ProviderURL" value="" />
    <param name="QueueConnectionFactoryName" value=""/>
  -->

  <!-- enable the following to provide WebSphere MQ parameters for
    JMS connection -->

  <!--
  <param name="MQQueueManagerName" value="" />
  <param name="MQClientConnectionHost" value="" />
  <param name="MQClientConnectionPort" value="" />
  <param name="MQClientConnectionChannel" value="" />
  <param name="TargetMQClient" value="false"/>
  -->

</appender>
```

第 III 部

UI/Job サーバのインストールおよび設定

第 9 章

TransactionVision UI/Job サーバをインストールする準備

本章の内容

- ▶ TransactionVision UI/Job サーバについて (113 ページ)
- ▶ デプロイメント環境における TransactionVision UI/Job Server (113 ページ)

TransactionVision UI/Job サーバについて

TransactionVision UI/Job サーバは、Apache Tomcat Servlet/JSP コンテナにバンドルされている Web アプリケーションです。TransactionVision UI/Job サーバは、TransactionVision Analyzer と Business Availability Center ゲートウェイ・サーバとの間の通信をサポートします。

デプロイメント環境における TransactionVision UI/Job Server

ほとんどのデプロイメント環境では、TransactionVision UI/Job サーバを Business Availability Center ゲートウェイ・サーバと同じホストにインストールしなければなりません。

TransactionVision UI/Job サーバのシステム要件については、33 ページ「サポートされている TransactionVision UI/Job サーバ・プラットフォーム」を参照してください。

第 10 章

UNIX プラットフォームへの UI/Job サーバのインストール

本章の内容

- ▶ インストール・ファイル (115 ページ)
- ▶ UNIX 上での UI/Job サーバ・インストール・プログラムの起動 (115 ページ)
- ▶ 初めてのインストール (117 ページ)
- ▶ アップグレード・インストール (117 ページ)
- ▶ UI/Job サーバのアンインストール (119 ページ)

インストール・ファイル

次の表は、TransactionVision UI/Job サーバ・パッケージのインストール・ファイルの名前を、サポートされるプラットフォーム別に示しています。

プラットフォーム	ファイル
Solaris	tvuijs_800_sol_sparc.tar.gz

UNIX 上での UI/Job サーバ・インストール・プログラムの起動

Solaris と Linux では、UI/Job サーバは **/opt/HP/TransactionVision** ディレクトリにインストールされます。AIX では、UI/Job サーバは **/usr/lpp/HP/TransactionVision** にインストールされます。

TransactionVision UI/Job サーバを UNIX プラットフォームにインストールするには、次の手順を実行します。

- 1 TransactionVision のインストール・ファイルがあるディレクトリの場所（DVD デバイスまたはダウンロード・ディレクトリ）に移動します。

注：Solaris と HP-UX では、代わりに DVD デバイスからホストのハード・ドライブ上の一時ファイルにインストール・ファイルをコピーしなければなりません。

- 2 使用するプラットフォームに対応するパッケージを展開します。115 ページ「インストール・ファイル」を参照してください。次に例を示します。

```
gunzip tvuijs_800_sol_sparc.tar.gz t
```

- 3 スーパーユーザとしてログインします。

```
su
```

- 4 次のコマンドを入力して、インストール手順を開始します。

```
./tvinstall_800_unix.sh
```

パッケージが展開されると、使用可能なコンポーネントを示したメニューが表示されます。次に例を示します。

```
The following TransactionVision packages are available for installation:
```

1. TransactionVision Analyzer
2. TransactionVision UI/Job Server
3. TransactionVision WebSphere MQ Agent
4. TransactionVision User Event Agent

```
99. All of above
```

```
q. Quit install
```

```
Please specify your choices (separated by,) by number/letter:
```

インストール先のコンピュータに TransactionVision コンポーネントを初めてインストールする場合は、「初めてのインストール」に進んでください。そのコンピュータに TransactionVision の以前のバージョンがインストールされている場合は、117 ページ「アップグレード・インストール」に進んでください。

初めてのインストール

- 1 **2**を入力して **Return** キーを押します。

インストール・スクリプトがパッケージをインストールして、次のメッセージを表示します。

```
Installation of <TVUIJOB> was successful.
```

TransactionVision コンポーネントのメニューが表示されます。

- 2 **q** と入力して **Return** キーを押し、インストール・プロセスを終了します。

このホストに追加のコンポーネントをインストールする場合は、それらのコンポーネントのインストール手順を参照してください。

アップグレード・インストール

- 1 **1**を入力して **Return** キーを押します。

インストール・スクリプトは、UI/Job サーバの以前のバージョンがインストールされていると判断すると、次のメッセージを表示します。

```
There is an earlier version of TransactionVision installed on the system.  
The earlier version has to be uninstalled before installing the current package(s).  
Continue with the uninstallation? (Y/N) [N]:
```

- 2 以前のバージョンをアンインストールするために、**Y**を入力して **Return** キーを押します。次のプロンプトが表示されます。

```
Before uninstalling the previous version, TransactionVision provides the option of  
migrating configuration to the new installation:  
Installation has detected a previous installation of TransactionVision.  
Do you want to migrate existing TransactionVision configuration files? (y/n) [y] :
```

- 3 設定を新しいバージョンに移行するために、**Y**を入力して **Return** キーを押します。既存の UI/Job サーバ設定ファイルのバックアップ・コピーが移行のために自動的に作成されます。その後、UI/Job サーバの以前のバージョンがアンインストールされ、TransactionVision コンポーネントのメニューが表示されます。手順 4 に進みます。

N を入力して **Return** キーを押すと、将来のために設定ファイルのバックアップ・コピーを作成するためのオプションが提供されます。

```
Although migration will not be performed at this time, you may optionally back up
configuration files from your previous installation for reference purposes. Note that
answering N will overwrite these files, causing any existing setup information to be lost.
Do you wish to back up configuration files from the previous installation? (y/n) [y] :
```

設定ファイルのバックアップ・コピーを作成する場合は **Y** を入力して **Return** キーを押します。すると、バックアップの場所を指定するように求められます。

```
Enter the directory to which existing TransactionVision configuration files should be
backed up [/opt/TVision/migrate_tv780_date_time]:
```

標準設定の場所を使用する場合は **Return** キーを押し、それ以外の場所を使用する場合は希望するバックアップ・ディレクトリの場所を入力します。インストール・ユーティリティは次の作業を実行します。

- ▶ 現在の設定ファイルを指定されたディレクトリにコピーします。
- ▶ TransactionVision の旧バージョンをアンインストールします。
- ▶ TransactionVision コンポーネントのメニューを再び表示します。

4 **q** と入力して **Return** キーを押し、インストール手順を終了します。

UI/Job サーバのアンインストール

どの TransactionVision コンポーネントをアンインストールする場合も、次の手順を実行します。

- 1 スーパーユーザとしてログインします。

`su`

- 2 次のコマンドを入力します。

```
./tvinstall_800_unix.sh -u
```

次のメニューが表示されます（実際のオプションはインストールされている TransactionVision コンポーネントによって異なります）。

The following TransactionVision packages are available for installation:

1. TransactionVision UI/Job Server
3. TransactionVision WebSphere MQ Agent
4. TransactionVision User Event Agent

99. All of above

q. Quit install

Please specify your choices (separated by,) by number/letter:

注： オプションと番号は、当該コンピュータ上で使用できるインストール・ファイルによって決まります。

- 3 **1** を入力して **Return** キーを押します。

すべての TransactionVision コンポーネントをアンインストールするには、**99** と入力して **Return** キーを押します。

指定したパッケージがアンインストールされ、再びメニューが表示されます。

- 4 **q** と入力して **Return** キーを押し、インストールを終了します。

第 11 章

Windows への UI/Job サーバのインストール

本章の内容

- ▶ Windows 上での UI/Job サーバ・インストール・プログラムの起動 (121 ページ)
- ▶ 初めてのインストール (122 ページ)
- ▶ UI/Job サーバのアンインストール (123 ページ)

Windows 上での UI/Job サーバ・インストール・プログラムの起動

UI/Job サーバをインストールするには、次の手順を実行します。

- 1 インストール先のシステムに、Administrator として、または Administrator 権限を持つユーザとしてログインします。
- 2 UI/Job サーバを以前のリリースからアップグレードする場合は、JAVA_HOME 環境変数が設定されていることを確認するか、パスに Java が含まれていることを確認してください。
- 3 当該コンピュータで現在実行されている Windows プログラムを、自動バックアップ・プログラムも含めてすべて終了します。ウイルス対策、スパイウェア防止、および脅威対策プログラムを終了する必要はありません。
- 4 Windows エクスプローラから **tvuijs_800_win.exe** をダブル・クリックします。[InstallShield Welcome] 画面が開きます。
- 5 [Next] をクリックし、[Welcome] 画面が表示されるのを待ちます。
- 6 InstallShield の [Save Files] 画面が表示された場合は、[Next] をクリックして標準設定のフォルダ（たとえば C:\TEMP\Hewlett-Packard\TransactionVision）にインストール・ファイルを抽出するか、[Change] をクリックして希望するフォルダを選択して [Next] をクリックします。

このホストに TransactionVision UI/Job サーバを初めてインストールする場合は、122 ページ「初めてのインストール」に進んでください。インストール先のコンピュータに Analyzer の以前のバージョンがインストールされている場合は、123 ページ「UI/Job サーバのアンインストール」に進んでください。

初めてのインストール

初めてのインストールの場合は、[Welcome] 画面が表示されます。

- 1 [Welcome] 画面で [Next] をクリックすると、TransactionVision のライセンス契約が表示されます。
- 2 [Yes] をクリックして、ライセンス契約に同意します。[User Information] 画面が表示されます。
- 3 自分の名前と会社名を入力し、[Next] をクリックします。[Choose Destination Location] 画面が開きます。
- 4 標準設定のインストール・フォルダ (C:\Program Files\Hewlett-Packard\TransactionVision) を使用する場合は、[Next] をクリックします。別のインストール・フォルダを選択する場合は、[Browse...] をクリックして希望するインストール・フォルダを選択し、[Next] をクリックします。
選択したパッケージが指定した場所にインストールされます。[Setup Complete] ページが表示されます。
- 5 [Finish] をクリックすると、インストールが完了します。

UI/Job サーバのアンインストール

Analyzer をアンインストールするには、次の手順を実行します。

- 1 [スタート] メニューから [コントロール パネル] を選択します。
- 2 [アプリケーションの追加と削除] をダブル・クリックします。
- 3 アンインストールする HP TransactionVision パッケージを選択し、[変更と削除] をクリックします。保守メニュー画面が表示されます。
- 4 [Remove] を選択して [Next] をクリックし、TransactionVision コンポーネントを削除します。
- 5 指定したパッケージをアンインストールすることを確認し、[OK] をクリックします。指定したパッケージがアンインストールされます。次の種類のファイルは削除されません。
 - ▶ インストール後に追加されたファイル
 - ▶ まだインストールされているパッケージに関連付けられた共有ファイル
共有ファイルがインストールされたパッケージと関連付けられていると判断された場合（たとえば、ほかの TransactionVision パッケージがすべてアンインストールされた場合）は、[Locked File Detected] 画面が表示されます。
 - ▶ インストールされた共有ファイルをすべて残す場合は、[Don't display this message again] にチェック・マークを付けて [No] をクリックします。
 - ▶ 現在のファイルは残すけれど、ほかの共有ファイルについてはこのメッセージが表示されるようにする場合は、[No] をクリックします。
 - ▶ 共有ファイルを削除する場合は、[Yes] をクリックします。
- 6 [Uninstallation Complete] 画面が表示されます。[Finish] をクリックしてアンインストール手順を完了します。

注： TransactionVision の Web ユーザ・インタフェースをアンインストールしたら、その一時キャッシュと配布ディレクトリをクリーンアップする必要があります。WebSphere がこれを自動的に行うことはなく、古いファイルが残っていると新しくインストールしたバージョンが正しく動作しない可能性があります。

第 12 章

UI/Job サーバの設定

本章の内容

- ▶ UI/Job サーバの設定について (125 ページ)
- ▶ TVisionSetupInfo による変更を受けるファイル (126 ページ)
- ▶ TVisionSetupInfo が必要とする情報 (126 ページ)
- ▶ TVisionSetupInfo の実行 (127 ページ)
- ▶ UI/Job サーバの管理 (132 ページ)

UI/Job サーバの設定について

TransactionVision は、UI/Job サーバを設定するプロセスをガイドするために、**TVisionSetupInfo** ユーティリティを提供しています。このユーティリティは、セットアップ・プロセスに必要な情報を入力するように求め、必要な環境変数を設定します。

重要 : UI/Job サーバを設定する前に、Analyzer を設定する必要があります。詳細については、第 7 章「Analyzer の設定」を参照してください。

このユーティリティを実行した後、本章で説明しているように任意でいくつかの設定を実行できます。

すべての設定が終わったら、本章の説明に従って UI/Job サーバを起動する必要があります。

TVisionSetupInfo による変更を受けるファイル

TVisionSetupInfo ユーティリティは、UI/Job サーバを設定するときに以下のファイルに変更を加えます。

- ▶ TVISION_HOME/config/datamgr/Database.properties
- ▶ TVISION_HOME/config/setup/Setup.properties

さらに、**TVisionSetupInfo** は以下の操作を実行します。

- ▶ ソフトウェア・ツールのインストール・パスを **TVISION_HOME/config/setup/DefaultInstallPath.xml** に保存します。
- ▶ **TVISION_HOME/bin/SetupEnv.[sh|bat]** を生成します。このファイルは、JAVA_HOME, CLASSPATH, および TransactionVision が必要とするシステム・ライブラリ・パス環境変数を設定するために、TVisionSetup によって使用されます。

TVisionSetupInfo が必要とする情報

TVisionSetupInfo ユーティリティは、設定に必要な以下の情報を入力するように求めます。**TVisionSetupInfo** を実行する前に必要な情報を確認して、適切な情報を入力できるようにしてください。

ログ・ファイルの場所

TransactionVision が UI/Job サーバのログ・ファイルを保存することになるディレクトリのパス名を入力する必要があります。標準設定の値は TVISION_HOME/logs です。詳細については、第 8 章「Analyzer のログ記録の設定」を参照してください。

データベース情報

TransactionVision で使用するデータベースの種類（IBM DB2, Oracle, または SQL Server）と、データベース固有のプロパティを指定する必要があります。

収集される情報は、Analyzer をセットアップするときに収集される情報と同じです。詳細については、73 ページ「データベース情報」を参照してください。

Business Availability Center への登録

TransactionVision コンポーネントが使用する Business Availability Center ゲートウェイ・サーバの場所を指定する必要があります。

TVisionSetupInfo の実行

- 1 インストール先のシステムに、root, Administrator, または Administrator 権限を持つユーザとしてログインします。
- 2 TVisionSetupInfo.[sh|bat] と入力して TVisionSetupInfo スクリプトを実行します。

オペレーティング・システム	スクリプト・コマンド
AIX, Linux, Solaris	< TVISION_HOME > /bin/TVisionSetupInfo.sh
Windows	< TVISION_HOME > %bin%TVisionSetupInfo.bat

注： TVISION_HOME 環境変数はインストール時に、インストール・ディレクトリの絶対パスに設定されます。たとえば、TVISION_HOME は Solaris では /opt/HP/TransactionVision, AIX では /usr/lpp/HP/TransactionVision のようになります。

TVisionSetupInfo は、以下を設定するために必要な情報を入力するように求めます。

- ▶ ログ・ファイル
- ▶ データベースのプロパティ
- ▶ Business Availability Center への登録

TVisionSetupInfo からの入力要求に対して情報を入力するときに、大括弧で囲まれた標準設定の値を受け入れるには、**Enter** キーを押します。受け入れない場合は、正しい値を入力して **Enter** キーを押します。空の値を指定するには、スペースバーを押してから **Enter** キーを押します。以降の各項では、入力の例が斜体で示されています。

TVisionSetupInfo は、各設定セクションでの設定が完了するたびに、どのファイルが更新されたかを知らせるメッセージを表示します。

例

Windows での **TVisionSetupInfo** セッションの例を次に示します。

This program collects configuration information in order to set up the TransactionVision environment. This includes:

- Location of software that TransactionVision depends upon such as the messaging middleware and the relational database system
- Parameters required to connect to the messaging middleware
- Parameters to connect to the database system
- Setup parameters for installed TransactionVision components

You will be prompted to input required configuration parameters.

If a default value is provided in [], pressing <Enter> will set the parameter to this default value. Pressing <Space><Enter> will set the parameter to an empty value.

Please specify name of the directory where you want to store your log files

[C:/PROGRA~1/HP/TRANSA~1/logs]:

TransactionVision Info(FileUpdated): File

"C:\PROGRA~1\HP\TRANSA~1\config\setup\Setup.properties" has been successfully updated

Modifying *.Logging.xml files to use log file directory

C:/PROGRA~1/HP/TRANSA~1/logsTransactionVision Info(FileUpdated): File

"C:\PROGRA~1\HP\TRANSA~1\config\logging\bpitveventimporter_loggingconfig.properties" has been successfully updated

Please provide your TransactionVision license key: PreSalesTraining@hp.com-020EF5-815082021D0482C

TransactionVision Info(FileUpdated): File

"C:\PROGRA~1\HP\TRANSA~1\config\license\License.properties" has been successfully updated

Database Settings

Retrieving database configuration parameters...

Type of Database? (DB2/Oracle/SQLServer) [Oracle]:

Name of the host the database is running on [your_database_host]: ros89891duh.rose.hp.com

Name of database TransactionVision connects to.

This is the Oracle database SID [your_database_name]: orcl

Enter the port number that the database listener is on [1521]:

Database user name []:system

User password []: rpctest

TransactionVision Info(FileUpdated): File

"C:\PROGRA~1\HP\TRANSA~1\config\datamgr\Database.properties" has been successfully updated

LW-SSO (Lightweight Single Sign-On) Settings

Enter the LW-SSO init string, used for the initialization of the enc algorithm. This value needs to be the same in all applications integ with LW-SSO. Unless you have modified the default value in other app such as BAC, you can just accept the default. [Xy6stqZ]:

Enter the LW-SSO application domain, used for LW-SSO cookie creation.This is the full or partial domain part of the fully-qualified hostn of the machine running the TransactionVision UI. (for example, 'mydomain.com') [hpqcorp.net]:

Enter a comma delimited list of LW-SSO protected domains. Applicable if other applications integrated with LW-SSO are running in differen domains. It is not necessary to include the TransationVision UI doma (for example, if BAC is in domain bac.com and Diagnostics is in domadiag.com, enter 'bac.com,diag.com'):TransactionVision Info(FileUpdated): File "C:\PROGRA~1\HP\TRANSA~1\cproperties" has been successfully updated

Generating script file for environment setup...

TransactionVision Info(NewFileCreated): File "C:\PROGRA~1\HP\TRANSA~nv.bat" has been successfully created

Initialize and verify database setup for TransactionVision

Do you wish to initialize the TransactionVision Database ? [y] :

Verifying database initialization for TransactionVision...TransactionVision Info(DBInitialized): Database has been properly initialized

Register TransactionVision with Business Availability Center

Do you wish to register TransactionVision with Business Availability Center at this time? [n] : y

Note: If your BAC server has Lightweight Single Sign On enabled, the hostnames of the TransactionVision UI and BAC servers must be a fully qualified domain name.

Please enter the hostname of the BAC server [] : ovresx1-vm11.rose.hp.com

Please enter the port of the BAC server [80] :

Do you wish to configure TransactionVision to access BAC using HTTPS? [n] :

TransactionVision has been configured to use the BAC server at: http://ovresx1-vm11.rose.hp.com:80

Do you wish to register the TransactionVision UI with BAC? [y] :

Note: If your BAC server has Lightweight Single Sign On enabled, the hostnames of the TransactionVision UI and BAC servers must be a fully qualified domain name.

Enter TransactionVision UI host [ROS89891DUH.americas.hpqcorp.net] :

Do you wish to configure BAC to access TransactionVision using HTTPS? [n] :

Enter TransactionVision UI port [21000] :

Do you wish to configure BAC to access TransactionVision using HTTPS? [n] :

Enter TransactionVision UI port [21000] :

Do you wish to register a TransactionVision Analyzer for RUM Data Publishing with BAC? [y] :

Enter the TransactionVision Analyzer host for RUM Data Publishing [ROS89891DUH.americas.hpqcorp.net] :

Enter RUM data publishing port [21113] :

User name for Basic Authentication to TransactionVision Analyzer used by Real User Monitor :

Password for Basic Authentication to TransactionVision Analyzer used by Real User Monitor :

The BAC server at http://ovresx1-vm11.rose.hp.com:80 will be configured with:

TransactionVision Web Server:
 http://ROS89891DUH.americas.hpqcorp.net:21000
 TransactionVision URL for RUM:
 http://ROS89891DUH.americas.hpqcorp.net:21113/tv_rum
 TransactionVision User for RUM:
 TransactionVision Password for RUM:

Please review the above settings and verify that they are correct.

Proceed with registration to BAC? [y] :

Registration with BAC was successful.

TransactionVision Info(TVisionSetupInfoSuccess): TVisionSetupInfo has completed successfully.

All program output and user input has been logged to
 "C:/PROGRA~1/HP/TRANSA~1/logs/setup.log".

Please start HP Business Availability Center by running
 %TVISION_HOME%\bin\SupervisorStart.bat

完了情報

設定が完了すると、TVisionSetupInfo は次のメッセージを表示します。

TransactionVision Info(TVisionSetupInfoSuccess): TVisionSetupInfo has completed successfully.

Business Availability Center の再起動

TVisionSetupInfo の実行後に、Business Availability Center を再起動する必要がある場合があります。再起動が必要な場合は、**TVisionSetupInfo** スクリプトが使用するべき適切なコマンドを次のように表示します。

Windows の場合

Please start HP Business Availability Center by running
 %TVISION_HOME%\bin\SupervisorStart.bat

UNIX の場合

Please start HP Business Availability Center by running
`$TVISION_HOME/bin/run_topaz start`

UI/Job サーバの管理

UI/Job サーバ・サービスは、HP Business Availability Center サービスによって管理されます。『**HP Business Availability Center デプロイメント・ガイド**』（PDF 形式）を参照してください。

第 13 章

UI/Job サーバのログ記録の設定

本章の内容

- ▶ ログ・ファイル (133 ページ)
- ▶ 循環的なログ記録 (133 ページ)
- ▶ トレース・ログ (135 ページ)
- ▶ Windows と UNIX のシステム・ログの使用 (135 ページ)

ログ・ファイル

標準設定では、すべての TransactionVision コンポーネントがエラー・メッセージと警告メッセージを適切なログ・ファイルに記録します。ログ・ファイルの場所は、TVisionSetupInfo または SensorSetup を実行したときに指定され、Setup.properties ファイルに保存されます。

UI/Job サーバは、エラー・メッセージを ui.log ファイルに記録します。

循環的なログ記録

標準設定では、UI/Job サーバは循環的なログ記録の形式を使用します。ログ・ファイルが設定された最大サイズに達すると、名前を変更されてバックアップ・ファイルとなり、新しい空のログ・ファイルが作成されます。標準設定では、最大ログ・サイズは 10 MB で、バックアップ・ログ・ファイルは 1 つです。

標準設定を使用した場合、たとえば UI/Job サーバ・ログ・ファイル ui.log のサイズが 10 MB に達すると、その名前が ui.log.1 に変更され、新しい ui.log ファイルが作成されます。設定を変更してバックアップ・ファイルの数を 2 個にすると、ui.log が 10 MB に達したときに以下のことが起こります。

- ▶ ui.log.2 が存在する場合は、それが削除されます。
- ▶ ui.log.1 の名前が ui.log.2 に変更されます。
- ▶ ui.log の名前が ui.log.1 に変更されます。
- ▶ 新しい ui.log が作成されます。

循環的なログ記録を望まない場合は、直線的なログ記録を使用するように設定を変更できます。その場合は、単一のログ・ファイルが生成されます。

< TVISION_HOME > /config/logging/*.Logging.xml ファイルは、使用されるログ記録の種類、最大ログ・ファイル・サイズ、およびバックアップ・ログ・ファイルの数を、各コンポーネントごとに指定します。たとえば、Sensor.Logging.xml は、Servlet Sensor と JMS Sensor に関する設定を指定します。このファイルには、次の例のようなエントリが含まれます。

```
<appender class="tvision.org.apache.log4j.RollingFileAppender"
name="SENSOR_LOGFILE">
  <param name="File" value="C:¥Program Files¥ C:¥Program
Files¥HP¥TransactionVision¥logs¥sensor.log"/>
  <param name="Append" value="true"/>
  <param name="MaxBackupIndex" value="2"/>
  <param name="MaxFileSize" value="10MB"/>
  <layout class="tvision.org.apache.log4j. PatternLayout">
    <param name="ConversionPattern" value="%d [%t] %-5p %c %x - %m%n"/>
  </layout>
</appender>
```

ログ・ファイルの最大サイズ

ログ・ファイルの最大サイズを変更するには、MaxFileSize パラメータの値を希望するサイズに変更します。指定する値の最後には、メガバイトとキロバイトを区別するために、MB か KB を付けます。

バックアップ・ログ・ファイルの最大数

バックアップ・ファイルの最大数を変更するには、MaxBackupIndex パラメータの値を希望するバックアップ・ファイル数に変更します。

循環的なログ記録から直線的なログ記録への変更

循環的なログ記録ではなく直線的なログ記録を使用するには、次の手順を実行します。

- 1 `appender class` の値を `RollingFileAppender` から `FileAppender` に変更します。たとえば、上記の例では、最初の行を次のように変更します。

```
<appender class="tvision.org.apache.log4j.FileAppender"  
name="SENSOR_LOGFILE">
```

- 2 `MaxBackupIndex` パラメータと `MaxFileSize` パラメータのエントリを削除します。

トレース・ログ

トレース・ログは、TransactionVision Analyzer が内部で何を行っているかについて詳細な情報を提供します。トレース・ログは、主に問題のトラブルシューティングに使用されるものであり、実運用環境ではオンにするべきではありません。

TransactionVision の Web ユーザ・インタフェースに対してトレース・ログを有効にするには、`< TVISION_HOME > /config/ui/UI.properties` ファイルで `trace` プロパティの値を `on` に設定します。この設定ファイルを変更した後、その変更を有効にするために、Web ユーザ・インタフェースを再起動する必要があります。TransactionVision の Web ユーザ・インタフェースを再起動するには、アプリケーション・サーバの管理コンソールを使用します。

Windows と UNIX のシステム・ログの使用

UNIX プラットフォームと Windows プラットフォームでは、出力をシステム・イベント・ログ (Windows の場合はイベント・ログ、UNIX の場合は `syslog`) に記録するように、TransactionVision を設定することができます。そのために必要なログ設定ファイルの例は、`TVISION_HOME/config/logging/system/*/Sensor.Logging.xml` の中にあります。

Windows でも UNIX でも、専用のイベント・アペンダを定義する必要があります。

Windows イベント・アペンダ

次の例は、イベント・ログを使用するための Windows イベント・アペンダを設定する方法を示しています。

```
<appender name="NT_EVENT_LOG" class="tvision.org.apache.log4j.nt
.NTEventLogAppender">
  <layout class="tvision.org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%d [%t] - %m%n"/>
  </layout>
</appender>
```

その後、NT_EVENT_LOG を選択したカテゴリ定義の中で参照できます。次に例を示します。

```
<category additivity="false" class="com.bristol.tvision.util.log.XCategory"
name="sensorLog">
  <priority class="com.bristol.tvision.util.log.XPriority" value="info"/>
  <appender-ref ref="NT_EVENT_LOG"/>
</category>
```

Windows ではさらに、特別な DLL をパスに追加する必要があります。この DLL は NTEventLogAppender.dll で、config¥logging¥system¥bin ディレクトリの中にあります。次に例を示します。

```
set path=%TVISION_HOME%¥config¥logging¥system¥bin;%PATH%
```

UNIX イベント・アペンダ

次の例は、syslog を使用するための UNIX イベント・アペンダを示しています。

```
<appender name="SYSLOG" class="tvision.org.apache.log4j.net.SyslogAppender">
  <param name="SyslogHost" value="localhost"/>
  <param name="Facility" value="local0"/>
  <layout class="tvision.org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="[%t] %-5p %c %x
- %m%n"/>
  </layout>
</appender>
```

SyslogHost パラメータと Facility パラメータを環境に合わせて適切に指定します。

第 IV 部

Sensor と Agent のインストールおよび設定

第 14 章

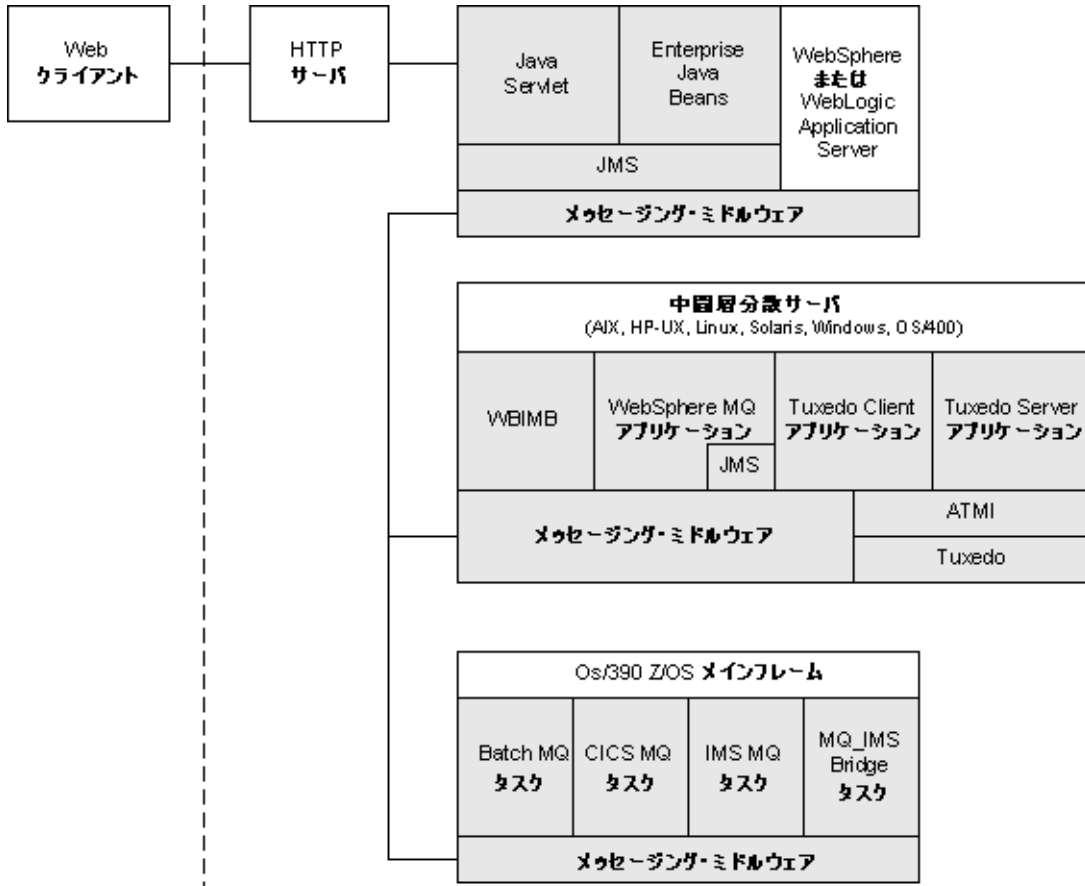
TransactionVision Sensor を インストールする準備

本章の内容

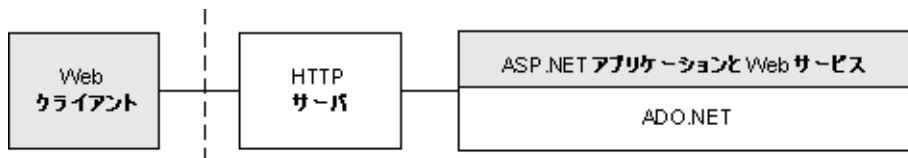
- ▶ 監視できるアプリケーション (140 ページ)
- ▶ 使用可能な Agent と Sensor の種類 (141 ページ)

監視できるアプリケーション

次の図で灰色で表示された部分は、TransactionVision がイベントを追跡する対象にできる Web アプリケーションの部分を表しています。



ASP.NET アプリケーションも監視できます。



NET Remoting クライアントおよびサーバ・アプリケーションも監視できます。

NonStop TMF アプリケーションも監視できます。



使用可能な Agent と Sensor の種類

TransactionVision は、以下の種類の Sensor と Agent を提供します。

- ▶ WebSphere MQ (WMQ) Sensor
- ▶ Java Agent
 - ▶ Servlet Sensor
 - ▶ JMS Sensor
 - ▶ EJB Sensor
 - ▶ JDBC Sensor
- ▶ CICS Sensor
- ▶ .NET Agent
- ▶ BEA Tuxedo Sensor
- ▶ NonStop TMF Sensor

WebSphere MQ (WMQ) Sensor

WebSphere MQ Sensor は、MQ API 呼び出しを追跡します。これらの API 呼び出しには、MQ API セット全体（主な API は MQPUT, MQGET, MQCONN, MQDISC, MQOPEN, MQCLOSE など）が含まれます。TransactionVision が分散プラットフォームで提供する WebSphere MQ Sensor には、WebSphere MQ Library Sensor と WebSphere MQ API Exit Sensor の 2 種類があります。これらの Sensor は両方とも、MQ API 呼び出しからの同じ情報を報告します。この 2 種類の主な違いは、それらが MQ API 呼び出しを傍受するメカニズム、それぞれの用途、およびそれらがシステムから収集するデータの量です。

- ▶ **WebSphere MQ Library Sensor** は、共有ライブラリ（または DLL）傍受方式によって、分散プラットフォームで WebSphere MQ API 呼び出しを傍受します。これには、アプリケーション・ライブラリ・パスの中で TransactionVision の Sensor ライブラリを WebSphere MQ ライブラリの前に置くことが含まれます。この方法は、特定のアプリケーションについて MQ API を追跡する必要がある場合に役立ちます。
- ▶ **WebSphere MQ API Exit Sensor** は、WebSphere MQ v5.3 以降の配布先プラットフォームで使用可能な WebSphere MQ API 出口サポートを使用します。この Sensor は、キュー・マネージャへの出口として登録され、キュー・マネージャに接続するプログラムが WebSphere MQ API を呼び出したときに呼び出されます。この方法は、キュー・マネージャ上、特にリスナおよびチャネル・エージェント内のすべてのアプリケーションから MQ イベントを収集する場合に適しています。
- ▶ **z/OS WebSphere MQ Sensor** は、IBM z/OS システム上の CICS、バッチ、および IMS 環境で MQI API 呼び出しを追跡するために提供されています。CICS 環境では、CICS アダプタが WebSphere MQ 用に提供している API 交差出口が、MQ API を傍受するために使用されます。バッチ環境と IMS 環境で MQ API 呼び出しを傍受するためには、アプリケーションを Sensor と再バインドする必要があります。

WebSphere MQ については、以下の補足的な Sensor が用意されています。

- ▶ **Proxy Sensor** は、ビジネス・トランザクションを、TransactionVision Sensor ライブラリを使って監視されていないプロセス（たとえば、Sensor の監視対象となるアプリケーションと、外部パートナー・システムのように Sensor がインストールされていないシステムで実行されているアプリケーションとの間のイベント）に相関させます。
- ▶ **WebSphere Business Integration Sensor**（以前の MQSI Sensor）は、さまざまなメッセージ・フローを区別し、WBI 内の個々の論理トランザクション・パスを識別します。この Sensor は、メッセージ・フローの通常の実行パスに挿入されるトレース・ノードと、メッセージ・フローの失敗パスに挿入される失敗ノードを提供する、WBI プラグインです。これらのノードは、WBI 内のメッセージ・フローの追跡を可能にする MQSI2TRACE イベントを生成します。

- ▶ **WebSphere MQ-IMS Bridge Sensor** は、呼び出し元のアプリケーションからの WebSphere MQ API 呼び出しではなく、WebSphere MQ-IMS ブリッジ・メッセージを追跡します。MQ-IMS ブリッジは、WebSphere MQ アプリケーションが IMS トランザクションを呼び出してそれらから応答メッセージを受信することを可能にするコンポーネントです。MQ-IMS Bridge Sensor は、ブリッジに入ってくる MQ メッセージを追跡し、それらを IMS から受信した応答と関連させます。ブリッジに入って出ていく各メッセージについて、2 つのイベント、MQIMS_BRIDGE_ENTRY と MQIMS_BRIDGE_EXIT が生成されます。これらのイベントには、MQ メッセージ・ヘッダと、どの IMS トランザクションが呼び出されたかに関する情報が含まれています。

Java Agent

- ▶ **Servlet Sensor** は、J2EE アプリケーション・サーバ内のサーブレット・メソッドを追跡します。この Sensor は、HTTP_POST, HTTP_GET, HTTP_PUT などの、J2EE コンテナに対するメソッド呼び出しを引き起こす HTTP 呼び出しを追跡します。Servlet Sensor は、各呼び出しの入口と出口でイベントを収集するようにサーブレットを設定することによって、これらのメソッド呼び出しを追跡します。
- ▶ **JMS Sensor** は、独立した Java アプリケーションからの、および J2EE アプリケーション・サーバからの、WebSphere MQ Java Message Service (JMS) または TIBCO EMS イベントを追跡します。この Sensor は、送信、受信などの JMS インタフェース・メソッドを追跡します。これらのメソッドは、各呼び出しの入口と出口でイベントを収集するように JMS ライブラリを設定することによって追跡されます。
- ▶ **EJB Sensor** は、J2EE アプリケーション・サーバ内でビジネス・ロジックを通じてトランザクションを追跡します。この Sensor は、エンティティ・ビーン、セッション・ビーン、またはメッセージ駆動ビーン内のすべてのパブリック・ビジネス・メソッドを追跡します。この Sensor はビジネス・メソッドに加えて、ejbCreate, ejbPostCreate, ejbRemove, ejbLoad, ejbStore, および onMessage メソッドも追跡します。これらのメソッドは Sensor によって、各呼び出しの入口と出口でイベントを収集するように設定されます。

- ▶ **JDBC Sensor** を使用すると、リレーショナル・データベースに対して JDBC API を通じて行われた SQL 呼び出しとトランザクションの API およびタイミング情報を収集して分析することができます。

TransactionVision Java Sensor (JMS, Servlet, EJB, および JDBC) と Diagnostics Java Probe の機能が組み合わされて単一のコンポーネントになったものが、HP Diagnostics/TransactionVision Java Agent です。Java Agent は、アプリケーションからのイベントを測定して捕捉し、その情報を Diagnostics Server, TransactionVision Analyzer, またはその両方に送ります。本リリースでは、Diagnostics 環境内で Java Probe として、または TransactionVision 環境内で Java Sensor として動作するように、Java Agent を設定することができます。両方が混在する環境では、Java Agent が同時に Probe と Sensor の両方として動作できます。詳細については、第 15 章「Java Agent のインストールと設定」を参照してください。

CICS Sensor

CICS Sensor は、メインフレーム環境内のトランザクションを追跡するために、WebSphere MQ CICS 以外のイベントを収集します。CICS Sensor は、ファイル制御、一時格納場所、トランジェント・データ、間隔制御、およびプログラム制御という 5 種類のイベントのデータを収集します。すべての種類のイベントについて、トランザクション ID、ユーザ ID、ターミナル ID、および SYSID が追跡されます。収集されるその他の情報は、イベントの種類によって異なります。

.NET Agent

.NET Agent は、ASP.NET 環境内の Webservice を追跡します。.NET Agent は、サーバ上の Webservice メソッドとクライアントから送信される Webservice 呼び出しの入口と出口でイベントを収集するように .NET コードを設定することによって、それらの Webservice メソッド呼び出しを追跡します。

.NET Agent はまた、ASP.NET 環境、ADO、およびリモート・イベントにおける HTTP_POST, HTTP_GET, HTTP_PUT などの HTTP 呼び出しも追跡します。

BEA Tuxedo Sensor

BEA Tuxedo Sensor は、C 環境と C++ 環境で Tuxedo ATMI 呼び出しを行うアプリケーションを監視します。この Sensor は、ATMI メソッドを傍受して収集します。その最小限のセットには、tpenqueue, tpdequeue, および tpcall が含まれます。

収集されるメソッドについて、API+テクノロジー・データを収集するモードと、API+テクノロジー・データ+ペイロード・データを収集するモードがあります。

この Sensor は、すべてのテクノロジーに共通の条件と、Tuxedo キュー・スペース、キュー名、および Tuxedo サービス名を含む（ただし、それだけに限定されない）Tuxedo ATMI 固有の条件に基づいた、データ収集のフィルタリングもサポートしています。

NonStop TMF Sensor

NonStop TMF Sensor は、監査対象の Enscribe ファイル・システムへのアクセスを追跡します。HP NonStop 上の監査対象のトランザクションは、TMF 監査証跡に記録されます。TMF は NonStop システム上の監査対象ファイルを保護するので、システム上のデータ全体に対するすべての変更（追加、削除、修正）のリポジトリとして機能します。

この Sensor は TMF 監査証跡を読んで、ユーザが設定したフィルタ条件に一致する Enscribe ファイルに対するアクセスをすべて追跡します。

第 15 章

Java Agent のインストールと設定

本章では、Windows と UNIX での HP Diagnostics/TransactionVision Java Agent のインストールと設定の手順を説明します。

本章の内容

- ▶ Java Agent のインストールと設定について (147 ページ)
- ▶ Windows での Java Agent のインストールと設定 (149 ページ)
- ▶ UNIX での Java Agent のインストールと設定 (163 ページ)
- ▶ Java Agent のサイレント・インストール (172 ページ)
- ▶ JRE Instrumenter の実行 (174 ページ)
- ▶ アプリケーション・サーバの設定 (183 ページ)
- ▶ メッセージング・システム・プロバイダの設定 (184 ページ)
- ▶ カスタム・ユーザ・イベントの設定 (186 ページ)

Java Agent のインストールと設定について

Java Agent は、TransactionVision Java Sensor (JMS, Servlet, JDBC, および EJB) と Diagnostics Java Probe の機能を 1 つのコンポーネントに組み合わせたものです。Java Agent は、Diagnostics 環境では Java Probe として、TransactionVision 環境と複合環境では Java Sensor として機能するように設定できます。同時に Probe と Sensor の両方として機能することもできます。

Java Agent を TransactionVision Java Sensor として使用するには、以下の操作を実行する必要があります。

1 HP Diagnostics/TransactionVision Java Agent をインストールします。

Java Agent は、監視するアプリケーションのホスト・マシンにインストールされます。詳細については、149 ページ「Windows での Java Agent のインストールと設定」と 163 ページ「UNIX での Java Agent のインストールと設定」を参照してください。

2 Java Agent を設定します。

Java Agent は、TransactionVision Java Sensor, J2EE Probe, またはその両方として機能するように設定されます。本書では、Java Agent を TransactionVision Java Sensor として設定する手順を説明します。詳細については、151 ページ「Windows 上で TransactionVision Java Sensor として機能するように Java Agent を設定する」と 165 ページ「UNIX 上で TransactionVision Java Sensor として機能するように Java Agent を設定する」を参照してください。

3 アプリケーション・サーバを設定します。

Java Sensor を使ってアプリケーションを監視するには、インストールする JRE を設定して (174 ページ「JRE Instrumenter の実行」を参照)、アプリケーション・サーバを設定する (183 ページ「アプリケーション・サーバの設定」を参照) 必要があります。

インストール・ファイル

次の表は、TransactionVision Java Agent のインストール・ファイルの名前をプラットフォーム別に示しています。

プラットフォーム	ファイル
Windows	JavaAgentSetup_win_8_00.exe
AIX	JavaAgentSetup_ibm_8_00.bin
Linux	JavaAgentSetup_linux_8_00.bin
Solaris	JavaAgentSetup_sol_8_00.bin

Windows での Java Agent のインストールと設定

Windows マシンに Java Agent をインストールするための詳細な手順は次のとおりです。また、これらの手順は、グラフィック・インストーラを使って Java Agent を UNIX マシンにインストールする際にも使用します。

Java Agent, 従来の J2EE Probe, または従来の TransactionVision 5.0 Sensor がすでにホスト・マシンにインストールされている場合は、Java Agent をインストールする前にそれをアンインストールする必要があります。

本項の内容

- ▶ Windows 上でのインストーラの起動
- ▶ Windows 上でのインストールの実行
- ▶ Windows 上で TransactionVision Java Sensor として機能するように Java Agent を設定する

Windows 上でのインストーラの起動

Java Agent インストーラは、HP ソフトウェア Web サイトから、Diagnostics または TransactionVision 製品のディスクから、あるいは Business Availability Center の [ダウンロード] ページから起動できます。

Java Agent をインストールするユーザは、Administrators グループに属していなければなりません。

HP ソフトウェア Web サイトからインストーラを起動する場合

- 1 HP ソフトウェア Web サイトの [Download Center HP- BTO Software] に移動します。
- 2 [Keyword] フィールドに TransactionVision と入力し、[Refine Search by Resource Type] フィールドに Trial Software と入力して [Search] をクリックします。
- 3 150 ページ「Windows 上でのインストールの実行」に進みます。

Business Availability Center からインストーラを起動する場合

- 4 Business Availability Center のトップ・メニューから [管理] > [プラットフォーム] を選択し、[セットアップと保守] タブをクリックします。

- 5 [ダウンロード] ページで、適切なリンクをクリックして Java Agent installer for Windows をダウンロードします。
- 6 150 ページ「Windows 上でのインストールの実行」に進みます。

Business Availability Center 製品のインストール・ディスクからインストーラを起動する場合

- 1 インストール・ディスクのルート・ディレクトリにある **setup.exe** ファイルを実行します。Diagnostics セットアップ・プログラムが起動し、インストール・メニュー・ページが表示されます。
- 2 インストール・メニュー・ページで、**[Diagnostics/TransactionVision Agent for Java]** を選択してインストーラを起動します。

Business Availability Center 製品のインストール DVD からインストーラを起動する場合

- 3 HP Business Availability Center のインストール DVD から、使用しているプラットフォームに対応する実行可能ファイルを選択して実行します。詳細については、148 ページ「インストール・ファイル」を参照してください。
- 4 150 ページ「Windows 上でのインストールの実行」に進みます。

Windows 上でのインストールの実行

インストーラを起動すると、ソフトウェア使用許諾契約書が開き、インストールを実行することができます。

注： 当該コンピュータで現在実行されている Windows プログラムを、自動バックアップ・プログラムも含めてすべて終了します。ウィルス対策、スパイウェア防止、および脅威対策プログラムを終了する必要はありません。

Windows マシンに Java Agent をインストールするには、次の手順を実行します。

- 1 エンド・ユーザ使用許諾契約書に同意します。

使用許諾契約書を読み、**[使用条件の条項に同意します]** を選択します。

[次へ] をクリックして先に進みます。

- 2 Agent のインストール先を指定します。

標準設定のディレクトリを受け入れるか、**[インストール先ディレクトリ名]** ボックスにインストール・ディレクトリのパスを入力するか、または **[ブラウズ]** をクリックしてインストール・ディレクトリを指定して、ほかの場所を選択します。

ディレクトリ名に含めることができるのは、英語の文字だけです。

[次へ] をクリックして先に進みます。

- 3 サマリ情報を確認します。

インストール先ディレクトリとサイズの要約がリストされます。

[次へ] をクリックして先に進みます。

- 4 インストールのサマリ情報を確認します。サマリ情報パネルにエラーが表示されていないければ、**[次へ]** をクリックして進みます。

Java Agent セットアップ・モジュールが起動します。これにより、Java Agent の設定が始まります。

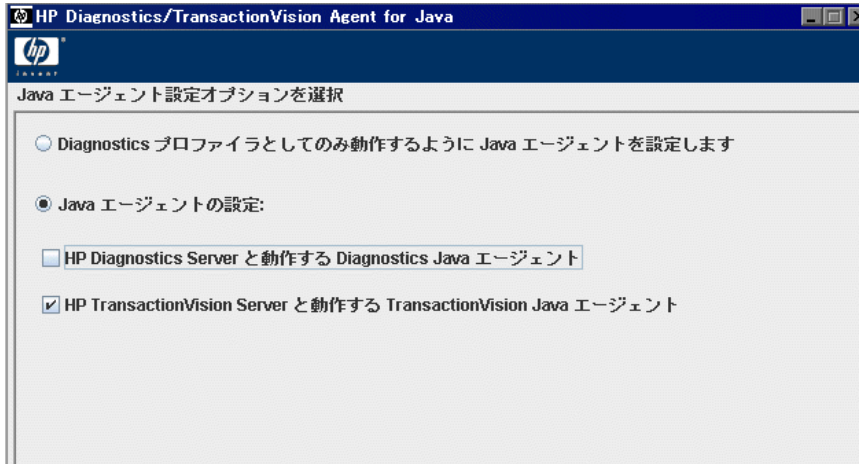
Windows 上で TransactionVision Java Sensor として機能するように Java Agent を設定する

ここでは、Java Agent セットアップ・モジュールのユーザ・インタフェースを使って Java Agent を TransactionVision Java Sensor として機能するように設定する方法について、詳しい手順を説明します。

Java Agent セットアップ・モジュールは、Java Agent のインストールの最後に自動的に起動します。**[スタート]** > **[すべてのプログラム]** > **[HP Java Agent]** > **[Setup Module]** を選択することにより、いつでもセットアップ・モジュールを起動できます。

Java Agent を TransactionVision Java Sensor として機能するように設定するには、次の手順を実行します。

- 1 [HP TransactionVision Server と動作する TransactionVision Java エージェント] オプションを選択します。

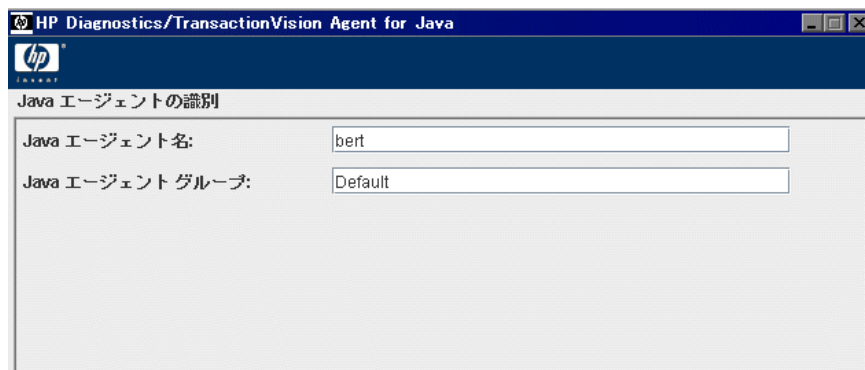


次のオプションも選べます。

- ▶ **Diagnostics プロファイラ としてのみ動作するように Java エージェントを設定します。** Java Agent を Diagnostics Profiler としてのみ設定する場合は、『**HP Diagnostics インストールおよび設定ガイド**』の「Java Agent の Profiler 専用の設定」を参照してください。
- ▶ **HP Diagnostics Server と動作する Diagnostics Java エージェント。** Java Agent を J2EE Probe として Diagnostics Server と連携して機能するように設定する場合は、『**HP Diagnostics インストールおよび設定ガイド**』の「Configuring the Probe to Work with a Diagnostics Server」を参照してください。
- ▶ **HP TransactionVision Server と動作する TransactionVision Java エージェント。** Java Agent を、Diagnostics Server と連携する J2EE Probe として、および TransactionVision Java Sensor として機能するように設定するには、両方のチェック・ボックスを選択して手順 2 に進みます。この手順の後で、まず Java Agent を J2EE Probe として設定し（『**HP Diagnostics インストールおよび設定ガイド**』の「Configuring the Java Agent to Work with a Diagnostics Server」を参照）、その後 J2EE TransactionVision Java Sensor として設定します（手順 3 から）。

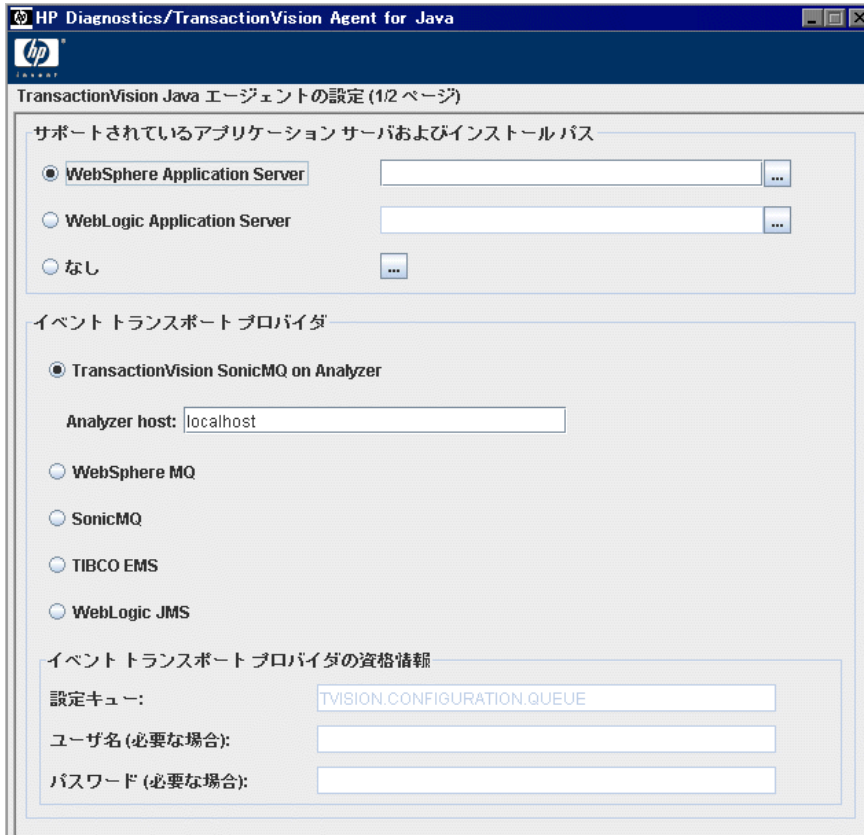
[次へ] をクリックして先に進みます。

- 2 Java Agent に名前を割り当て、それが属するグループを指定します。



- ▶ Java Agent 名には、TransactionVision 内で Agent を一意に識別する名前を入力します。名前に使用できる文字は、-, _ およびすべての英数字です。Agent 名が Java Sensor 名になるように割り当てられます。
Agent に名前を割り当てる際は、Agent が監視しているアプリケーションと Java Sensor の種類を識別しやすい名前を選択してください。
- ▶ Java Agent グループ名に対しては、既存のグループの名前か新規作成するグループの名前を入力します。Agent グループ名では大文字と小文字が区別されます。
[次へ] をクリックして先に進みます。

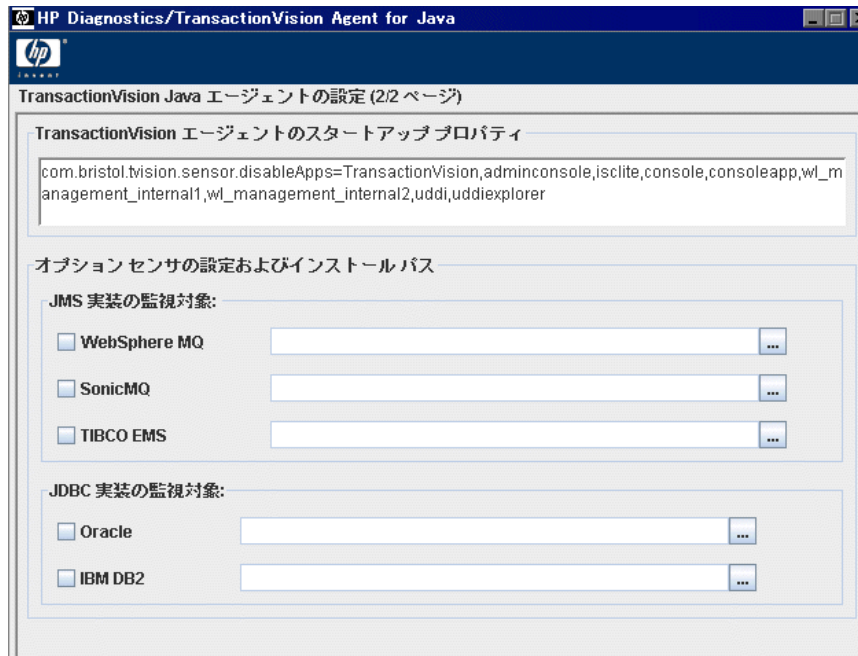
- 3 監視するアプリケーション・サーバとそのインストール・ディレクトリを選択します。イベント配送プロバイダを選択し、資格情報を指定します。



- ▶ テキスト・ボックスの中を右クリックすると、ファイル選択ダイアログが開きます。
- ▶ TransactionVision に付属の SonicMQ イベント配送プロバイダを使用する場合は、Analyzer が実行されているホストの名前を指定します。
- ▶ TransactionVision Analyzer が別のキューを使用する場合は、構成キューの名前を変更します。
- ▶ イベント配送プロバイダ用のユーザ名とパスワードを入力する必要がある場合は、それらを入力します。
- ▶ **[次へ]** をクリックして次に進みます。

4 監視対象となる JMS および JDBC システムを設定します。

対応するシステムのインストール・ディレクトリ・パスを入力します。テキスト・ボックスの中を右クリックすると、ファイル選択ダイアログが開きます。



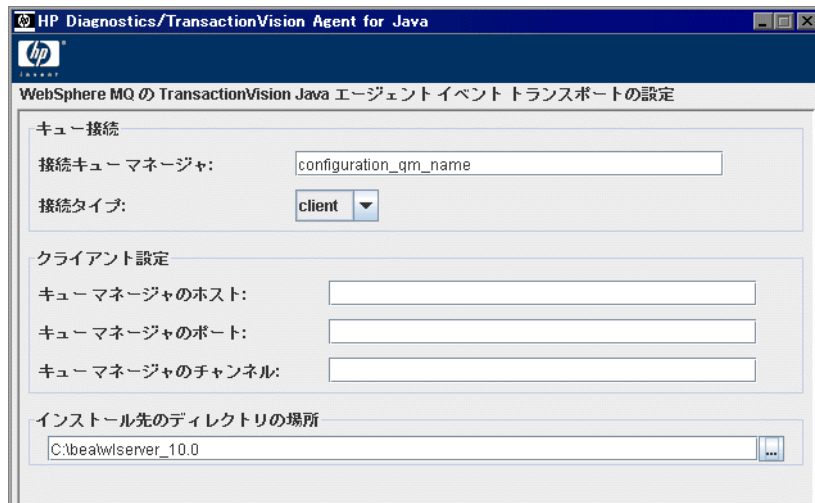
[TransactionVision Java エージェントの設定] フィールドでは、Agent の操作とインストールメンテナーに関するディレクティブを指定します。通常は、HP カスタマ・サポートから別の指示がないかぎり、標準設定をそのまま受け入れます。プロパティ間はセミコロンで区切ります。

注 : BEA JMS for WebLogic 8.1.x は自動的に監視されます。WebLogic アプリケーション・サーバに統合されるので、追加の設定は不要です。

5 JMS トランスポートを設定します。次に表示されるダイアログは、使用されるメッセージング・ミドルウェアによって異なります。

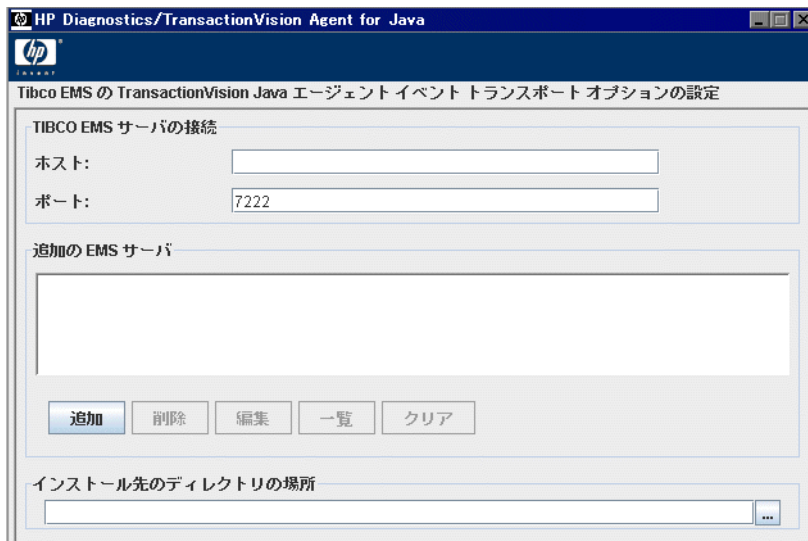
- ▶ 通信リンク・トランスポートとして WebSphere MQ を選択した場合は、WebSphere MQ JMS サーバ・バインディングまたはクライアント接続を使ってトランスポート設定を行います。

注：64 ビット Windows で WebSphere MQ 6.0 を使用していて、64 ビット JVM を監視する場合は、サーバではなくクライアントを選択する必要があります。WebSphere MQ 6.0 は、64 ビット Windows プラットフォームでのサーバ・バインディングをサポートしていません。

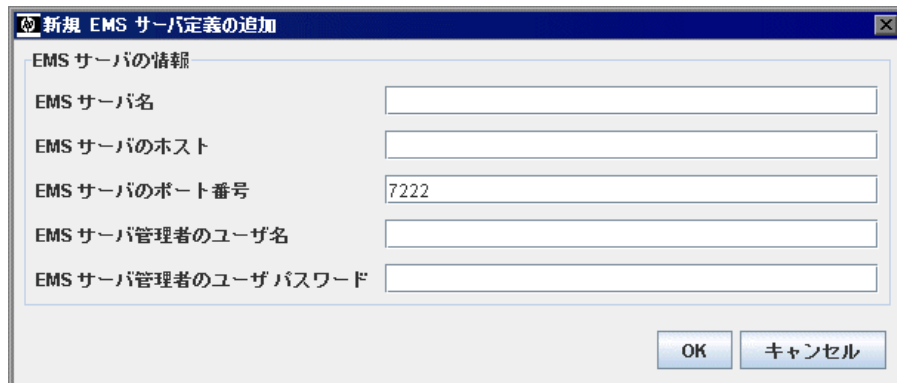


- ▶ 構成キュー・マネージャの名前を入力します。
- ▶ クライアント接続を選択した場合は、クライアントのキュー・マネージャ設定情報（ホスト、ポート、およびチャンネル）を入力します。
- ▶ WebSphere MQ のインストール場所を入力するか、参照機能を使って選択します。

- ▶ 通信リンク・トランスポートとして TIBCO EMS を選択した場合は、TIBCO EMS についてトランスポート設定を行います。



- ▶ ホスト名を入力します。必要に応じて、標準設定のポートを変更できます。
- ▶ TIBCO EMS のインストール場所を入力するか、参照機能を使って選択します。
- ▶ EMS サーバの追加と定義ができます。[追加] をクリックすると、追加ダイアログが開きます。



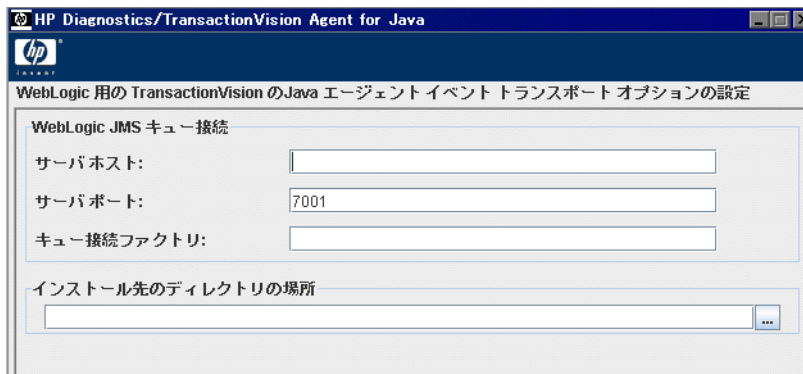
サーバ定義を入力し、[OK] をクリックします。定義を変更するには、サーバを選択して [編集] をクリックします。選択したサーバの定義を表示するには、[一覧] をクリックします。

- ▶ 通信リンク・トランスポートとして SonicMQ を選択した場合は、SonicMQ についてトランスポート設定を行います。



- ▶ 別のプロトコルを使用していないかぎり、標準設定の tcp プロトコルを使用します。
- ▶ ホスト名を入力します。必要に応じて、標準設定のポートを変更できます。
- ▶ SonicMQ のインストール場所を入力するか、参照機能を使って選択します。

- ▶ 通信リンク・トランスポートとして WebLogic JMS を選択した場合は、WebLogic JMS についてトランスポート設定を行います。



- ▶ ホスト名を入力します。必要に応じて、標準設定のポートを変更できます。
- ▶ キュー接続ファクトリを入力します。
- ▶ WebLogic JMS のインストール場所を入力するか、参照機能を使って選択します。通常、これは WebLogic アプリケーション・サーバのインストール場所と同じです。

いつでも任意のページに移動して必要な変更を行うことができます。

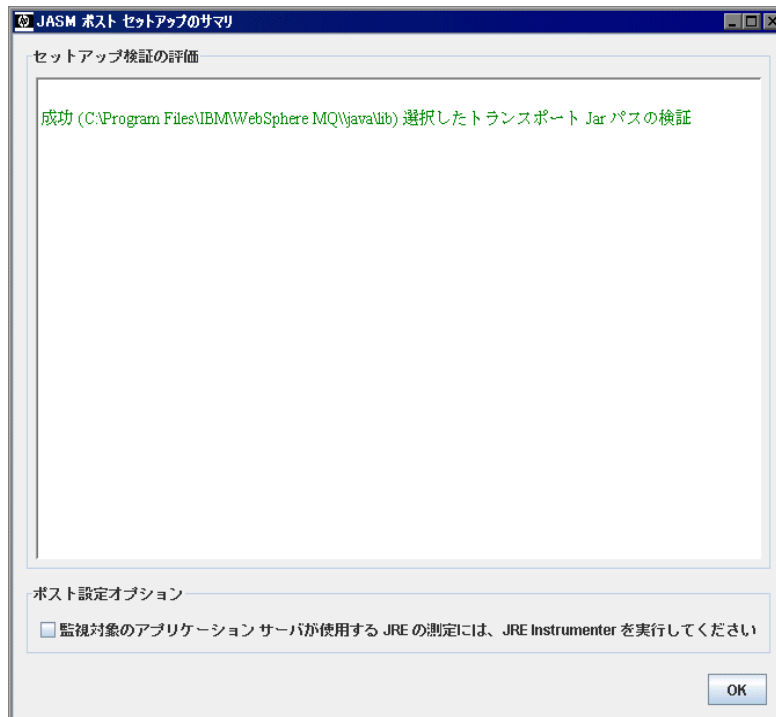
6 設定を保存します。

すべての設定が完了したら、**[完了]** をクリックして設定を保存します。この操作によって、Diagnostics と TransactionVision の設定ファイルが両方とも変更されます。

設定後のオプション

設定ファイルに対する変更が終わると、Java Agent セットアップ・モジュールが、システムにインストールされているさまざまなソフトウェアのバージョンに基づいて、自動的にマスタ・インストールメンテーション・ファイルを生成します。このプロセスには数分かかります。このプロセスの間、待機ダイアログが表示されています。

Java Agent セットアップ・モジュールは最後に、構成の設定を検証してテストするために、一連のテストを実行します。



いずれかの検証が失敗した場合は、トランスポート設定をチェックして、JMS サーバまたはキュー・マネージャが正しい設定で実行されているかどうかを確認してください。

[監視対象のアプリケーション サーバが使用する JRE の測定には、JRE Instrumenter を実行してください。] を選択すれば、JRE Instrumenter が自動的に実行されます。標準設定では、このオプションは選択されません。使用している JRE のバージョンが 1.5 より古い場合は、このチェック・ボックスを選択するか、手動で JRE Instrumenter を実行する必要があります。JRE Instrumenter の詳細と、それを手動で実行する方法については、174 ページ「JRE Instrumenter の実行」を参照してください。

Windows 上のアプリケーション内で Java Agent を有効にする

Java 1.5 以降の場合

- ▶ JRE 1.5 以降で動作しているアプリケーションを監視するために Java Agent を有効にするには、そのアプリケーションを起動する Java コマンド行に次の JVM オプションを追加します。

```
java -javaagent: < java_agent_install_dir > %DiagnosticsAgent%  
lib%probeagent.jar
```

< java_agent_install_dir > は、Java Agent のインストール・ディレクトリのパスです。標準設定のパスは **C:%MercuryDiagnostics%JavaAgent** です。

- ▶ アプリケーション・サーバに対して Java Agent を有効にする方法については、183 ページ「アプリケーション・サーバの設定」を参照してください。

Java 1.4 の場合

JRE バージョン 1.4 で動作しているアプリケーションまたはアプリケーション・サーバ (WebSphere 5.1, 6.0 や WebLogic 8.1 など) の場合は、Java Agent の JRE Instrumenter ツールを実行して、そのアプリケーションまたはアプリケーション・サーバが使用している JRE をインストールメントする必要があります。詳細については、174 ページ「JRE Instrumenter の実行」を参照してください。

UNIX 上のアプリケーション内で Java Agent を有効にする

Java 1.5 以降の場合

- ▶ JRE 1.5 以降で動作しているアプリケーションを監視するために Java Agent を有効にするには、そのアプリケーションを起動する Java コマンド行に次の JVM オプションを追加します。

```
java -javaagent: < java_agent_install_dir >  
/DiagnosticsAgent/lib/probeagent.jar
```

< java_agent_install_dir > は、Java Agent のインストール・ディレクトリのパスです。標準設定のパスは **/opt/MercuryDiagnostics/JavaAgent** です。

- ▶ アプリケーション・サーバに対して Java Agent を有効にする方法については、183 ページ「アプリケーション・サーバの設定」を参照してください。

Java 1.4 の場合

JRE バージョン 1.4 で動作しているアプリケーションまたはアプリケーション・サーバ（WebSphere 5.1, 6.0 や WebLogic 8.1 など）の場合は、Java Agent の JRE Instrumenter ツールを実行して、そのアプリケーションまたはアプリケーション・サーバが使用している JRE をインストゥルメントする必要があります。詳細については、174 ページ「JRE Instrumenter の実行」を参照してください。

WebSphere MQ に関する注意点

通信トランスポートとして WebSphere MQ を使用していて、Java Agent のセットアップ時に接続タイプとしてサーバ（標準設定）を選択した場合は、システムのライブラリ・パス環境変数に WebSphere MQ java/lib へのパスを追加する必要があります。次に例を示します。

AIX の場合は、次の行を追加します。

```
set LIBPATH=$LIBPATH:/usr/mqm/java/lib  
export LIBPATH
```

Solaris または Linux の場合は、次の行を追加します。

```
set LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/mqm/java/lib  
export LD_LIBRARY_PATH
```

JVM が 64 ビットの場合は、lib を lib64 で置き換えてください。

Windows の場合は、Web Sphere MQ をインストールするときにこのパスが PATH 環境変数に追加されているので、通常はこのような設定を行う必要はありません。

UNIX での Java Agent のインストールと設定

複数の UNIX プラットフォーム用に Java Agent インストーラが用意されています。ここでは、グラフィック・モード・インストールまたはコンソール・モード・インストールを使って、ほとんどの UNIX 環境に Java Agent をインストールするのに必要な手順を紹介します。

以下の手順とスクリーン・ショットは、AIX マシンに Agent をインストールするためのものです。ほかの認定 UNIX プラットフォームでも同じ手順でインストールすることができます。

Java Agent, 従来の J2EE Probe, または従来の TransactionVision 5.0 Sensor がすでにホスト・マシンにインストールされている場合は、Java Agent をインストールする前にそれをアンインストールする必要があります。

UNIX でのインストーラのダウンロード

Java Agent インストーラは、HP ソフトウェア Web サイトから、Diagnostics または TransactionVision 製品のディスクから、あるいは Business Availability Center の [ダウンロード] ページからダウンロードできます。

Java Agent をインストールするには root ユーザである必要があります。

製品インストール・ディスクからインストーラをコピーするには

- 1 < HP TransactionVision インストール・ディスク >
/TransactionVision_Installers ディレクトリから、TransactionVision サーバがインストールされるマシンにインストーラ **JavaAgentSetup <プラットフォーム>_8_00.bin** をコピーします。
- 2 164 ページ「UNIX 上でのインストールの実行」に進みます。

[ダウンロード] ページからインストーラをダウンロードするには (Business Availability Center ユーザの場合)

- 3 Business Availability Center のトップ・メニューから [管理] > [プラットフォーム] を選択し、[セットアップと保守] タブをクリックします。

- 4 [ダウンロード] ページで、環境に適したインストーラのリンクをクリックし、Agent をインストールするマシンに保存します。

UNIX 上でのインストールの実行

Java Agent をインストールするマシンにインストーラをコピーしたら、インストールを実行できます。

UNIX マシンに Java Agent をインストールするには

- 1 インストーラを実行します。

必要に応じて、インストーラ・ファイルのモードを変更して実行可能にします。

- ▶ root ユーザとしてログインしていることを確認します。
- ▶ コンソール・モードでインストーラを実行するには、UNIX コマンド・プロンプトで次のように入力します。

```
./JavaAgentSetup_ <プラットフォーム> _8_00.bin -console
```

インストーラに、次の手順で示されるようにコンソール・モードでインストール・プロンプトが表示されます。

- ▶ グラフィック・モードでインストーラを実行するには、UNIX コマンド・プロンプトで次のように入力します。

```
xhost + # コンソールに UI を表示できる
```

```
export DISPLAY= <ホスト名> :0.0
```

```
./JavaAgentSetup_ <プラットフォーム> _8_00.bin
```

インストーラには、149 ページ「Windows での Java Agent のインストールと設定」のように、Windows インストーラのものと同じ画面が表示されます。

- 2 エンド・ユーザ使用許諾契約書に同意します。

エンド・ユーザのソフトウェア使用許諾契約書が表示されます。

契約書を読みます。読んでいるときに、**Enter** キーを押すと次のページに進み、**q** を入力すると使用許諾契約書の末尾にジャンプします。

契約書の規約に同意するために、数字の **1** を入力して **Enter** キーを押します。

0 (ゼロ) を入力して **Enter** キーを押した後、数字の **1** を入力して **Enter** キーを押し、インストールを続行します。

- 3 Agent のインストール先を指定します。

[**インストール先ディレクトリ名**] プロンプトで、括弧内に表示される標準設定のインストール先を受け入れます。

数字の **1** を入力して **Enter** キーを押し、インストールを続行します。

- 4 インストール先を確認します。

インストール先と推定サイズがリストされます。

インストール先とそのサイズに問題がなければ、数字の **1** を入力して **Enter** キーを押し、インストールを開始します。

インストールには数分かかります。

Java Agent セットアップ・モジュールが起動します。

UNIX 上で TransactionVision Java Sensor として機能するように Java Agent を設定する

ここでは、ほとんどの UNIX 環境で Java Agent セットアップ・モジュールをグラフィック・モードまたはコンソール・モードで使用して、Java Agent を TransactionVision Java Sensor として設定するために必要な手順を説明します。

Java Agent セットアップ・モジュールは、インストール・プログラムの最後に自動的に起動します。Java Agent セットアップ・モジュールは、次のファイルを実行することで、いつでも起動できます。

```
< java_agent_install_dir > /DiagnosticsAgent/bin/setupModule.sh
```

< java_agent_install_dir > は、Java Agent のインストール・ディレクトリのパスです。標準設定のパスは /opt/MercuryDiagnostics/JavaAgent です。

UNIX 上でグラフィック・モードで Java Agent を設定する

グラフィック・モードで Java Agent を Java Sensor として機能するように設定するには

- 1 グラフィカル・インタフェースをセットアップします。

ディスプレイをターミナルにエクスポートします。

```
xhost + # コンソールに UI を表示できる
```

```
export DISPLAY= <ホスト名> :0.0
```

- 2 Java Agent セットアップ・モジュールを実行します。

```
< java_agent_install_dir > /DiagnosticsAgent/bin/setupModule.sh
```

Java Agent セットアップ・モジュールでは、151 ページ「Windows 上で TransactionVision Java Sensor として機能するように Java Agent を設定する」のように、Windows Java Agent セットアップ・モジュールのものと同じ画面が表示されます。

UNIX 上でコンソール・モードで Java Agent を設定する

コンソール・モードで Java Agent を Java Sensor として機能するように設定するには

- 3 [TransactionVision Java Agent working with an HP TransactionVision Server] オプションに対して **X** を入力し、このオプションを選択します。

```

*****
INFORMATION-> [ようこそ]: Java エージェント セットアップ モジュール (JASM) へようこそ
INFORMATION-> [初期化]: 初期化しています...
INFORMATION-> [初期化]: 初期化しています... 完了しました
INFORMATION-> [オプション]:
全ての入力プロンプトで次を入力できます:
'Enter' で標準設定値、
'-' で 1 つ前のステップ、
'+' で次のステップ、
'c' でキャンセル、または
'f' で終了
*****
HP Diagnostics/TransactionVision Agent for Java
Focus: Java エージェント設定オプションを選択
Progress: 1 of 10
*****
PLEASE INPUT (X:Yes, 0:No)-> Diagnostics Profiler のみ [X]:0
PLEASE INPUT (X:Yes, 0:No)-> HP Diagnostics Server と動作する Diagnostics Java エージェント [0]:0
PLEASE INPUT (X:Yes, 0:No)-> HP TransactionVision Server と動作する TransactionVision Java エージェント [0]:x

```

- ▶ **O** (大文字の O) を入力して [Diagnostics Profiler のみ] オプションをスキップし、もう 1 回 **O** を入力して [HP Diagnostics Server と動作する Diagnostics Java エージェント] オプションをスキップします。
- ▶ Java Agent を、Diagnostics Server と連携する J2EE Probe と、TransactionVision Java Sensor の両方として機能するように設定するには、両方のオプションに **O** を入力して手順 2 に進みます。

Enter キーを押して続行します。

4 Java Agent に名前を割り当て、それが属するグループを指定します。

```

*****
HP Diagnostics/TransactionVision Agent for Java
Focus: Java エージェントの識別
Progress: 2 of 4
*****
PLEASE INPUT-> Java エージェント名 [testAgent]:TestAgentForTV
PLEASE INPUT-> Java エージェント グループ [Default]:

```

- ▶ Java Agent 名には、TransactionVision 内で Agent を一意に識別する名前を入力します。名前に使用できる文字は、-, _ およびすべての英数字です。Agent 名が Java Sensor 名になるように割り当てられます。

Agent に名前を割り当てる際は、Agent が監視しているアプリケーションと Java Sensor の種類を識別しやすい名前を選択してください。

- ▶ Java Agent グループ名に対しては、既存のグループの名前か新規作成するグループの名前を入力します。Agent グループ名では大文字と小文字が区別されます。

Enter キーを押して続行します。

5 監視するアプリケーション・サーバを設定して、そのインストール・ディレクトリを入力します。通信リンク・トランスポートに使用する JMS ベンダを選択します。

```

PLEASE INPUT-> Java エージェント名 [sun-001-002-node-13]:TestTVAgent
PLEASE INPUT-> Java エージェント グループ [Default]:
*****
HP Diagnostics/TransactionVision Agent for Java
Focus: TransactionVision Java エージェントの設定 (1/2 ページ)
Progress: 3 of 4
*****
PLEASE MAKE SELECTION-> アプリケーション サーバの選択:
Selection 1. WebSphere
Selection 2. WebLogic
Selection 3. None
SELECT-> Please type in corresponding Number or
SELECT->   press Enter for default [None]:
1
PLEASE INPUT-> Please type in the installation path for AppServer WebSphere []:/opt/mqm
PLEASE MAKE SELECTION-> イベント トランスポート プロバイダを選択してください:
Selection 1. TransactionVision SonicMQ on Analyzer
Selection 2. WebSphere MQ
Selection 3. SonicMQ
Selection 4. TIBCO EMS
Selection 5. WebLogic JMS
SELECT-> Please type in corresponding Number or
SELECT->   press Enter for default [TransactionVision SonicMQ on Analyzer]:
2
PLEASE INPUT-> 設定キュー [TVISION.CONFIGURATION.QUEUE]:
PLEASE INPUT-> ユーザ名 (必要な場合) []:
PLEASE INPUT-> パスワード (必要な場合) []:*
*****

```

- ▶ 別のキューを使用する場合は、構成キューの名前を変更します。

- ▶ JMS プロバイダ用のユーザ名とパスワードを入力する必要がある場合は、それらを入力します。
- ▶ **Enter** キーを押して続行します。

6 どの JMS および JDBC システムを監視するかを設定します。

```

HP Diagnostics/TransactionVision Agent for Java
Focus: TransactionVision Java エージェントの設定 (2/2 ページ)
Progress: 4 of 5
*****
PLEASE INPUT-> 追加の TransactionVision プロパティ: [com.bristol.tvision.sensor.disableApp
s=TransactionVision,adminconsole,isclite,console,consoleapp,wl_management_internall,wl_man
agement_internall,uddi,uddiexplorer]:
PLEASE INPUT (X:Yes, 0:No)-> JMS 実装の監視対象: WebSphere MQ [0]:x
PLEASE INPUT-> JMS トランスポートのインストール パス WebSphere MQ [/opt/mqm]:
PLEASE INPUT (X:Yes, 0:No)-> JMS 実装の監視対象: SonicMQ [0]:
PLEASE INPUT (X:Yes, 0:No)-> JMS 実装の監視対象: TIBCO EMS [0]:
PLEASE INPUT (X:Yes, 0:No)-> JDBC 実装の監視対象: Oracle [0]:
PLEASE INPUT (X:Yes, 0:No)-> JDBC 実装の監視対象: IBM DB2 [0]:x
PLEASE INPUT-> JDBC トランスポートのインストール パス IBM DB2 [ ]:/usr/opt/db2
    
```

注 : [Transaction Agent Startup Properties] フィールドでは、Agent の操作とインストールメンテーションに関するディレクティブを指定します。通常は、HP カスタマ・サポートから別の指示がないかぎり、標準設定をそのまま受け入れます。プロパティ間はセミコロンで区切ります。

- ▶ **[追加の TransactionVision プロパティ]** プロンプトに対して、Enter キーを押します。
- ▶ 各 JMS トランスポートの横に大文字の X (Yes を意味する) または大文字の O (No を意味する) を入力して、1 つ以上の JMS トランスポートを選択します (もし望むなら、すべてのプロパティに対して No を入力することもできます)。
- ▶ いずれかのトランスポートについて Yes を指定した場合は、そのトランスポートのインストール・パスを入力する必要があります。
- ▶ 各 JDBC データベースの横に大文字の X (Yes を意味する) または大文字の O (No を意味する) を入力して、JDBC データベースを選択します。
- ▶ BEA JMS for WebLogic 8.1.x は自動的に監視されます。WebLogic アプリケーション・サーバに統合されるので、追加の設定は不要です。

7 JMS トランSPORTを設定します。

- ▶ 通信リンク・トランSPORTとして WebSphere MQ を選択した場合は、WebSphere MQ JMS サーバ・バインディングまたはクライアント接続を使ってトランSPORT設定を行います。

```

*****
HP Diagnostics/TransactionVision Agent for Java
Focus: WebSphere MQ の TransactionVision Java エージェント イベント トランSPORTの設定
Progress: 5 of 5
*****
PLEASE INPUT-> 接続キュー マネージャ [configuration_qm_name]:TRADING
PLEASE MAKE SELECTION-> 接続タイプを選択してください:
Selection 1. server
Selection 2. client
SELECT-> Please type in corresponding Number or
SELECT->   press Enter for default [server]:
2
PLEASE INPUT-> ホスト []:localhost
PLEASE INPUT-> ポート []:1421
PLEASE INPUT-> チャンネル []:TRADING.CHL
PLEASE INPUT-> インストール パス [/opt/mqm]:/opt/mqm
PLEASE INPUT->
CONFIRM-> 変更を保存しますか? Enter Y or N: [Y]:N

```

- ▶ 接続キュー マネージャの名前を入力し、**Enter** キーを押します。
- ▶ クライアント接続を選択した場合は、クライアントのキュー・マネージャ設定情報（ホスト、ポート、およびチャンネル）を入力します。
- ▶ WebSphere MQ のインストール場所を入力し、**Enter** を押します。
- ▶ 通信リンク・トランSPORTとして SonicMQ を選択した場合は、SonicMQ についてトランSPORT設定を行います。

```

*****
HP Diagnostics/TransactionVision Agent for Java
Focus: SonicMQ の TransactionVision Java エージェント イベント トランSPORT オプションの
設定
Progress: 5 of 5
*****
PLEASE MAKE SELECTION-> プロトコル タイプを選択してください:
Selection 1. tcp
SELECT-> Please type in corresponding Number or
SELECT->   press Enter for default [tcp]:
1
PLEASE INPUT-> ホスト [localhost]:
PLEASE INPUT-> ポート [2507]:
PLEASE INPUT-> インストール パス []:/opt/Sonic75/MQ7.5
CONFIRM-> 変更を保存しますか? Enter Y or N: [Y]:N

```

- ▶ プロトコルの名前を入力し、**Enter** キーを押します。
- ▶ ホスト名を入力し、**Enter** キーを押します。
- ▶ **Enter** キーを押して標準設定のポートを選択するか、別のポートを入力して **Enter** キーを押します。

- ▶ SonicMQ のインストール場所を入力し、**Enter** を押します。
- ▶ 通信リンク・トランスポートとして TIBCO EMS を選択した場合は、TIBCO EMS についてトランスポート設定を行います。

```

*****
HP Diagnostics/TransactionVision Agent for Java
Focus: Tibco EMS の TransactionVision Java エージェント イベント トランスポート オプション
の設定
Progress: 5 of 5
*****
PLEASE INPUT-> ホスト []:localhost
PLEASE INPUT-> ポート [7222]:
PLEASE INPUT-> インストール パス []:/opt/mqm
INFORMATION-> [既存の TIBCO EMS サーバ定義]:
PLEASE INPUT-> オプション? (A: 追加, Rn: 削除, En: 編集, または 0: 閉じる) [0]:A
PLEASE INPUT-> EMS サーバ名 []:ems_server
PLEASE INPUT-> EMS サーバのホスト []:localhost
PLEASE INPUT-> EMS サーバのポート番号 []:7222
PLEASE INPUT-> EMS サーバ管理者のユーザ名 []:Administrator
PLEASE INPUT-> EMS サーバ管理者のユーザパスワード []:*****
INFORMATION-> [既存の TIBCO EMS サーバ定義]:
1: ems_server
   URL: tcp://ems_server:7222/
   Username: Administrator
   Password: *****
PLEASE INPUT-> オプション? (A: 追加, Rn: 削除, En: 編集, または 0: 閉じる) [0]:

```

- ▶ ホスト名を入力し、**Enter** キーを押します。
- ▶ **Enter** キーを押して標準設定のポートを選択するか、別のポートを入力して **Enter** キーを押します。
- ▶ TIBCO EMS のインストール場所を入力します。
- ▶ EMS サーバの追加と定義ができます。
A を入力して **Enter** キーを押します。各サーバ定義を入力し、それぞれ
 の後で **Enter** キーを押します。
- ▶ 通信リンク・トランスポートとして WebLogic JMS を選択した場合は、
 WebLogic JMS についてトランスポート設定を行います。

```

*****
HP Diagnostics/TransactionVision Agent for Java
Focus: WebLogic 用の TransactionVision の Java エージェント イベント トランスポート オプシ
ョンの設定
Progress: 5 of 5
*****
PLEASE INPUT-> サーバ ホスト []:localhost
PLEASE INPUT-> サーバ ポート [7001]:7001
PLEASE INPUT-> キュー接続ファクトリ []:myQueueFactory
PLEASE INPUT-> インストール パス []:/opt/bee/weblogic81

```

- ▶ ホスト名を入力し、**Enter** キーを押します。

- ▶ **Enter** キーを押して標準設定のポートを選択するか、別のポートを入力して **Enter** キーを押します。
 - ▶ キュー接続ファクトリを入力し、**Enter** キーを押します。
 - ▶ WebLogic JMS のインストール場所を入力し、**Enter** を押します。通常、これは WebLogic アプリケーション・サーバのインストール場所と同じです。
- 8 Java Agent セットアップ・モジュールへの変更を保存するようにプロンプトが表示されたら、**Y**を入力します。
- 9 JRE をインストールメントします。

```

CONFIRM-> 変更を保存しますか? Enter Y or N: [Y]:y
INFORMATION-> [保存]: ダイアログ Java エージェント設定オプションを選択 を保存しています
INFORMATION-> [保存]: ダイアログ Java エージェントの識別 を保存しています
INFORMATION-> [保存]: ダイアログ Diagnostics Java エージェントの設定 を保存しています
INFORMATION-> [保存]: ダイアログ TransactionVision Java エージェントの設定 (1/2 ページ) を保存しています
INFORMATION-> [保存]: ダイアログ TransactionVision Java エージェントの設定 (2/2 ページ) を保存しています
INFORMATION-> [保存]: ダイアログ default を保存しています
INFORMATION-> [保存]: ダイアログ WebSphere MQ の TransactionVision Java エージェント イベント トランスポートの設定 を保存しています
INFORMATION-> [保存]: ダイアログ SonicMQ の TransactionVision Java エージェント イベント トランスポート オプションの設定 を保存しています
INFORMATION-> [保存]: ダイアログ Tibco EMS の TransactionVision Java エージェント イベント トランスポート オプションの設定 を保存しています
INFORMATION-> [保存]: ダイアログ WebLogic 用の TransactionVision のJava エージェント イベント トランスポート オプションの設定 を保存しています
INFORMATION-> [保存]: Diagnostics エージェントのプロパティ ファイルの保存
INFORMATION-> [保存]: TransactionVision エージェントのプロパティ ファイルの保存
INFORMATION-> [結合]: TransactionVision ルール ファイルを結合しています...お待ちください
INFORMATION-> [JASM ポスト セットアップのサマリ]:
成功 (/opt/mqm/java/lib) 選択したトランスポート Jar パスの検証
INFORMATION-> [現在測定されている VM]:
IBM 1.5.0 </usr/java5/jre>
IBM 1.4.2 </usr/java14/jre>
IBM 1.5.0 </usr/java/jre>
PLEASE INPUT-> オプション? ('コマンド', H: ヘルプ, または 0: 閉じる) [0]:

```

180 ページ「UNIX マシンでの JRE Instrumenter の実行」の説明に従って、インストールメンテーション・コマンドを入力します。

- 10 **0** (ゼロ) を入力してセットアップを終了します。

Agent のインストール、Java Probe としての設定、および JRE のインストールメントが終わったら、以下のインストール後の作業を実行する必要があります。

- 11 Probe が監視対象アプリケーションと連携して起動するようにアプリケーション・サーバの起動スクリプトを変更します。

詳しい手順については、183 ページ「アプリケーション・サーバの設定」を参照してください。

- 12 Java Agent のインストールを検証します。

Java Agent のサイレント・インストール

サイレント・インストールとは、ユーザが操作する必要がなく、自動的に実行されるインストールです。ユーザの入力の代わりに、サイレント・インストールでは、各インストール手順の応答ファイルからの入力を受け入れます。

たとえば、複数のマシンにコンポーネントをデプロイする必要があるシステム管理者は、必要な設定情報がすべて含まれる応答ファイルを作成し、複数のマシン上でサイレント・インストールを実行します。これにより、インストール中に手動で入力する必要がなくなります。

複数のマシン上でサイレント・インストールを実行する前に、入力をインストール中に提供する応答ファイルを作成する必要があります。この応答ファイルは、インストール時に同じ入力を必要とするすべてのサイレント・インストールで使用できます。

サイレント・インストールでは 2 つの応答ファイルを使用します。1 つは Java Agent のインストール用で、もう 1 つは Java Agent セットアップ・モジュール用です。

Java Agent のインストール用に応答ファイルを作成するには

次のコマンド・ライン・オプションを使って通常のインストールを実行します。

<インストーラ> -options-record <インストール用応答ファイル名>

これにより、インストール中に送信されたすべての情報が含まれる応答ファイルが作成されます。

Java Agent のセットアップ・プログラム用に応答ファイルを作成するには

次のコマンド・ライン・オプションを使って Java Agent セットアップ・プログラムを実行します。

▶ Windows の場合

```
< Java Agent のインストール・ディレクトリ > %bin%setupModule.cmd -
createBackups -console -recordFile < Java Agent セットアップ・モジュール用
応答ファイル名 >
```

▶ UNIX の場合

```
< Java Agent のインストール・ディレクトリ > /bin/setupModule.sh -
createBackups -console -recordFile < Java Agent セットアップ・モジュール用
応答ファイル名 >
```

どちらのコマンドでも、インストール中に送信されたすべての情報が含まれる応答ファイルが作成されます。

サイレント・インストールまたは設定を実行するには

対応する応答ファイルを使って、サイレント・インストールまたは設定を実行します。

環境変数を設定し、次のようにコマンド・ライン・オプション **-silent** を使用します。

```
set HP_JAVA_AGENT_SETUP=-DoNotRun <インストーラ> -options <イン
ストール用応答ファイル名> -silent
```

その後、次のコマンドを実行します。

```
set HP_JAVA_AGENT_SETUP=cd <セットアップ・モジュール> -
createBackups -console -installFile < Java Agent セットアップ・モジュール用
応答ファイル名 >
```

UNIX システムでは、"-DoNotRun" のように引用符で囲む必要があります。

JRE Instrumenter の実行

JRE Instrumenter は、アプリケーションで使用している JVM の ClassLoader クラスをインストールし、< **Java Agent のインストール・ディレクトリ** > /**DiagnosticsAgent/classes** ディレクトリの下フォルダにインストールされた ClassLoader を置きます。また、アプリケーション・サーバでインストールされたクラス・ローダを使用できるように、アプリケーションまたはアプリケーション・サーバが起動したときに使う必要のある JVM パラメータを提供します。

アプリケーション・サーバで使用される JDK (**java.exe** 実行可能ファイル) が変更された場合は、Java Agent で処理の監視を継続できるように、JRE Instrumenter を再度実行する必要があります。

注：

- ▶ IBM の 1.4.2 J9 JRE をインストールする場合、適切な ClassLoader をインストールして、アプリケーションのコマンド・ラインで **-Xj9** オプションを追加する必要があります。適切な ClassLoader は、< **Java ディレクトリ** > **¥jre¥lib¥jclSC14** ディレクトリ（たとえば **jreinstrumenter.sh -i ¥usr¥java14_64¥jre¥lib¥jclSC14**）にあります。
- ▶ Java Agent が複数の JVM の監視に使われている場合、各 JVM で実行中のアプリケーションを Java Agent がインストールできるようにするために、各 JVM に対して JRE Instrumenter を 1 回実行する必要があります。詳細については、『**HP Diagnostics インストールおよび設定ガイド**』の「**Configuring the Probes for Multiple Application Server JVM Instances**」を参照してください。

JRE Instrumenter の機能

JRE Instrumenter は、次の機能を実行します。

- ▶ インストール可能な JVM を特定する。
- ▶ 指定したディレクトリで、追加 JVM を検索する。
- ▶ 指定した JVM をインストールし、インストールされた ClassLoader クラスの場所を示すために JVM の起動スクリプトに追加する必要のあるパラメータを追加する。

JRE Instrumenter の手動での実行

以下は、Windows 環境とコンソール・モードの UNIX 環境で、JRE Instrumenter を実行するための手順です。

本項の内容

- ▶ 175 ページ「Windows マシンでの JRE Instrumenter の実行」
- ▶ 180 ページ「UNIX マシンでの JRE Instrumenter の実行」

Windows マシンでの JRE Instrumenter の実行

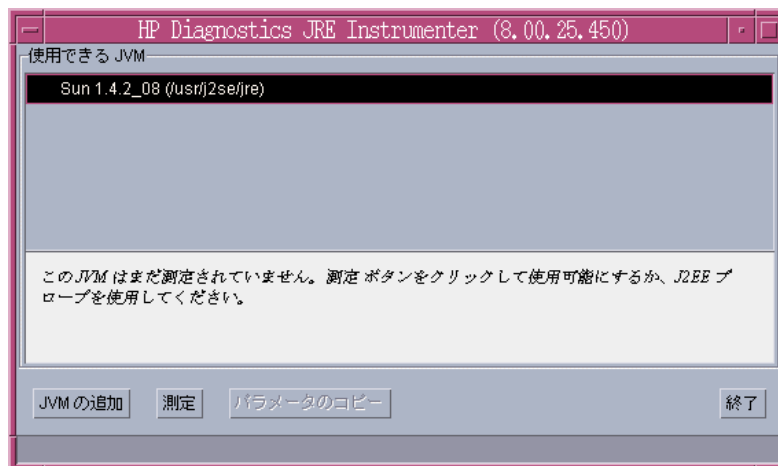
JRE Instrumenter を Windows 環境で実行すると、Instrumenter には、グラフィカル・ユーザ・インタフェースのダイアログが表示されます。Instrumenter をグラフィック・モードで実行している場合に、インストーラを UNIX マシンで実行したときも、同じダイアログが表示されます。

Windows マシンでの JRE Instrumenter の起動

- 1 < Java Agent のインストール・ディレクトリ > %DiagnosticsAgent%bin を開いて、JRE Instrumenter の実行可能ファイルを見つけます。
- 2 次のコマンドを実行します。

```
jreinstrumenter.cmd
```

Instrumenter が起動すると、[HP Diagnostics JRE Instrumenter] ダイアログが表示されます。



Instrumenter には、Instrumenter が検出した JVM が一覧表示され、インストールメンテーションに使用することができます。インストールされた JVM には、JVM の名前の前に緑色の四角形が付きます。

[HP Diagnostics JRE Instrumenter] ダイアログから、以下のタスクを実行できます。

- ▶ ダイアログの [使用できる JVM] リストにインストールする JVM がない場合は、176 ページ「使用可能な JVM リストへの JVM の追加」の説明に従ってリストに JVM を追加できます。
- ▶ リストにインストールする JVM があるものの、インストールされていない場合は、179 ページ「選択した JVM のインストール」の説明に従って JVM をインストールできます。
- ▶ リストにインストールする JVM があり、すでにインストールされている場合は、179 ページ「アプリケーション・サーバの起動スクリプトに JVM パラメータを含める」の説明に従って、JVM の起動スクリプトに挿入する必要のある JVM パラメータをコピーして、Probe の監視をアクティブにできます。
- ▶ JRE Instrumenter での作業が完了したら、[終了] をクリックして、JRE Instrumenter を閉じます。

使用可能な JVM リストへの JVM の追加

- 1 [JRE Instrumenter] ダイアログで、[JVM の追加] をクリックしてほかの JVM を検索し、[使用できる JVM] リストに追加します。

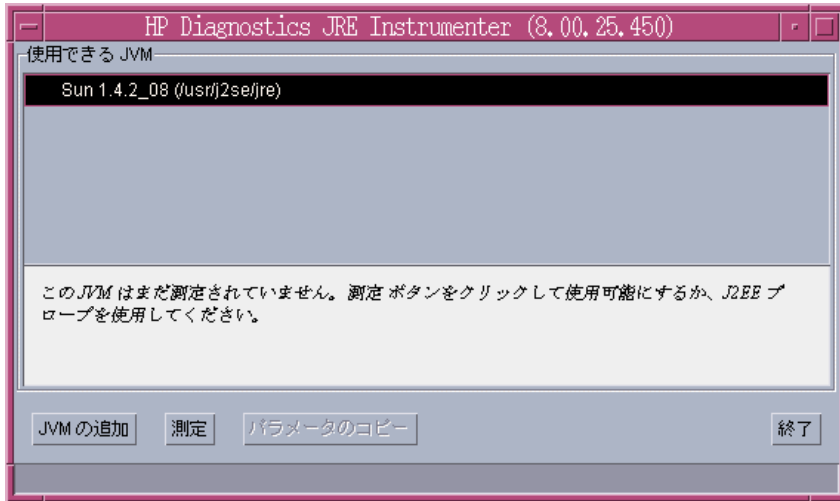
Instrumenter で [JVM があるディレクトリを選択してください] ダイアログ・ボックスが表示されます。



- 2 Instrumenter で JVM の検索を開始するディレクトリの場所を入力します。
- 3 [更新] をクリックすると、そのディレクトリの中にあるすべてのフォルダが [フォルダ] リストに表示されます。
- 4 [ファイル] ボックスに名前が表示されるように、検索を開始するフォルダを選択します。
- 5 [ここから検索] をクリックして、JVM の検索を開始します。

Instrumenter はダイアログ・ボックスを閉じ、もう一度 [JRE Instrumenter] ダイアログ・ボックスを表示します。ダイアログのコマンド・ボタンは無効になり、Instrumenter は JVM を検索します。ダイアログ下部の進行状況バーに、Instrumenter が検索中であることと、検索プロセスに要する時間が表示されます。

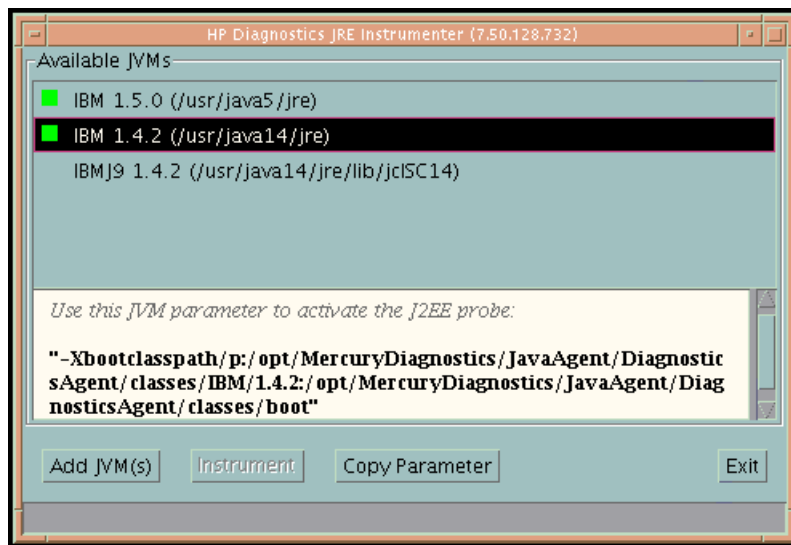
ツールが JVM を見つけると、その JVM が **「使用できる JVM」** リストに表示されます。



Instrumenter が検索を完了すると、ダイアログの Command ボタンが有効になります。選択した行がインストゥルメントされた JVM の場合は、[測定] ボタンが無効になります。緑色の四角形は、その JVM がインストゥルメントされたことを示します。

選択した JVM のインストールメント

[**使用できる JVM**] リストからインストールメントされていない JVM を選択し、[**測定**] をクリックします。



また、[**使用できる JVM**] リストの下のボックスに、アプリケーション・サーバの起動時に使用する必要のある JVM パラメータが表示されます。

アプリケーション・サーバの起動スクリプトに JVM パラメータを含める

JRE Instrumenter が JVM をインストールメントする際、アプリケーションでインストールメントしたクラス・ローダを使うために、アプリケーション・サーバの起動スクリプトに含めなければならない JVM パラメータも作成されます。[**使用できる JVM**] リストからインストールメントされた JVM を選択すると、リストの下のボックスに JVM パラメータが表示されます。インストールメントされた JVM をコピーして（一度に 1 つずつ）、アプリケーション・サーバにとって適切な場所に貼り付けることができます。

このボックスに表示された JVM パラメータをクリップボードにコピーするには、[**パラメータのコピー**] をクリックします。JVM パラメータがクリップボードにコピーされ、アプリケーション・サーバの起動時に取得できる場所に貼り付けることができます。

注：すべての JVM パラメータがコピーされてアプリケーション・サーバの適切な場所に貼り付けられたら、設定を有効にするために、アプリケーション・サーバを停止して再起動してください。コピーができなかった場合は、JVM パラメータを手動で入力してください。WebLogic JVM は、通常は **%bea_home%** ディレクトリにあります。WebSphere JVM は、通常は **%WebSphere_home%¥java** ディレクトリにあります。

UNIX マシンでの JRE Instrumenter の実行

ここでは、グラフィック・モード・インストールまたはコンソール・モード・インストールを使って、JRE Instrumenter を実行するのに必要な手順を紹介しません。

グラフィック・モードで表示される JRE Instrumenter の画面は、175 ページ「Windows マシンでの JRE Instrumenter の実行」に記載されている Windows インストーラで使用されている画面と同じです。

UNIX マシンでの JRE Instrumenter の起動

< **Java Agent のインストール・ディレクトリ** > **/DiagnosticsAgent/bin** を開いて、JRE Instrumenter の実行可能ファイルを見つけます。次のコマンドを実行します。

./jreinstrumenter.sh -console

Instrumenter が起動すると、次の表に示す使用可能な処理オプションのリストが表示されます。

Instrumenter の機能	説明
jreinstrumenter -l	既知の JVM のリストが表示されます。詳細については、181 ページ「インストールされた JVM のリストの表示」を参照してください。
jreinstrumenter -a DIR	DIR ディレクトリの下にある JVM を検索します。詳細については、181 ページ「使用可能な JVM リストへの JVM の追加」を参照してください。

Instrumenter の機能	説明
<code>jreinstrumenter -i JVM_DIR</code>	JVM_DIR 内の JVM をインストールメントします。詳細については、182 ページ「リストに表示された JVM のインストールメント」を参照してください。
<code>jreinstrumenter -b JVM_DIR</code>	JVM_DIR 内の JVM をインストールメントし、ClassLoader をく Java Agent のインストール・ディレクトリ > /DiagnosticsAgent/classes/boot に入れます。詳細については、182 ページ「リストに表示された JVM のインストールメント」を参照してください。

`jreinstrumenter.sh` コマンドを実行する際、`-x` オプションを指定して、オプションのリストを再表示することができます。

`./jreinstrumenter.sh -x`

インストールメントされた JVM のリストの表示

JRE Instrumenter の既知の JVM のリストを表示するには、次のコマンドを入力します。

`./jreinstrumenter.sh -l`

Instrumenter は、JVM ベンダ、JVM バージョンおよび JVM が置かれている場所を含め、行に JVM を一覧表示します。

使用可能な JVM リストへの JVM の追加

特定のディレクトリ内で JVM を検索し、見つかった JVM を JRE Instrumenter の既知の JVM のリストに追加するには、次のコマンドを入力します。

`./jreinstrumenter.sh -a DIR`

DIR には、Instrumenter で検索を開始する場所へのパスを指定します。

Instrumenter は、ディレクトリとサブディレクトリを含め、指定の場所からディレクトリを検索します。検索が完了すると、使用可能な JVM のリストが更新されて表示されます。

リストに表示された JVM のインストールメント

[使用可能な JVM] リストに表示された JVM をインストールメントするには、次のいずれかのコマンドを使用します。

- ▶ ClassLoader への特定のパス

`./jreinstrumenter.sh -i JVM_DIR`

JVM_DIR には、[使用可能な JVM] リストで指定した JVM の場所へのパスを指定します。

このコマンドは、JRE Instrumenter が、選択した JVM の ClassLoader クラスをインストールメントするように指示し、< **Java Agent のインストール・ディレクトリ** > /**DiagnosicsAgent/classes/** < **JVM のベンダ** > / < **JVM のバージョン** > ディレクトリの下フォルダにインストールメントされた ClassLoader を置きます。

このコマンドは、特に単一の Java Agent で監視する複数の JVM をインストールメントする場合に使用します。

- ▶ ClassLoader への一般的なパス

`./jreinstrumenter.sh -b JVM_DIR`

JVM_DIR を、[使用可能な JVM] リストで指定された JVM の場所へのパスと置き換えます。

このコマンドは、Java Agent を使って単一の JVM を監視しており、-i コマンド・オプションを使用するときに作成される明示的なパスを使いたくない何らかの理由があるときに使用します。

Instrumenter で JVM のインストールを終了すると、インストルメンテーションをアクティブにして、Java Agent によるアプリケーションの監視を有効にするために使用する必要がある JVM パラメータが表示されます。JVM パラメータに続いて、Instrumenter は、次の例のように [使用可能な JVM] リストを再表示します。

```

bash-2.04$ ./jreinstrumenter.sh -i /opt1/weblogic/boa/jdk150_01/jre
"-javaagent:/home/optibnch/clrtemp/JAUAProbe/lib/probeagent.jar"
HP 1.4.0.00 (</home/optibnch/clrtemp/JAUAProbe/_jvm)
HP 1.5.0.01 (</opt1/weblogic/boa/jdk150_01/jre)
HP 1.5.0.02 (</opt1/weblogic/boa/jdk150_02/jre)
HP 1.4.2.08 (</opt1/weblogic/boa/jdk142_08/jre)
HP 1.4.1.05 (</opt1/optibnch/DeepDiagnostics/jre)
HP 1.4.1.02 (</opt1/oracle/temp/Disk1/stage/Components/oracle.swd.jre/1.4.1.2.0/1
/DataFiles/Expanded)
HP 1.3.1.02 (</opt1/oracle/temp/Disk4/stage/Components/oracle.swd.jre/1.3.1.0.0c/
1/DataFiles/Expanded/jre/hpunix)
HP 1.4.1.05 (</opt1/oracle/OraHome1/jdk/jre)
HP 1.4.1.02 (</opt1/oracle/OraHome1/jre/1.4.1)
HP 1.4.1.05 (</opt1/DeepDiagnostics-1/jre)
bash-2.04$

```

アプリケーション・サーバの起動スクリプトへの JVM パラメータの追加

JRE Instrumenter が JVM をインストールする際、アプリケーションでインストールしたクラス・ローダを使うために、アプリケーション・サーバの起動スクリプトに含めなければならない JVM パラメータも作成されます。Instrumenter が JVM のインストールを完了すると、JVM パラメータが表示されます。

JVM パラメータをクリップボードにコピーし、アプリケーション・サーバの起動時に取得できる場所に貼り付けます。

アプリケーション・サーバの設定

Java Agent について JRE Instrumenter を実行した後、アプリケーションを監視する Java Agent がそのアプリケーションの起動時に立ち上がるように、アプリケーションの起動スクリプトを修正する必要があります。

アプリケーション・サーバの起動スクリプトを手動で更新して、アプリケーション・サーバを設定できます。次のセクションでは、アプリケーション・サーバを手動で更新する方法を説明します。

メッセージング・システム・プロバイダの設定

TransactionVision は、Analyzer と Sensor との通信にキューを使用します。少なくとも、構成キュー、イベント・キュー、例外キューの 3 つが必要です。これらのキューの標準設定の名前は、それぞれ

TVISION.CONFIGURATION.QUEUE, TVISION.EVENT.QUEUE, TVISION.EXCEPTION.QUEUE です。これらのキューは、使用しているメッセージ・システム・プロバイダで、ベンダ固有の方法でセットアップする必要があります。『**Using TransactionVision**』（英語版）の「Managing Communication Links」を参照してください。

IBM WebSphere MQ

IBM WebSphere MQ runmqsc ユーティリティ・プログラムか、Windows および Linux で使用できる MQ Explorer グラフィカル・ユーザ・インタフェースを使用して、キュー・マネージャ上にキューを作成できます。クライアント接続タイプを使用するためには、キュー・マネージャ上でサーバ接続チャンネルおよびリスナも定義する必要があります。詳細については、ベンダのドキュメントを参照してください。

TIBCO EMS

TIBCO EMS 管理ツールを使ってキューを作成できます。詳細については、ベンダのドキュメントを参照してください。

Progress SonicMQ

SonicMQ の管理コンソールを使って、SonicMQ Broker 上にキューを作成できます。詳細については、ベンダのドキュメントを参照してください。

BEA WebLogic JMS

WebLogic 管理コンソールを使って、JMS サーバとキューを設定できます。WebLogic 8.1 の場合に必要な手順の要約を以下に示します。詳細については、ベンダのドキュメントを参照してください。

- 1 一時テンプレートを作成します。
[サービス] > [JMS] > [テンプレート] を選択し、[新しい JMS テンプレートのコンフィグレーション] をクリックします。
- 2 永続的な JMS ストアを作成します。

[サービス] > [JMS] > [ストア] を選択し、**[新しい JMS ファイルストアの作成]** をクリックします。

ストアのディレクトリ・パス (/tmp のような) を指定する必要があります。
代わりに、JDBC ストアを使用することもできます。

3 JMS サーバを作成します。

[サービス] > [JMS] > [サーバ] を選択し、**[新しい JMS サーバのコンフィグレーション]** をクリックします。

このページで、上記で作成した永続的ストアと一時テンプレートを指定する必要があります。

4 キューを作成します。

[サービス] > [JMS] > [サーバ] を選択して**使用するサーバの名前**を選択し、**[送り先]** をクリックして **[送り先、新しい JMS キューのコンフィグレーション]** をクリックします。

5 接続ファクトリを作成します。

[サービス] > [JMS] > **[接続ファクトリ]** を選択し、**[新しい JMS 接続ファクトリのコンフィグレーション]** をクリックします。

カスタム・ユーザ・イベントの設定

イベントとして提示されるトランザクション・パスの一部として、カスタム・メソッドを含めることができます。これらのメソッドは標準設定では、標準 Java アプリケーション・フレームワークの一部（JMS、EJB、サーブレット、JSP など）でないかぎり、イベント履歴に含められません。

auto_detect.points ファイルを修正して、TransactionVision 専用ポイントを作成するか、details 行で tv:user_event タグを指定することによって既存の Diagnostics ポイントを TransactionVision ユーザ・イベントを指定するように修正します。

次に例を示します。

```
#
# GENERIC EVENT
#
[GenericEvent]
class    = !com.company.importantclass
method  = !.*
signature = !.*
detail   = tv:user_event
```

インストゥルメンテーション・ポイントの作成と既存のインストゥルメンテーション・ポイントの変更の方法については、『**HP Diagnostics インストールおよび設定ガイド**』のカスタム・インストゥルメンテーションの説明を参照してください。

第 16 章

Windows での WebSphere MQ および User Event Sensor のインストール

本章の内容

- ▶ Windows 上でのインストール・プログラムの起動 (187 ページ)
- ▶ 初めてのインストール (189 ページ)
- ▶ アップグレード・インストール (190 ページ)
- ▶ インストールの変更 (192 ページ)
- ▶ Sensor のアンインストール (194 ページ)

Windows 上でのインストール・プログラムの起動

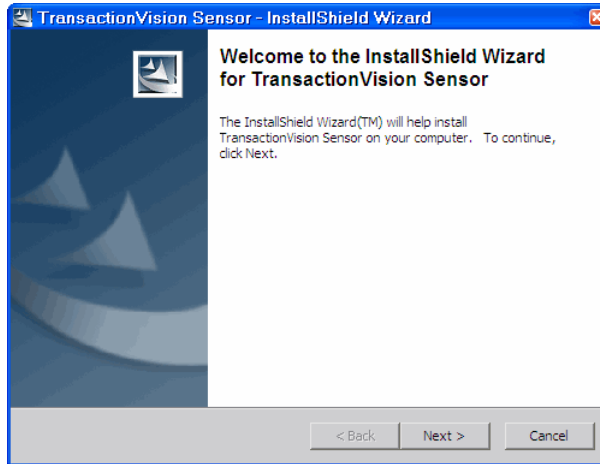
WebSphere MQ およびユーザ・イベント用の TransactionVision Sensor は、単一のパッケージとして Windows 上にインストールされます。本章では、これらの Sensor をインストールする手順について説明します (Java アプリケーションおよびアプリケーション・サーバ用の TransactionVision Sensor のインストールについては、第 15 章「Java Agent のインストールと設定」を参照してください)。

インストール先のシステムには、Administrator として、または Administrator 権限を持つユーザとしてログインする必要があります。

このパッケージをインストールするには、次の手順を実行します。

- 1 当該コンピュータで現在実行されている Windows プログラムを、自動バックアップ・プログラムも含めてすべて終了します。ウィルス対策、スパイウェア防止、および脅威対策プログラムを終了する必要はありません。

- 2 Windows エクスプローラで **tvwmque_800_win.exe** をダブルクリックします。
[InstallShield Welcome] 画面が開きます。



- 3 [Next>] をクリックすると, [InstallShield Save Files] 画面が表示されます。
- 4 インストール・ファイルを抽出するのに標準設定のフォルダを使用する場合は, [Next>] をクリックします。別のフォルダを選択する場合は, [Change] をクリックして希望するフォルダを選択し, [Next>] をクリックします。
InstallShield がインストール・ファイルを抽出します。

インストール先のコンピュータに TransactionVision Sensor を初めてインストールする場合は、「初めてのインストール」に進んでください。そのコンピュータに TransactionVision Sensor の以前のバージョンがインストールされている場合は、「アップグレード・インストール」に進んでください。

初めてのインストール

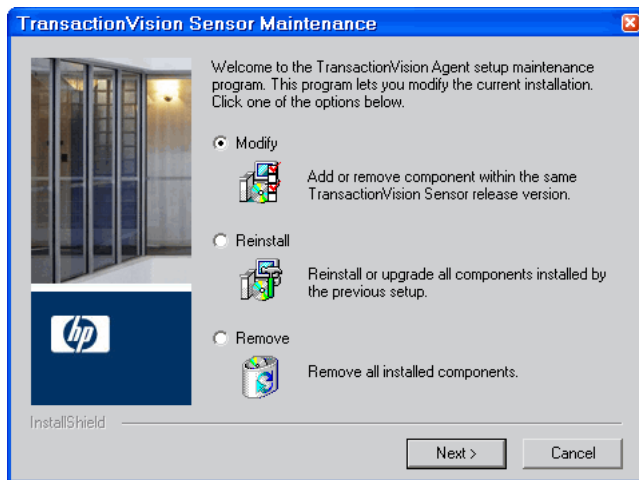
初めてのインストールの場合は、[Welcome] 画面が表示されます。

- 1 [Welcome] 画面で [Next] をクリックすると、TransactionVision のライセンス契約が表示されます。
- 2 [Yes] をクリックして、ライセンス契約に同意します。[User Information] 画面が表示されます。
- 3 自分の名前と会社名を入力し、[Next>] をクリックします。[Choose Destination Location] 画面が開きます。
- 4 WebSphere MQ およびユーザ・イベント用の Sensor をインストールするには、[Complete] を選択して [Next>] をクリックします。一部の Sensor だけをインストールする場合は、[Custom] を選択して [Next>] をクリックし、インストールする Sensor を選択して [Next>] をクリックします。

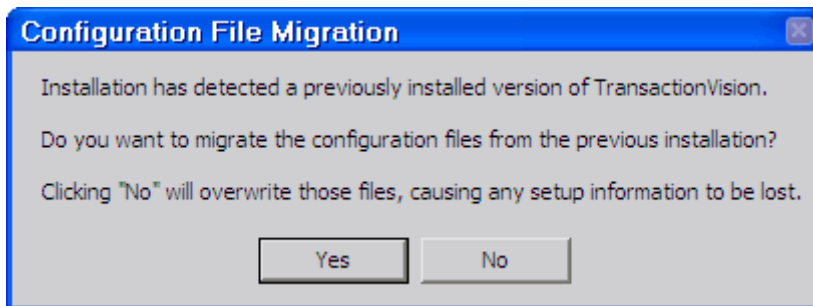
選択した Sensor が指定した場所にインストールされます。[Setup Complete] ページが表示されます。
- 5 [Finish] をクリックすると、インストールが完了します。

アップグレード・インストール

アップグレード・インストールの場合は、**twwmque_800_win.exe** をダブルクリックし、[InstallShield Welcome] 画面で **[Next>]** をクリックして、Sensor のセットアップ保守メニューを表示します。

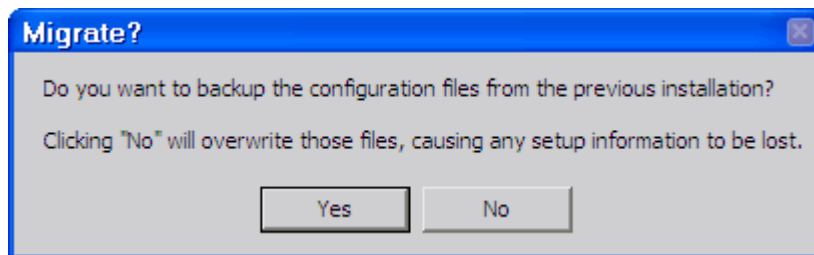


- 1 インストールされている TransactionVision をアップグレードするには、次のいずれかを選択します（インストールを変更する方法については、192 ページ「インストールの変更」を参照してください）。
 - ▶ 以前のインストールとは異なる設定で TransactionVision Sensor をインストールする場合は、以前のインストールを削除するために **[Remove]** を選択して **[Next>]** をクリックしてから、インストール手順を再び開始します。
 - ▶ 以前のリリースからアップグレードする場合は、**[Reinstall]** を選択して **[Next>]** をクリックし、以前のインストールからの設定を使って TransactionVision Sensor をインストールします。[Configuration File Migration] ダイアログ・ボックスが表示されます。



- 2 以前のインストールからの設定情報を保持する場合は、[Yes] をクリックします。インストール・ウィザードは、既存の設定ファイルのバックアップ・コピーを作成し、新しいバージョンの TransactionVision をインストールして、MS-DOS ウィンドウを開き、既存の設定ファイルを新しいバージョンに移行します。この作業が完了すると、[Setup Complete] 画面が表示されます。

既存の設定ファイルを上書きする場合は、[No] をクリックします。インストールを続行する前に既存の設定ファイルのバックアップ・コピーを作成するかどうかを尋ねるメッセージ・ボックスが表示されます。



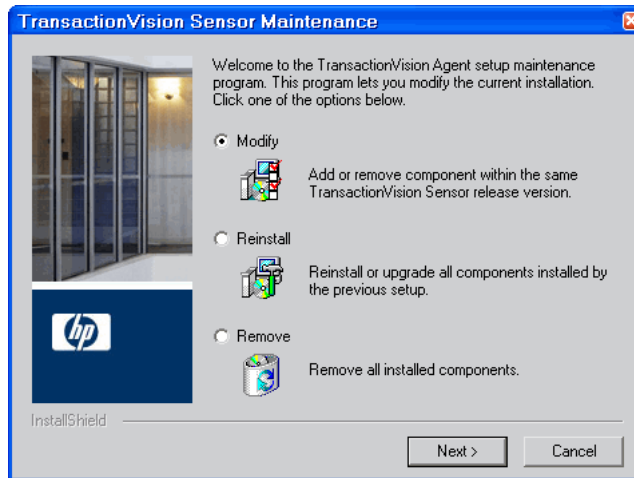
バックアップ・コピーを作成する場合は [Yes] をクリックし、設定ファイルをバックアップせずにインストールを続行する場合は [No] をクリックします。インストール・ウィザードが TransactionVision の新しいバージョンをインストールして既存の設定ファイルを上書きし、[Setup Complete] 画面を表示します。

- 3 インストール・ウィザードが TransactionVision の新しいバージョンをインストールし、[Setup Complete] 画面を表示します。
- 4 [Finish] をクリックすると、インストールが完了します。

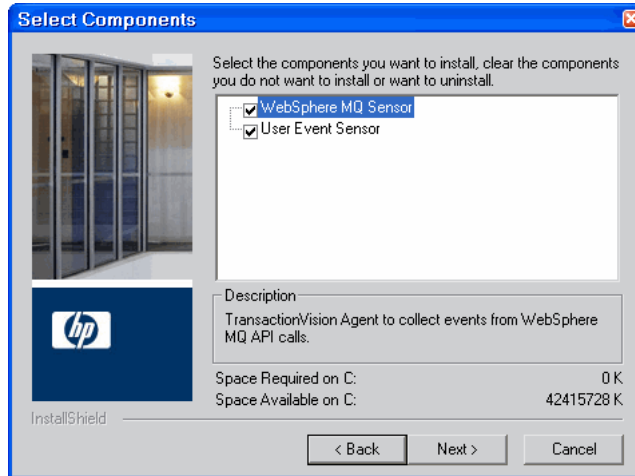
インストールの変更

TransactionVision Sensor をホストにインストールした後で、インストールの内容を変更する必要があることがあります。たとえば、最初に WebSphere MQ Sensor をホストにインストールした後、User Event Sensor をインストールしようと決めた場合などです。インストールを変更するには、次の手順を実行します。

- 1 **tvwmque_800.exe** をダブルクリックし、[InstallShield Welcome] 画面で [Next>] をクリックして、[TransactionVision Sensor Maintenance] 画面を表示します。



- 2 追加の Sensor をインストールするか、インストールされている Sensor を削除するには、**[Modify]** を選択して **[Next>]** をクリックし、**[Select Components]** 画面を表示します。



- 3 インストールする Sensor を選択して **[Next>]** をクリックし、インストールの変更を実行します。
- 4 **[Finish]** をクリックすると、インストールが完了します。

Sensor のアンインストール

TransactionVision コンポーネントをアンインストールするには、次の手順を実行します。

- 1 [スタート] メニューから [設定] > [コントロール パネル] を選択します。
- 2 [アプリケーションの追加と削除] をダブル・クリックします。
- 3 アンインストールする HP TransactionVision Sensor パッケージを選択し、[変更と削除] をクリックします。保守メニュー画面が表示されます。
- 4 [Remove] を選択して [Next>] をクリックし、TransactionVision コンポーネントを削除します。
- 5 指定したパッケージをアンインストールすることを確認し、[OK] をクリックします。指定したパッケージがアンインストールされます。次の種類のファイルは削除されません。
 - ▶ インストール後に追加されたファイル
 - ▶ まだインストールされているパッケージに関連付けられた共有ファイル
共有ファイルがインストールされたパッケージと関連付けられていると判断された場合（たとえば、ほかの TransactionVision パッケージがすべてアンインストールされた場合）は、[Locked File Detected] 画面が表示されます。
 - ▶ インストールされた共有ファイルをすべて残す場合は、[Don't display this message again] にチェック・マークを付けて [No] をクリックします。
 - ▶ 現在のファイルは残すけれど、ほかの共有ファイルについてはこのメッセージが表示されるようにする場合は、[No] をクリックします。
 - ▶ 共有ファイルを削除する場合は、[Yes] をクリックします。[Uninstallation Complete] 画面が表示されます。
- 6 [Next] をクリックしてアンインストール手順を完了します。

第 17 章

UNIX プラットフォームでの WebSphere MQ および User Event Sensor のインストール

本章では、UNIX プラットフォームで WebSphere MQ および ユーザ・イベント用の TransactionVision Sensor をインストールする手順について説明します。Java アプリケーションおよびアプリケーション・サーバ用の TransactionVision Sensor のインストールについては、第 15 章「Java Agent のインストールと設定」を参照してください。

TransactionVision は、各 Sensor ごとに固有のパッケージを提供します。

本章の内容

- ▶ Sensor のインストール (196 ページ)
- ▶ Sensor のアンインストール (199 ページ)

Sensor のインストール

インストール・ファイル

次の表は、WebSphere MQ Sensor のインストール・ファイルの名前を示しています。

プラットフォーム	ファイル
AIX	tvwmq_800_aix_power.tar.gz
HP-UX	tvwmq_800_hpux_parisc.tar.gz tvwmq_800_hpux_ia64.tar.gz
Linux	tvwmq_800_linux_x86.tar.gz
Solaris	tvwmq_800_sol_sparc.tar.gz
IBM i5/OS	tvwmq_800_i5os_iseries.savf

次の表は、User Event Sensor のインストール・ファイルの名前を示しています。User Event Sensor は共通パッケージに依存しているため、User Event Sensor をインストールすると共通パッケージが自動的にインストールされます。

プラットフォーム	ファイル
AIX	tvue_800_aix_power.tar.gz
Linux	tvue_800_linux_x86.tar.gz
Solaris	tvue_800_sol_sparc.tar.gz

インストールの手順

- 1 TransactionVision のインストール・ファイルがあるディレクトリの場所（DVD デバイスまたはダウンロード・ディレクトリ）に移動します。

注：Solaris と HP-UX では、DVD デバイスからホストのハード・ドライブ上の一時ファイルにインストール・ファイルをコピーしなければなりません。

- 2 スーパーユーザとしてログインします。

`su`

- 3 次のコマンドを入力して、インストール手順を開始します。

`./tvinstall_800_unix.sh`

次に示すメニューが表示されます。

The following TransactionVision packages are available for installation:

1. TransactionVision Analyzer
2. TransactionVision Web
3. TransactionVision WebSphere MQ Agent
4. TransactionVision User Event Agent

99. All of above

q. Quit install

Please specify your choices (separated by,) by number/letter:

注：実際のオプションと番号は、当該コンピュータ上で使用できるインストール・ファイルによって決まります。

- 4 単一のコンポーネントだけをインストールする場合は、その TransactionVision コンポーネント・パッケージに対応する番号を入力して **Enter** キーを押します。

複数の（すべてではない）コンポーネントをインストールする場合は、インストールする各コンポーネントに対応する番号をカンマで区切って入力し、**Enter** キーを押します。たとえば、上記のメニューからすべての **Sensor** をインストールする場合は、次のように入力して **Enter** キーを押します。

`3,4`

使用可能なコンポーネントをすべてインストールするには、**99** と入力して **Enter** キーを押します。

指定したパッケージがインストールされ、再びメニューが表示されます。

- 5 インストール手順を終了するには、**q** と入力して **Enter** キーを押します。追加のコンポーネントをインストールする場合は、それらのコンポーネントのインストール手順を参照してください。

AIX 上での WebSphere MQ Sensor の再バインド

AIX プラットフォーム上の WebSphere MQ Sensor の場合、インストール・プログラムは Sensor ライブラリをリンクしなおすために、**rebind_sensor** スクリプトを呼び出します。WebSphere MQ ライブラリ (**libmqm.a**, **libmqic.a**, **libmqm_r.a**, **libmqic_r.a**) に変更を加える WebSphere MQ サポート・パックをインストールした場合は、Sensor による監視の対象となるアプリケーションを正しく動作させるために、このスクリプトを再び実行する必要があります。このスクリプトの詳細については、付録 A 「ユーティリティ・リファレンス」を参照してください。

Sensor のアンインストール

TransactionVision コンポーネントをアンインストールするには、次の手順を実行します。

- 1 スーパーユーザとしてログインします。

```
su
```

- 2 次のコマンドを入力します。

```
./tvinstall_800_unix.sh -u
```

次のメニューが表示されます（実際のオプションは当該コンピュータにインストールされている TransactionVision パッケージによって異なります）。

```
The following TransactionVision packages are installed on the system:
```

1. TransactionVision Web
2. TransactionVision Analyzer
3. TransactionVision WebSphere MQ Agent
4. TransactionVision User Event Agent

```
99. All of above
```

```
q. Quit uninstall
```

```
Please specify your choices (separated by,) by number/letter:
```

- 3 アンインストールする TransactionVision パッケージに対応する番号を入力し、**Enter** キーを押します。

すべての TransactionVision コンポーネントをアンインストールするには、**99** と入力して **Enter** キーを押します。

指定したパッケージがアンインストールされ、再びメニューが表示されます。

- 4 アンインストールを終了するには、**q** と入力して **Enter** キーを押します。まだインストールされている TransactionVision パッケージが共通パッケージだけになると、共通パッケージも自動的にアンインストールされます。

第 18 章

i5/OS での Sensor のインストール

本章の内容

- ▶ i5/OS 上でのインストール・プログラムの起動 (201 ページ)

i5/OS 上でのインストール・プログラムの起動

Sensor を i5/OS プラットフォームにインストールするには、次の手順を実行します。

- 1 TransactionVision Sensor の以前のバージョンがインストールされている場合は、次のコマンドを実行してそのバージョンをアンインストールします。

```
DLTLICPGM LICPGM(3RBB9ES)
```

- 2 i5/OS マシンで、インストール・ファイルのコピー先として、既存のライブラリを見つけて使用するか、新規のライブラリを作成します (たとえば TVTMP)。
- 3 PC で、Sensor のインストール・ファイル tvwmq_800_i5os_ieries.savf を、CD_ROM から手順 1 で i5/OS マシン上に作成したライブラリに、名前を sensor800.savf に変更して FTP します。次のようにバイナリ・モード転送を設定してください。

```
ftp> binftp> cd /qsys.lib/tvtmp.lib ftp> put tvwmq_800_i5os_ieries.savf  
sensor800.savf
```

- 4 i5/OS マシンで、次のコマンドを実行して Sensor をインストールします。コマンド内の TVTMP は、sensor800.savf がパッケージが入っている実際のライブラリの名前と置き換えてください。

```
RSTLICPGM LICPGM(3RBB9ES) DEV(*SAVF)  
SAVF(TVTMP/SENSOR750)
```

5 次のコマンドを使ってインストールを検証します。

DSPSFWRSC

6 C Sensor を使用するには、プログラムを TVSENSOR/LIBMQM にバインドします。

7 手順 2 で新しい一時ライブラリを作成した場合は、この時点で安全に削除できます。

第 19 章

z/OS での Sensor のインストールと設定

本章の内容

- ▶ z/OS 環境での Sensor について (204 ページ)
- ▶ 基本コンポーネントのインストールの概要 (204 ページ)
- ▶ 基本コンポーネントのインストール手順 (205 ページ)
- ▶ z/OS での SLD Sensor コンポーネントの設定 : CICS, WebSphere MQ Batch, および WebSphere MQ IMS (212 ページ)
- ▶ z/OS での SLM Sensor コンポーネントの設定 : WebSphere MQ CICS Bridge および WebSphere MQ IMS Bridge (214 ページ)
- ▶ 背景情報 : CICS 用の WebSphere MQ Sensor (216 ページ)
- ▶ SLMC を CICS 用に設定する (217 ページ)
- ▶ 背景情報 : WebSphere MQ IMS Bridge Sensor (218 ページ)

z/OS 環境での Sensor について

IBM z/OS 環境へのデプロイメント用の TransactionVision Sensor は 2 種類あり、それぞれ固有の 3 文字の製品コードが付いています。

- ▶ SLD Sensor コンポーネントの監視対象は、CICS、WebSphere MQ (WMQ) Batch、および WMQ IMS イベントです。
- ▶ SLM Sensor コンポーネントの監視対象は、WMQ CICS Bridge および WMQ IMS Bridge イベントです。

本章では、基本コンポーネントのインストールとインストール後の設定作業の手順を、Sensor の種類別に説明します。基本インストールはどちらの種類でも事実上同じなので、1 回しか説明しません。Sensor コンポーネントの設定作業は大きく異なるので、別途説明します。

z/OS のその他の ESM (External System Requirements : 外部システム要件) については、付録 D 「その他の z/OS 設定」を参照してください。

基本コンポーネントのインストールの概要

- 1 インストール・メディアから zSeries プラットフォームにファイルを転送します。次の 2 つの方法があります。
 - ▶ インストール・メディアに含まれている転送支援用の **tvinstall_800_zos_zseries.bat** スクリプトを使用する (推奨)。
 - ▶ インストール・ファイルを手動で FTP する。
- 2 TSO RECEIVE コマンドを使って、転送されたファイルを受信して製品データ・セットを作成します。次の 2 つの方法があります。
 - ▶ 上の手順 1 で **tvinstall_800_zos_zseries.bat** によって作成された (TSORECV) Rexx スクリプトを使用して、TSO RECEIVE コマンドを実行する (推奨)。
 - ▶ 必要な TSO RECEIVE コマンドを手動で実行する。
- 3 環境にとって最良のインストール方法を決定します。次の 2 つの方法があります。
 - ▶ SMP/E (推奨)。新しい SMP/E グローバル・ゾーンを作成することも (推奨)、既存の SMP/E グローバル・ゾーンを使用することもできます。
 - ▶ 非 SMP/E

- 4 インストール・ジョブをカスタマイズして実行します。必要なカスタマイズの程度は、インストール方法（SMP/E または非 SMP/E）によって異なります。
- 5 インストール・ライブラリを確認します。

基本コンポーネントのインストール手順

手順 1：インストール・メディアから zSeries ホストにファイルを転送する
次の 2 つの方法があります。

- ▶ `tvinstall_800_zos_zseries.bat` スクリプトを実行する。
- ▶ インストール・ファイルを手動で FTP する。

ファイル名の表記規則

インストール・メディア上のファイル名は、以下のパターンに従っています。

`tv <製品コード><ファイル修飾子>_<バージョン>_zos_zseries.xmit`

ftp 出力データ・セット・ファイルの名前は、以下のパターンに従っています。

`< &custhlq > . <製品コード> | <バージョン> . <ファイル修飾子>`

説明：

<ファイル修飾子> = f1 | f2 | f3 | mcs

<製品コード> = SLD | SLM

<バージョン> = 800

< &custhlq > = ターゲット z/OS システム上の FTP PUT サブコマンドの出力に対する、カスタマが指定する高レベルのデータ・セット修飾子。

たとえば、インストール・メディア上のファイル名は次のようになります。

`tvslrdf1_800_zos_zseries.xmit`

ftp 出力データ・セット名は次のようになります。

`TVISION.SLD800.F1`

オプション 1 : tvinstall_800_zos_zseries.bat スクリプトを実行する。

- 1 Microsoft Windows が稼働していて TCP/IP FTP を使ってターゲット z/OS システムに接続できるワークステーションに、ログオンします。
- 2 MS-Windows のコマンド・プロンプト・セッションを開始します（[スタート] > [すべてのプログラム] > [アクセサリ] > [コマンド プロンプト]）。必要に応じてウィンドウのプロパティを調整し、そのサイズとバッファを増やします。
- 3 ディレクトリ変更 (cd) コマンドを使って、インストール・メディアがあるフォルダへ移動します。
- 4 次のように入力して、スクリプトが収集する情報を確認します。
`tvinstall_800_zos_zseries.bat /h | more`
- 5 パラメータなしで **tvinstall_800_zos_zseries.bat** を起動し、プロンプトに対して情報を入力します。

注：入力した値に満足できなかった場合は、ファイル転送を開始せずに `tvinstall_800_zos_zseries.bat` を終了することができます。また、**CTRL+C** キーを繰り返し押せば、いつでもこのスクリプトを終了できます。

- 6 **tvinstall_800_zos_zseries.bat** によって発行された FTP コマンド出力の結果を注意深く確認します。
 - ▶ エラーがあってはなりません。
 - ▶ ターゲット z/OS システムで指定された高レベル修飾子を持つインストール用 RELFILE が存在することを確認します。
 - ▶ 指定した PDS ライブラリの中に Rexx スクリプト (TSORECV) が存在することを確認します。
 - ▶ データ・セットが存在しない場合や、FTP エラーが検出された場合は、それらを調査して修正し、`tvinstall_800_zos_zseries.bat` プロセスを繰り返す必要があります。

オプション 2：インストール・ファイルを手動で FTP する。

tvshintall_800_zos_zseries.bat スクリプトを使ってファイルをインストールした場合は、この手順を実行しないでください。

- 1 Microsoft Windows が稼働していて TCP/IP FTP を使ってターゲット z/OS システムに接続できるワークステーションに、ログオンします。
- 2 MS-Windows のコマンド・プロンプト・セッションを開始します ([スタート] > [すべてのプログラム] > [アクセサリ] > [コマンド プロンプト])。必要に応じてウィンドウのプロパティを調整し、そのサイズとバッファを増やします。
- 3 ディレクトリ変更 (cd) コマンドを使って、インストール・メディアがあるフォルダへ移動します。
- 4 希望する高レベル修飾子を付けてデータ・セットを作成するのに十分な権限を持つ z/OS アカウントを使って、ターゲット z/OS システムとの FTP セッションを開始します。次に例を示します。

```
> ftp hostname
```

- 5 次の FTP サブコマンドを発行します。

```
a ftp> quote site fixrecfm 80 lrecl=80 recfm=fb blksize=3120 vol=&custvol
u=&custunit pri=30 sec=5 tr
```

&custvol は有効なディスク volser で、**&custunit** は有効なディスク・デバイス・タイプまたはエソテリックです。

```
b ftp> bin
```

```
c ftp> put tv<製品コード><ファイル修飾子>_<バージョン>_zos_zseries.xmit
'< &hlq >.<製品コード>< |バージョン>.<ファイル修飾子>'
```

説明：

<製品コード> = slm | sld

< &hlq > = カスタマが選択した高レベル修飾子

<ファイル修飾子> = f1 | f2 | f3 | mcs

<バージョン> = 800

次に例を示します。

```
put tvsldf1_800_zos_zseries.xmit 'TVISION.SLD800.F1'
```

- d ファイル名サフィックスが **fn** または **mcs** の、残りの製品インストール・ファイルについて、ftp **put** コマンドを発行します。

- e ftp コマンドの出力を注意深くチェックして、エラーがないか確認することが重要です。エラーを検出した場合は、z/OS システム上のターゲット・データ・セットを削除して ftp put コマンドを再実行しなければならないことがあります。

f ftp> quit

手順 2 : TSO RECEIVE コマンドを使って、転送されたファイルを受信して製品データ・セットを作成する

次の 2 つの方法があります。

- ▶ 205 ページ「手順 1 : インストール・メディアから zSeries ホストにファイルを転送する」で tvinstall_800_zos_zseries.bat によって作成された (TSORECV) Rexx スクリプトを実行する。
- ▶ 必要な TSO RECEIVE コマンドを手動で実行する。

オプション 1 : (TSORECV) Rexx スクリプトを実行する

TSO コマンド行または TSO/ISPF コマンド行から、205 ページ「手順 1 : インストール・メディアから zSeries ホストにファイルを転送する」で作成された (TSORECV) Rexx スクリプトを実行します。その結果、指定されたすべての z/OS ターゲット・データ・セットについて TSO RECEIVE コマンドが実行されます。

受信されたデータ・セットは、同じ高レベル修飾子を使って割り当てられますが、製品コードにプレフィックスとして文字 A が付け加えられます。たとえば、入力パターンが TVISION.SLD800.filequal の場合、出力パターンは TVISION.ASLD800.filequal のようになります。

オプション 2 : 必要な TSO RECEIVE コマンドを手動で実行する

TSO コマンド行または TSO/ISPF コマンド行から、必要に応じて前に TSO を付けて、次の表に示した RECEIVE コマンドを実行します。プロンプト「INMR906A Enter restore parameters or 'DELETE' or 'END」に対しては、**DSN(&custhlq.A <製品コード> 800.filequal)** と入力します。

コマンド	ファイル名
RECEIVE INDSNAME('&hlq.SL_800.F1')	DSN('&hlq.ASL_800.F1')
RECEIVE INDSNAME('&hlq.SL_800.F2')	DSN('&hlq.ASL_800.F2')

コマンド	ファイル名
RECEIVE INDSNAME('&hlq.SL_800.F3')	DSN('&hlq.ASL_800.F3')
RECEIVE INDSNAME('&hlq.SL_800.MCS')	DSN('&hlq.ASL_800.SMPMCS')

手順 3：最良のインストール方法を決定する

次の 2 つの方法があります。

- ▶ SMP/E (推奨)。新しい SMP/E グローバル・ゾーンを作成することも (推奨)、既存の SMP/E グローバル・ゾーンを使用することもできます。
- ▶ 非 SMP/E

手順 4：インストール・ジョブをカスタマイズして実行する

次の 2 つの方法があります。SMP/E または非 SMP/E

オプション 1：SMP/E インストール・ジョブ

表に示された順序でジョブをカスタマイズして実行します。ジョブは、データ・セット **&custhlq.A <製品コード> 800.F3** の中にあります。以下の表に示したジョブ名のアンダースコア文字 (_) を製品コードの 3 番目の文字で置き換えてから、現場の要件に合わせて個々のメンバをカスタマイズしてください。各ジョブの先頭にあるコメントを読み、それに従ってカスタマイズしてください。

実行が完了したら、次のジョブに移る前にすべての出力を注意深くチェックして、すべての手順が正常に完了したことを確認します。既存の SMP/E グローバル・ゾーンにインストールする場合は、SMP/E 手順、パラメータ、および入力を注意深くチェックしてください。

SMP/E インストール・ジョブ

順序	ジョブ	説明
1.	SL_ALLOC	ターゲット・ライブラリと配布ライブラリを割り当てます。
2.	SL_GZON	SMP/E グローバル・ゾーンを定義します (既存のグローバル・ゾーンを使用する場合は不要)。
3.	SL_DZON	SMP/E 配布ゾーンを定義します。
4.	SL_TZON	SMP/E ターゲット・ゾーンを定義します。

順序	ジョブ	説明
5.	SL_DDDEF	SMP/E DDDEF を定義します。
6.	SL_RECV	SMP/E Receive
7.	SL_APPLY	SMP/E Apply
8.	SL_IVP	インストールの検証
9.	SL_ACCPT	SMP/E Accept 重要 !SMP/E ACCEPT 機能がいったん実行されてしまうと、ターゲットまたは配布ライブラリから製品を削除できなくなるため、このジョブはインストールが設定されて完全にテストされた後のみ実行してください。

オプション 2 : 非 SMP/E インストール・ジョブ

非 SMP/E インストールを実行するには、以下のジョブを表に示された順序でカスタマイズして実行します。ジョブは、データ・セット **&custh1q.A** <製品コード> **800.F3** に含まれている場合があります。各ジョブの先頭にあるコメントを読み、それに従ってカスタマイズしてください。実行が完了したら、すべての出力を注意深くチェックして、すべての手順が正常に完了したことを確認します。

非 SMP/E インストール・ジョブ

順序	ジョブ	説明
1.	SL_INSTL	
2.	SL_IVP	インストールの検証

手順 5：インストール・ライブラリを確認する

DS#	データ・セット名	説明
1	TVISION.SSL_AUTH	Sensor ロード・モジュール (APF Auth)
2	TVISION.SSL_INST	Sensor インストール JCL のサンプル
3	TVISION.SSL_LOAD	Sensor ロード・モジュール
4	TVISION.SSL_PROC	Sensor サンプル JCL
5	TVISION.SL_800.F1	SMP/E JCLIN
6	TVISION.SL_800.F2	Inst RELFILE - ロード・モジュール
7	TVISION.SL_800.F3	Inst RELFILE - インストール JCL
8	TVISION.SL_800.SMPMCS	Inst RELFILE - SMP/E MCS ステートメント

- ▶ 上の表には、インストール用 SMP/E 配布ライブラリ (DLIBS) は含まれていません。SMP/E Accept 処理が実行されるまでは、これらのライブラリにデータは保存されません。
- ▶ 「TVISION」は標準設定の高レベル修飾子ですが、実際に使用されるものはこれとは異なる場合があります。

手順 6：基本コンポーネントのインストール手順（必要に応じて手順 3～5）を繰り返す

ライセンスを供与されているメインフレーム・センサ・コンポーネントによっては、基本コンポーネントのインストール手順の手順 3～5 を別の製品コード (SLM または SLD) を使って繰り返さなければならない場合があります。

z/OS での SLD Sensor コンポーネントの設定 : CICS, WebSphere MQ Batch, および WebSphere MQ IMS

第 20 章「CICS, WMQ Batch, および WMQ-IMS Bridge Sensor の設定」に記載されている, Sensor の操作, 動作, および設定の詳細情報も参照してください。この情報を読めば, TransactionVision Sensor の操作と, ほかのシステム・コンポーネントと Sensor との相互作用について, 理解が深まります。

z/OS で SLD Sensor コンポーネント (CICS, WebSphere MQ Batch, および WebSphere MQ IMS) を設定するには, 次の手順を実行します。

- 1 DS#2 (Sensor インストール JCL のサンプル) に含まれるジョブ SLDCICSD を実行して, CICS CSD ファイルを SLD Sensor に必要なリソース定義で更新します。環境内に複数の CICS 領域があり, それらが別々の CSD または異なる起動リストを使用している場合は, Sensor で監視される各 CICS 領域に対してこの手順を繰り返してください。
- 2 CICS Sensor プログラム・ロード・モジュールを, DFHRPL 連結内のライブラリに入れます。DFHRPL 連結内にある既存のライブラリにモジュールをコピーするか, SSLDLOAD ライブラリを DFHRPL 連結に追加します。別々の起動 JCL を使用する複数の CICS 領域を実行する場合は, Sensor で監視される各 CICS 領域に対してこの手順を繰り返してください。以下の CICS モジュールがあります。

SLDPCCX
SLDPCMIX
SLDPCPX
SLDPCSX
SLDPDSX
SLDPFCX
SLDPICX
SLDPPCX
SLDPPSX
SLDPTCX
SLDPTDX
SLDPTSX

- 3 この手順は省略可能です。Sensor の CICS 出口プログラムを CICS の起動時に自動的に有効にする場合は、SLDPCSX を 2 番目のパス PLTPI として定義します。『**CICS Resource Definition Guide**』（英語版）の DFHPLT に関する説明を参照してください。その定義を、既存の DFHPLTxx に追加します。新しい PLT が定義されたら、その定義への参照を CICS システム初期化テーブル (SIT) に追加します。『**CICS System Definition Guide**』（英語版）の「Specifying CICS system initialization parameters」を参照してください。Sensor で監視される各 CICS 領域に対して、この手順を繰り返します（この省略可能な手順を実行しない場合は、CICS トランザクション SLDS を実行して Sensor を有効にすることができます）。

SIT エントリの例

```
...
PLTPI=BI,
...
```

サフィックス・エントリを伴う PLT の定義の例

```
//DFHPLTPI EXEC DFHAUPLE
//ASSEM.SYSUT1 DD *
    DFHPLT TYPE=INITIAL,SUFFIX=BI
    DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
    DFHPLT TYPE=ENTRY,PROGRAM=SLDPCSX
    DFHPLT TYPE=FINAL
    END
//*
```

- 4 TransactionVision ライブラリ &hlq.SSLDAUTH を APF によって認証します。詳細については、『**MVS Initialization and Tuning Reference**』（英語版）の PROGxx または IEAAPFxx システム・パラメータの説明を参照してください（注：ライブラリ &hlq.SSLDLOAD は APF で認証してはなりません）
- 5 CICS Sensor Driver コンポーネントと TransactionVision Analyzer との通信に使用される WebSphere MQ キュー・マネージャ上に、適切な Sensor 設定およびイベント・キューを作成します。DS#2 (Sensor インストール JCL) 内のジョブ SLDCRTQS をカスタマイズして実行します。
- 6 Sensor 起動プロシージャのサンプル、つまり DS#4 (サンプル Sensor JCL) に含まれる TVISION および TVISIONC をカスタマイズして、現場にとって適切なプロシージャ・ライブラリにコピーします。起動プロシージャのキーワード・パラメータとそれらの意味の詳細については、第 20 章「CICS, WMQ Batch, および WMQ-IMS Bridge Sensor の設定」を参照してください。

z/OSでのSLM Sensorコンポーネントの設定：WebSphere MQ CICS Bridge および WebSphere MQ IMS Bridge

第20章「CICS, WMQ Batch, および WMQ-IMS Bridge Sensor の設定」には、Sensor の操作と動作に関する詳細情報が記載されています。この情報を読めば、TransactionVision Sensor の操作と、ほかのシステム・コンポーネントと Sensor との相互作用について、理解が深まります。

- ▶ 以下の設定手順の一部は、CICS 用の WebSphere MQ Sensor にのみ当てはまりません。そのような手順の先頭には「CICS のみ」と記されています。CICS 用の WebSphere MQ Sensor を使用しない場合は、それらの手順を省略してください。
- ▶ 以下の設定手順の一部は、IMS 用の WebSphere MQ Sensor にのみ当てはまりません。そのような手順の先頭には「IMS のみ」と記されています。IMS 用の WebSphere MQ Sensor を使用しない場合は、それらの手順を省略してください。
- ▶ 「CICS のみ」とも「IMS のみ」とも書かれていない手順は、両方の種類の Sensor に共通なので、CICS 用であるか IMS 用であるかに関係なく実行する必要があります。

第24章「WebSphere MQ Sensor の設定」に記載されている、Sensor のセットアップと操作に関する詳細情報も参照してください。

- 1 (CICS のみ) この設定の項の最後にある「背景情報：CICS 用の WebSphere MQ Sensor」を参照してください。
- 2 (CICS のみ) DS#2 (Sensor インストール JCL のサンプル) に含まれるジョブ SLMCICSD を実行して、CICS CSD ファイルを SLM Sensor に必要なリソース定義で更新します。環境内に複数の CICS 領域があり、それらが別々の CSD または異なる起動リストを使用している場合は、Sensor で監視される各 CICS 領域に対してこの手順を繰り返してください。

このジョブ・ステップは、CICS 領域内の CSQCAPX について、プログラム・リソース定義を作成します。その領域で、CSQCAPX に関するプログラム・リソースがすでに定義されていて、CSQCAPX プログラムがすでにインストールされている場合は、Sensor の CSQCAPX モジュールをインストールする前に、その古いプログラム・ロード・モジュールを DFHRPL 連結から削除する必要があります。これが必要な原因は、WebSphere MQ が CICS/WebSphere MQ 用の API 交差出口機能に使用する固定された命名メカニズムです（インストールできる交差出口は1つだけで、その名前は CSQCAPX でなければなりません）。

このジョブ・ステップは、SLMC のプログラム定義とトランザクション定義も作成します。

- 3 (CICS のみ) CSQCAPX および SLMC ロード・モジュールを、DFHRPL 連結内のライブラリに入れます。DFHRPL 連結内にある既存のライブラリにモジュールをコピーすることも、SSLMLOAD ライブラリを DFHRPL 連結に追加することもできます。
- 4 (CICS のみ) CKQC トランザクションを使用して、CICS 領域内の WebSphere MQ API 交差出口を有効にします。それには、CKQC パネルで [Connection] > [Modify] > [Enable API Exit] メニュー項目を選択するか、CKQC を起動するときに引数を指定します。たとえば、次のように入力します。
 - a CKQC MODIFY N E 交差出口を有効にします (E が有効を示す引数)
 - b CKQC MODIFY N D 交差出口を無効にします (D が無効を示す引数)
- 5 TransactionVision ライブラリ &hlq.SSLMAUTH を APF によって認証します。詳細については、『MVS Initialization and Tuning Reference』(英語版) の PROGxx または IEAAPFxx システム・パラメータの説明を参照してください (注: ライブラリ &hlq.SSLMLOAD は APF で認証してはなりません)。
- 6 Sensor Driver コンポーネントと TransactionVision Analyzer との通信に使用される WebSphere MQ キュー・マネージャ上に、適切な Sensor 設定およびイベント・キューを作成します。DS#4 (サンプル Sensor JCL) 内のジョブ SLMCRTQS をカスタマイズして実行します。

注: 以前に JOB SLDCATQS を実行している場合、この手順は必要ありません。

- 7 (CICS のみ) SLMC を CICS 用に設定します。この手順は必須ではありませんが、TransactionVision コンポーネントが CICS 領域を監視していないときに、その領域で実行される WMQ アプリケーション・プログラムのパフォーマンスを大きく向上させられるので、実行することを強くお勧めします。詳細については、本章の最後にある「SLMC を CICS 用に設定する」の項を参照してください。
- 8 (IMS のみ) この設定の項の最後にある「背景情報: WebSphere MQ IMS Bridge Sensor」を参照してください。
- 9 (IMS のみ) TVISIONB は、1 つのシステム・リンク・インデックス (LX) を必要とし、IPL 後に最初に起動されたときに LX を予約して、その後はその LX を再利用します。IBM z/OS または z/OS MVS の『Initialization and Tuning Reference』(英語版) に記載されている、IEASYSxx (システム・パラメータ・リスト) NSYSLX=nnn パラメータの説明を参照してください。

背景情報 : CICS 用の WebSphere MQ Sensor

z/OS CICS 用の WebSphere MQ Sensor は、2 つのコンポーネント・プログラムで構成されています。アプリケーションの WebSphere MQ API 呼び出しを監視するプログラムは、CSQCAPX という名前の WebSphere MQ API 交差出口プログラムです。Sensor がイベントを収集できるようにするために、このプログラムを CICS 領域にインストールする必要があります。2 つ目のプログラムである SLMC (同名の CICS トランザクションが関連付けられています) は、オプションです。このプログラムを実行すると、Sensor が必要としたときに CICS 領域内の交差出口 Sensor プログラムを自動的に有効または無効にすることができます。Sensor が正しく動作するために SLMC の実行は必要ありませんが、SLMC を実行すれば、Analyzer が当該領域を監視していないときに、その領域の WebSphere MQ アプリケーションのパフォーマンスが向上します。SLMC を実行しないことにした場合は、CICS 領域内で必ず交差出口を有効にしておいてください。

CICS 用の WebSphere MQ API 交差出口プログラムが必要なのは、固定したプログラム名 CSQCAPX を使用するためです。したがって、1 つの CICS 領域に一度にインストールできる WebSphere MQ API 交差出口は 1 つだけです。この名前のプログラムがすでにターゲット CICS 領域にインストールされている場合は、Sensor をインストールする前に、既存の CSQCAPX ロード・モジュールを DFHRPL 連結から削除して、CSQCAPX に関する既存の CICS プログラム定義を削除するか無効にしてください。

z/OS CICS 用の WebSphere MQ Sensor は、z/OS 言語環境ランタイム・サポートを必要とします。Sensor をインストールする前に、ターゲット CICS 領域で言語環境 (LE) が有効になっていることを確認してください。Sensor を使用するためには、ターゲット CICS 領域が C 言語からコンパイルされた LE プログラムをサポートしている必要があります。LE のサポートを有効にする手順は、『**CICS System Definition Guide**』(英語版)(CICS TS のドキュメントの「Installing Application Programs」の章)で説明されています。完全な詳細については、使用しているバージョンの CICS のドキュメントを参照してください。

SLMC を CICS 用に設定する

SLMC は、Analyzer からのメッセージ用の TransactionVision の構成キューを監視し、キュー内のメッセージを調べて、Analyzer の必要に応じて WebSphere MQ 交差出口を自動的に有効または無効にします。Analyzer が当該の領域を積極的に監視していないときに（キュー内に構成メッセージがない場合や、構成メッセージが当該の CICS 領域またはホストに無関係の場合に）交差出口を無効にすると、アプリケーションのパフォーマンスが向上します。

SLMC を使用するためには、SLMC プログラムを実行する SLMC トランザクションを実行するように、CICS 領域を設定する必要があります。SLMC は WebSphere MQ の CSQCRST プログラムを使用しますが、このプログラムはターミナルが接続された状態で実行する必要があります。したがって、交差出口を有効または無効にするためには、SLMC もターミナルが接続された状態で実行する必要があります。操作を簡単にするために、シーケンシャル・ターミナルを使って SLMC を実行することをお勧めします。そのターミナルは、CICS 領域が立ち上がったときにシーケンシャル・ターミナルがバックグラウンドで自動的に起動されるようにしてください。ただし、必要であれば通常のターミナルから直接起動することもできます。

シーケンシャル・ターミナル定義は、ターミナル入力および出力を提供するために CICS が使用する入力および出力データ・セットを定義します。入力データ・セットには実行する CICS トランザクション（このケースでは SLMC）の名前が含まれており、出力データ・セットはそのトランザクションからターミナル出力を受け取ります。シーケンシャル・ターミナル定義を作成するには、DFHAUPLE を使って CICS ターミナル・リソース・マクロ定義をアセンブルし、アセンブルされたマクロが置かれる TCT の読み取りを有効にするために SIT 内の TCT パラメータを設定して、ターミナルの入力および出力データ・セットを定義するために CICS 領域の起動 JCL に DD 名を追加する必要があります。

SLMC を実行するためのシーケンシャル・ターミナルを作成するには、CICS SDFHSAMP メンバ DFHTCT5\$ および DFH\$TCTS の中で IBM が提供しているサンプルのシーケンシャル・ターミナル定義を使用してください。このサンプルの完全な詳細と、シーケンシャル・ターミナルをセットアップする方法については、使用しているバージョンの CICS の『**CICS System Definition Guide**』（英語版）と『**CICS Resource Definition Guide**』（英語版）を参照してください。

SLMC トランザクションをシーケンシャル・ターミナルとともに実行するためには、そのシーケンシャル・ターミナルの入力データ・セット（たとえば、IBM のシーケンシャル・ターミナルのサンプルでは CARDIN DDNAME）に次の 2 行が含まれていなければなりません。

```
SLMC¥  
CESF LOGOFF¥
```

注：「¥」は、CICS の標準のデータ終了文字です。EODI システム初期化パラメータを通じてデータ終了文字を再定義している場合は、代わりにそのデータ終了文字を指定してください。

背景情報：WebSphere MQ IMS Bridge Sensor

WebSphere MQ IMS Bridge Sensor は、クロス・メモリ・プログラム呼び出しを通じて互いに通信する次の 3 つのコンポーネントとして実装されます。

- ▶ OTMA 入出力編集ルーチン DFSYIOE0：IMS 制御領域内で実行されます。
- ▶ 制御関数 TVISIONB：独立したアドレス空間で開始済みタスク（started task）として実行されます。
- ▶ イベント・ディスパッチャ関数 TVISIOND：別の独立したアドレス空間で開始済みタスクとして実行されます。

Sensor のセットアップと WebSphere MQ IMS Bridge Sensor の操作の手順については、第 24 章「WebSphere MQ Sensor の設定」を参照してください。

第 20 章

CICS, WMQ Batch, および WMQ-IMS Bridge Sensor の設定

TransactionVision CICS Sensor をインストールした後、各 Sensor を設定して起動する必要があります。

本章の内容

- ▶ 概要 (220 ページ)
- ▶ 共通の Sensor コンポーネント (220 ページ)
- ▶ CICS Sensor コマンド (222 ページ)
- ▶ Sensor の操作 (229 ページ)
- ▶ バッファ・キューに関する注意点 (231 ページ)
- ▶ TransactionVision Manager 起動プロシージャ (232 ページ)
- ▶ Sensor ドライバ起動プロシージャ (233 ページ)

概要

TransactionVision の z/OS Sensor はすべて、共通の基本コンポーネント (TransactionVision Manager, 環境 / サブシステム固有の Sensor マネージャ, および Sensor 自体), WMQ Batch Sensor, および WMQ-IMS Sensor の組み合わせとして実装されます。

TransactionVision Manager (TVM)

TVM は、開始済みタスクとして実行される許可済みのプログラムです。TVM はほかのコンポーネントを制御し、すべての Sensor が使用する共通のルーチンと構造体を提供します。TVM の起動後、Sensor の開始、停止、クエリ、または制御を要求するには、TVM 開始済みタスクのジョブ名に対して MVS 変更コマンドを発行します。使用可能なコマンドの一覧は、222 ページ「CICS Sensor コマンド」に記載されています。TVM をシャットダウンするには、MVS 停止コマンドを発行します。その場合は、TVM の制御下にある Sensor もすべてシャットダウンされます。詳細については、232 ページ「TransactionVision Manager 起動プロシージャ」を参照してください。

共通の Sensor コンポーネント

Sensor マネージャ

Sensor マネージャには、WebSphere MQ Sensor 用と CICS Sensor 用の 2 つがありますが、両方とも汎用モデルに準拠しており、すべての Sensor にとって必要な共通機能を実行します。Sensor マネージャは、TVM に対する Sensor 起動コマンドの結果として起動され、TVM のサブタスクとして実行されます。起動パラメータは、Sensor のタイプと監視されるターゲット・アプリケーション・セットを特定します。Sensor マネージャのタイプは、MQBATCH, WMQ-IMS, または CICS です。

WMQ-IMS タイプと CICS タイプについては、監視される IMS または CICS システムの IMSID または CICS SYSID も指定する必要があります。監視される WMQ-IMS または CICS システムそれぞれに対して、別々の Sensor マネージャを起動する必要があります。

Sensor マネージャは、WebSphere MQ または CICS イベントを保存するためにバッファ・キュー・データ・スペースを作成し、Sensor ドライバ・コンポーネントを起動します。WMQ-IMS システムのすべての依存領域ですべてのアプリケーションが生成したイベントを、1 つの Sensor が処理します。

Sensor ドライバ

Sensor ドライバには、WebSphere MQ Sensor 用と CICS Sensor 用の 2 つがありますが、両方とも汎用モデルに準拠しており、すべての Sensor にとって必要な共通機能を実行します。Sensor ドライバは、専用のアドレス空間で開始済みタスクとして実行される未許可のプログラムです。

Sensor ドライバは、Sensor マネージャが起動したときに Sensor マネージャによって自動的に起動され、Sensor マネージャが終了したときに終了されます。Sensor 起動コマンドのパラメータは、ドライバの起動に使用されるプロシージャを指定します。詳細については、233 ページ「Sensor ドライバ起動プロシージャ」を参照してください。

Sensor ドライバは、WebSphere MQ を通じて TransactionVision Analyzer と通信します。ドライバは Analyzer からの構成メッセージへの応答の中で、どの種類のイベントをキャプチャするかと、どのフィルタ条件を適用するかを、Sensor に対して指示します。また、キャプチャされたイベントをフィルタ条件に従ってバッファ・キュー・データ・スペースから取得し、Analyzer に送信します。

Sensor イベント・キャプチャ・コンポーネント

各 Sensor は固有のイベント・キャプチャ・コンポーネントを備えており、そのコンポーネントはアプリケーション環境内で動作してトランザクション・アクティビティを傍受し、関連イベント・データをバッファ・キューに保存します。

WebSphere MQ Sensor

監視されるバッチ・プログラムおよび IMS プログラムは、TransactionVision スタブ・プログラムにリンクされなければなりません。これらのスタブ・プログラムは、次の表のように WebSphere MQ スタブと置き換わります。

WebSphere MQ スタブ	置き換わるスタブ	説明
CSQBSTUB	SLDPSTBB	バッチ
CSQBRSI	SLDPSTBB	バッチ RRS
CSQBRSTB	SLDPSTBB	バッチ RRS
CSQQSTUB	SLDPSTBB	IMS

CICS Sensor コマンド

TransactionVision Manager の起動と停止に使用されるコマンドは、標準の MVS 起動コマンドおよび停止コマンドです。

TransactionVision Manager (TVM) の起動

```
START procname[.jobname],TVID=tvid
```

説明：

- ▶ **procname** は、TVM を起動するための、カタログに登録されたプロシージャの名前です。
- ▶ **jobname** は、開始済みタスクに割り当てられる MVS ジョブ名です。このパラメータを指定しなかった場合は、プロシージャ名がジョブ名として使用されます。
- ▶ **tvid** は、TVM のこのインスタンスの一意のシステム ID です。4 文字かそれより少ない数の文字で構成されます。プロシージャの定義によっては、**tvid** が省略可能なオプションのパラメータになる場合があります。詳細については、232 ページ「TransactionVision Manager 起動プロシージャ」を参照してください。

例：

```
S TVISION, TVID=TV01
```

TransactionVision Manager コンポーネントが起動されます。次のメッセージが表示されます。

```
SLDS434I TVISION Bristol TransactionVision for z/OS – V5.0.0  
SLDS400I TVISION TransactionVision Manager startup in progress.  
SLDS401I TVISION TV01 TransactionVision Manager startup complete
```

TransactionVision Manager (TVM) の停止

```
STOP jobname
```

説明：

- ▶ **jobname** は、指定された MVS ジョブ名です。または、停止される TVM ジョブの起動コマンドで指定された MVS ジョブ名が標準設定として使用されます。

例：

```
P TVISION
```

TransactionVision Manager コンポーネントが停止されます。停止された TVM の中で開始された Sensor がある場合は、その Sensor も停止されます。次のメッセージが表示されます。

```
SLDS402I TVISION TV01 STOP command received.  
SLDS404I TVISION TV01 TransactionVision Manager termination in progress.  
SLDS405I TVISION TV01 TransactionVision Manager termination complete.
```

その TVM の中で Sensor が実行されていた場合は、その Sensor のシャットダウン・メッセージも表示されます。

注：上記以外の Sensor コマンドはすべて、標準 MVS 変更コマンドとして発行されます。ほとんどのコマンド名とオペランド名は、その名前を一意に識別できる最低限の文字数まで短縮できます。短縮された名前が一意でない場合は、アルファベット順で最初に当てはまるコマンドまたはオペランド名であるものと想定されます。ただし、一部の名前は将来のために予約されています。

Sensor の起動

```
MODIFY tvm_jobname,START sensortype [SYSID(sysid) | IMSID(imsid)]
[DRVRPROC(drvrproc)]
[QMGR(qmgr)] [CONFIGQ(configq)]
[QBLKSIZE(qblksize)] [MAXQBLKS(maxqblks)]
```

説明：

- ▶ **tvm_jobname** は、指定された MVS ジョブ名です。または、起動される Sensor を制御する TVM ジョブの起動コマンドで指定された MVS ジョブ名が標準設定として使用されます。
- ▶ **START** はコマンド名で、必須です。
- ▶ **sensortype** は Sensor のタイプで、必須です。**MQBATCH**, **MQIMS**, または **CICS** を指定します。
sensortype が **MQBATCH** の場合は、**IMSID** または **SYSID** は指定しないでください。
- ▶ **sysid** が必要なのは、**sensortype** が **CICS** の場合です。**sysid** には、監視される CICS 領域の **SYSID** を指定しなければなりません。
- ▶ **imsid** が必要なのは、**sensortype** が **MQIMS** の場合です。**imsid** には、監視される IMS システムの **IMSID** を指定しなければなりません。
- ▶ **drvproc** は省略可能です。**drvproc** は、Sensor ドライバを起動するカタログに登録されたプロシージャの名前を指定します。詳細については、233 ページ「Sensor ドライバ起動プロシージャ」を参照してください。標準設定の名前は **TVISIONC** です。
- ▶ **qmgr** は省略可能です。**qmgr** は、WebSphere MQ キュー・マネージャの名前を指定します。このキュー・マネージャを通じて、Sensor ドライバ・コンポーネントは TransactionVision Analyzer と通信します。標準設定の値は、Sensor ドライバ起動プロシージャの中でパラメータとして指定されます。このパラメータと Sensor ドライバ起動プロシージャとの関係の詳細については、233 ページ「Sensor ドライバ起動プロシージャ」を参照してください。
- ▶ **configq** は省略可能です。**configq** は、WebSphere MQ キューの名前を指定します。このキューから、Sensor ドライバ・コンポーネントは TransactionVision Analyzer からの構成メッセージを受け取ります。標準設定の値は、Sensor ドライバ起動プロシージャの中でパラメータとして指定されます。このパラメータと Sensor ドライバ起動プロシージャとの関係の詳細については、233 ページ「Sensor ドライバ起動プロシージャ」を参照してください。

- ▶ **qblksize** は省略可能です。**qblksize** は、バッファ・キュー・データ・スペース内の各ブロックのサイズ (MB 単位) を指定します。**qblksize** の最小値は **1**, 最大値は **100**, 標準設定値は **3** です。詳細については, 231 ページ「バッファ・キューに関する注意点」を参照してください。
- ▶ **maxqblks** は省略可能です。**maxqblks** は, Sensor がそのデータ・スペースに割り当てることができるバッファ・キュー・ブロックの最大数を指定します。各キュー・ブロックのサイズは, **qblksize** に指定された値か, **qblksize** の標準設定の値になります。**maxqblks** の最小値は **3**, 最大値は **2046**, 標準設定値は **5** です。詳細については, 231 ページ「バッファ・キューに関する注意点」を参照してください。

例 1:

```
F TVISION,S MQB DR(TVISIONM) QM(CSQ1)
CO(TVISION.CONFIGURATION.QUEUE)
```

WebSphere QM Batch Sensor が起動します。TVM の Sensor マネージャ・サブタスクが起動し, Sensor ドライバ・タスクが独立したアドレス空間で起動します。次のメッセージが表示されます。

```
SLDS400I TVISION MQBATCH TransactionVision sensor startup in progress.
SLDS401I TVISION MQBATCH TransactionVision sensor startup complete.
IEF403I TVISIONM - STARTED [... および、ジョブ開始時に発行されるその他 MVS
メッセージ]
+SLMS278I : TransactionVision Sensor: WMQ **** Sensor Driver startup completed.
QMGR=CSQ1, CONFIGQ=TVISION.CONFIGURATION.QUEUE.
```

例 2:

```
F TVISION,S MQI IMSID(IVP1) DR(TVISIONM) QM(CSQ1)
CO(TVISION.CONFIGURATION.QUEUE)
```

WebSphere QM IMS Sensor が起動します。TVM の Sensor マネージャ・サブタスクが起動し、Sensor ドライバ・タスクが独立したアドレス空間で起動します。次のメッセージが表示されます。

```
SLDS400I TVISION MQIMS IVP1 TransactionVision sensor startup in progress.
SLDS401I TVISION MQIMS IVP1 TransactionVision sensor startup complete.

IEF403I TVISIONM - STARTED [...and other MVS messages issued when starting a
job]

+SLMS278I : TransactionVision Sensor: WMQ IVP1 Sensor Driver startup completed.
QMGR=CSQ1, CONFIGQ=TVISION.CONFIGURATION.QUEUE.
```

CICS Sensor を起動するときに、次の TransactionVision エラー SLDS499S が発生する場合があります。

```
SLDS499S TVISION CICS xxxx TVISION system error. Diagnostic data follow:
SLDS499S 01 07050224 19472077 0000000C 00000034
SLDS499S 02 00E00080 00000000 0000000C 6C000611
```

3 番目の行の最後の 2 つの値が上のメッセージのとおりであれば、現在のシステムで使用可能な **SCOPE=COMMON** データ・スペースの数が不足していることがエラーの原因です。SYS1.PARMLIB の適切な **IEASYSnn** メンバ内の **MAXCAD** パラメータの値を大きくしてから、MVS イメージを IPL してください。MAXCAD パラメータの調整の詳細については、IBM の『**MVS Initialization and Tuning Reference**』（英語版）ドキュメントを参照してください。

Sensor の停止

```
MODIFY tvm_jobname,STOP sensortype [SYSID(sysid) | IMSID(imsid)]
```

説明 :

- ▶ **tvm_jobname** は、指定された MVS ジョブ名です。または、停止される Sensor を制御している TVM ジョブの起動コマンドで指定された MVS ジョブ名が標準設定として使用されます。

- ▶ **STOP** はコマンド名で、必須です。
- ▶ **sensortype** は Sensor のタイプで、必須です。 **MQBATCH**, **MQIMS**, または **CICS** を指定します。
 - ▶ **sensortype** が **MQBATCH** の場合は、 **SYSID** または **IMSID** は指定しないでください。
- ▶ **sysid** が必要なのは、 **sensortype** が **CICS** の場合です。 **sysid** には、当該の Sensor の Sensor 起動コマンドで指定されたものと同じ **SYSID** を指定しなければなりません。
- ▶ **imssid** が必要なのは、 **sensortype** が **MQIMS** の場合です。 **imssid** には、当該の Sensor の Sensor 起動コマンドで指定されたものと同じ **IMSID** を指定しなければなりません。

例 1 :

```
F TVISION,STO MQB
```

WebSphere MQ Batch Sensor マネージャおよび Sensor ドライバが停止されます。次のメッセージが表示されます。

```
SLDS404I TVISION MQBATCH TransactionVision sensor termination in progress.
SLDS443I TVISION MQBATCH sensor quiescing: 14 events in buffer queue.
...
SLDS445I TVISION MQBATCH Sensor quiesce completed.
SLDS448I TVISION MQBATCH Sensor statistics:
  Events in queue:          0
  Events collected         474
  Events dispatched       474
  Events_lost              0

+SLDS27BI : TransactionVision Sensor: MQBATCH Sensor Driver is ending
IEF404I TVISIONM - ENDED [... および、ジョブ終了時に発行されるその他 MVS メッセージ]
SLDS405I TVISION MQBATCH TransactionVision sensor termination complete.
```

例 2 :

```
F TVISION,STO MQI IMSID(IVP1)
```

WebSphere MQ Batch Sensor マネージャおよび Sensor ドライバが停止されます。
次のメッセージが表示されます。

```
SLDS404I TVISION MQIMS IVP1 TransactionVision sensor termination in progress.
SLDS443I TVISION MQIMS IVP1 sensor quiescing: 14 events in buffer queue.
...
SLDS445I TVISION MQIMS IVP1 Sensor quiesce completed.
SLDS448I TVISION MQIMS IVP1 Sensor statistics:
  Events in queue:          0
  Events collected         474
  Events dispatched       474
  Events_lost              0

+SLDS27BI : TransactionVision Sensor: MQIMS IVP1 Sensor Driver is ending
IEF404I TVISIONM - ENDED [...and other MVS messages issued when stopping a job]
SLDS405I TVISION MQIMS IVP1 TransactionVision sensor termination complete.
```

調査

```
MODIFY tvm_jobname,INQUIRE STATISTICS [sensortype] [SYSID(sysid) |
IMSID(imsid)]
```

説明 :

- ▶ **tvm_jobname** は、指定された MVS ジョブ名です。または、調査の対象となる TVM ジョブの起動コマンドで指定された MVS ジョブ名が標準設定として使用されます。
- ▶ **INQUIRE** はコマンド名で、必須です。
- ▶ **sensortype** は Sensor のタイプで、必須です。MQBATCH, MQIMS, または CICS を指定します。**sensortype** を省略すると、すべての Sensor タイプがクエリの対象となります。
- ▶ **imsid** または **SYSID** を指定して、調査の対象とする MQIMS または CICS Sensor を特定できます。IMSID と SYSID が省略された場合は、指定されたタイプのすべての Sensor がクエリの対象となります。

例 :

```
F TVISION,IST
```

統計情報調査コマンドは、以下の情報を表示します。

```
SLDS448I TVISION MQBATCH Sensor statistics:
Events in queue:          56
Events collected:        530
Events dispatched:      474
Events_lost:             0
SLDS448I TVISION MQIMS IVP1 Sensor statistics:
Events in queue:          175
Events collected:        1068
Events dispatched:      893
Events_lost:             0
```

Sensor の操作

- 1 どの Sensor よりも先に TransactionVision Manager を起動する必要があります。
- 2 TransactionVision Manager は、インスタンスを一意に識別する TVID パラメータを使って起動されます。同時に複数の TransactionVision Manager を起動できません。これは、たとえばテスト環境と実運用環境に適しています。ただし、同一環境内では、1つの TransactionVision Manager の制御下で複数の Sensor を実行することによって、リソースが最も効率的に使用され、操作が簡易化されます。
- 3 特定の IMSID または SYSID を使用できる Sensor は、一度に 1 つだけです。
- 4 TransactionVision Manager のすべてのインスタンスに対して IMSID/SYSID ルールが禁止されるため、どの TransactionVision Manager インスタンスが関わっているかに関係なく、すでに Sensor によって監視されている特定の IMS システムまたは CICS 領域を監視するために Sensor を起動することはできません。また、WebSphere MQ バッチ・ジョブを監視できる TransactionVision Manager は一度に 1 つだけです。
- 5 TVID の値は、どの Sensor の IMSID または SYSID の値とも同じであってはなりません。将来の考慮事項として、サイト内のすべての TransactionVision Manager TVID, CICS SYSID, IMS ID, およびすべての MVS サブシステム ID の間で、TVID を一意にすることを強くお勧めします。

- 6 TransactionVision Manager を停止すると、その制御下にある Sensor がすべて自動的に停止されます。
- 7 Sensor が停止すると、Sensor マネージャがそれに対応する Sensor アプリケーション環境コンポーネントに対して、それ以上データをキャプチャしないように指示し、Sensor ドライバがバッファ・キューからすべてのイベントを取得するまで静止モードに入ります。すべてのイベントが取得されると（それまで時間がかかることがあります）、Sensor ドライバが終了し、その後 Sensor マネージャが終了します。
- 8 TransactionVision Manager の終了は、その制御下にある Sensor が終了するまで完了しません。
- 9 TransactionVision Manager をキャンセルすると、その制御下にある Sensor がすべて即座に終了し、バッファ・キュー内に残っているイベントはすべて破棄されます。さらに、TransactionVision Manager をキャンセルすると、一部の共通ストレージ制御ブロックが残りますが、その結果、その後 TransactionVision WebSphere MQ スタブにバインドされたバッチ・アプリケーションまたは IMS アプリケーションの実行により最初に発せられた WebSphere MQ 呼び出しが、エラー・メッセージの表示を引き起こすこととなります。アプリケーション自体は、それ以外の影響は受けません。WebSphere MQ 呼び出しは、Sensor なしの場合と同様に通常どおりに進行します。同一ジョブ内のスタブに対するその後の呼び出しはすべて、エラー・メッセージを生成しません。Sensor は自分自身を無効にし、すべての呼び出しをそのまま WebSphere MQ に渡します。その後問題を修正せずにジョブを実行するか IMS 依存領域を再起動すると、再びアプリケーションによる最初の呼び出しでエラー・メッセージが表示されます。TransactionVision Manager を再起動すると、残っていた制御ブロックが消去され、Sensor でのエラーが発生しなくなります。代わりに、リセット・プロシージャを実行して、TransactionVision Manager を起動せずに残っている制御ブロックを消去することもできます。hlq.SSLDPROC ライブラリの TVISIONR メンバの中に、リセット・プロシージャのサンプルが用意されています。

バッファ・キューに関する注意点

アプリケーション環境における Sensor コンポーネントのオーバーヘッドを最小限に抑えるために、Sensor はキャプチャしたすべてのトランザクション・イベントを、バッファ・キューというデータ・スペースに保存します。各 Sensor は、それぞれ 1 つのデータ・スペースを排他的に使用します。

バッファ・キューは、一定サイズのいくつかのキュー・ブロックとして構成されます。キュー・ブロックのサイズと数は、Sensor 起動コマンドのパラメータとして指定されます。MAXQBLKS はキュー・ブロックの最大数を指定し、QBLKSIZE は各キュー・ブロックのサイズ (MB) を指定します。使用されるデータ・スペースの最大サイズ (MB) は、MAXQBLKS と QBLKSIZE の積であり、2 GB を超えてはなりません。標準設定の MAXQBLKS は 5 で、最小数は 3 です。標準設定の QBLKSIZE は 3 で、最小数は 1 です。したがって、最小データ・スペース・サイズの要件は 3 MB となりますが、このサイズでは小規模なテストにしか対応できません。

バッファ・キューの適正なサイズは、監視対象アプリケーション環境のトランザクション・スループット、収集されるイベントの数と種類とサイズ、Sensor ドライバのスループット、および Sensor マネージャ内のバッファ・キュー管理機能の応答性によって大きく異なります。これらの変数については後ほど詳しく説明しますが、推測にすぎない数値をパラメータとして使って面倒な計算を始める前に、試してみてエラーを吟味するというアプローチを取ることをお勧めします。起動時に多めの領域を割り当てると、余分のオーバーヘッドがない代わりに、イベントの作業負荷を正確に見積もれない可能性があります。

Sensor の起動時に、3 つのキュー・ブロックが割り当てられます。実行された TransactionVision アプリケーション環境コンポーネントは、最初のキュー・ブロックから順にイベントを保存していきます。同時に、Sensor ドライバは保存時と同じ順序で (FIFO) バッファ・キューからイベントを取得していきます。バッファ・キューの状態をチェックするために、バッファ・キュー管理機能が 100 分の 1 秒ごとに呼び出されます。現在イベントを受け取っているキュー・ブロックが最後の割り当て済みキュー・ブロックである場合、割り当て済みキュー・ブロックの総数が MAXQBLKS の指定値を超えていなければ、追加のキュー・ブロックが割り当てられます。入っていたイベントがすべて取得されたキュー・ブロックは解放され、割り当て済みブロックの総数が減少します。したがって、バッファ・キューのサイズはトラフィックの増減につれて伸び縮みします。また、使用可能な最大サイズに関係なく、最低限の 3 つのキュー・ブロックを超える分については必要なスペースしか使用されません。起動時に多めの領域を割り当てると、余分のオーバーヘッドがない代わりに、イベントの作業負荷を正確に見積もれない可能性があります。

TransactionVision Manager 起動プロシージャ

TransactionVision Manager (TVM) を起動するときには、TVM のすべての同時インスタンスの間で一意の TVID をその TVM に割り当てる必要があります。TVID は、単一のパラメータとしてメイン・プログラム SLDPTVM に渡されます。TVM の起動用に、次のサンプル・プロシージャ TVISION が提供されています。

```
//TVISION PROC TVID=TV01,TVHLQ=TVISION
//TVISION EXEC PGM=SLDPTVM,TIME=1440,PARM='&TVID'
//STEPLIB DD DISP=SHR,DSN=&TVHLQ..SSLDAUTH
//      PEND
```

TVM が必要とするパラメータは、4 文字から成る TVID の値 1 つです。この値は、TVM の同時に実行されるすべてのインスタンスの間で一意でなければなりません。TVID はまた、同時に実行されるどの Sensor に割り当てられる ID とも異ならなければなりません。将来の考慮事項として、サイト内のすべての TVM, CICS SYSID, IMSID, およびすべての MVS サブシステム ID の間で、TVID を一意にすることを強くお勧めします。

TVID はプロシージャ・キーワード・パラメータとして定義され、標準設定値は TV01 です。START TVISION オペレータ・コマンドは、TVID の値として TV01 を使って TVM を起動します。START TVISION,TVID=TV02 オペレータ・コマンドは、TVID の値として TV02 を使って TVM を起動します。この 2 つのコマンドが両方とも発行されると、TVM のインスタンスが 2 つ起動され、その 2 つとも同じジョブ名 (TVISION) を持つこととなります。

TVM を停止したり、TVM の制御下にある Sensor を起動および停止するためには、対象となる TVM のジョブ名を指定して STOP または MODIFY コマンドを発行する必要があります。複数の TVM が同じジョブ名で起動されていた場合、そのコマンドはすべての同名の TVM に対して実行されます。この問題を避けるために、START コマンドには一意のジョブ名を指定してください。TVID をジョブ名として使用することをお勧めします。次に例を示します。

```
START TVISION,JOBNAME=TV01,TVID=TV01
START TVISION,JOBNAME=TV02,TVID=TV02
```


Sensor ドライバ起動プロシージャ

Sensor 起動コマンドを TVM に対して発行すると、Sensor ドライバ・コンポーネントが独立したアドレス空間で起動されます。Sensor マネージャは、Sensor 起動コマンドで指定されたプロシージャを使って自動的に Sensor ドライバを起動します (224 ページ「Sensor の起動」を参照)。Sensor 起動コマンドからのパラメータである **SYSID**, **QMGR**, および **CONFIGQ** が、Sensor ドライバ・プロシージャに渡されます。

Sensor 起動コマンドの **DRVRPROC** パラメータは、Sensor ドライバを起動するために呼び出されるプロシージャを指定します。標準設定のプロシージャ名は、WebSphere MQ Sensor の場合は **TVISIONM** で、CICS Sensor の場合は **TVISIONC** です。次のようなサンプルが hlq,SSLDPROC ライブラリの中に用意されています。

```
//TVISIONM PROC SYSID=,QMGR=CSQ1,
//          CONFIGQ=TVISION.CONFIGURATION.QUEUE,
//          TVHLQ=TVISION,MQHLQ=CSQ520
//TVISIONM EXEC PGM=SLDPMDR,TIME=1440,
//          PARM='&SYSID &QMGR &CONFIGQ'
//STEPLIB DD DISP=SHR,DSN=&TVHLQ..SSLDLOAD
//          DD DISP=SHR,DSN=&MQHLQ..SCSQAUTH
//          PEND
//TVISIONC PROC SYSID=,QMGR=CSQ1,
//          CONFIGQ=TVISION.CONFIGURATION.QUEUE,
//          TVHLQ=TVISION,MQHLQ=CSQ520
//TVISIONC EXEC PGM=SLDPDCDR,TIME=1440,
//          PARM='&SYSID &QMGR &CONFIGQ'
//STEPLIB DD DISP=SHR,DSN=&TVHLQ..SSLDLOAD
//          DD DISP=SHR,DSN=&MQHLQ..SCSQAUTH
//          PEND
```

プロシージャ・キーワード・パラメータとして、**SYSID**, **QMGR**, および **CONFIGQ** が、標準設定値とともに定義されています。前述のように、このプロシージャは Sensor マネージャによって自動的に呼び出されます。このプロシージャをほかの手段で呼び出してはなりません。たとえば、**START TVISIONM** コマンドを発行してはなりません。

Sensor ドライバと TransactionVision Analyzer との間の通信リンクに応じて、このプロシージャをカスタマイズしたり、複数のプロシージャを作成することができます。

Websphere MQ キュー・マネージャ (**QMGR**) パラメータと構成キュー (**CONFIGQ**) パラメータは、どの Analyzer が Sensor ドライバと通信するかを決定します。サンプル・プロシージャ内の **QMGR** パラメータと **CONFIGQ** パラメータは、**Sensor 起動** コマンドで同じパラメータを指定することによってオーバーライドできます。**SYSID** パラメータは、**Sensor 起動** コマンドから供給される必要があります。**SYSID**, **QMGR**, および **CONFIGQ** パラメータは、スペースで区切られた位置指定パラメータとして Sensor ドライブのメイン・プログラム **SLDPMDR** または **SLDPCDR** に渡されます。位置指定パラメータなので、省略できるのは最後の **CONFIGQ** パラメータだけです。このパラメータを省略すると、標準設定の **TVISION.CONFIGURATION.QUEUE** が使用されます。これらのプロシージャをカスタマイズしたり独自の Sensor ドライバ・プロシージャを作成するときには、以下の規則に従ってください。

- ▶ 3 つのパラメータ **SYSID**, **QMGR**, および **CONFIGQ** を、常にすべて指定します。
- ▶ **SYSID** パラメータの標準設定値として、常に **NULL** を指定します。
- ▶ **QMGR** パラメータの標準設定値として、常に **NULL** 以外の値を指定します。
- ▶ 常に、少なくとも **SYSID** の値と **QMGR** の値を、**PARM** フィールドを使って **SLDPCDR** に渡します。

注 : Sensor ドライバ・プロシージャ内の **SYSID** パラメータは、**SYSID** または **IMSID** として総称的に使用されています。

第 21 章

BEA Tuxedo への Sensor のインストールと設定

本章の内容

- ▶ インストールの準備 (235 ページ)
- ▶ インストールの実行 (236 ページ)
- ▶ Tuxedo Sensor の再バインド (237 ページ)
- ▶ Sensor のアンインストール (237 ページ)

インストールの準備

Sensor のインストーラは、Business Availability Center 製品のインストール・ディスクまたは Business Availability Center のダウンロード・ページから起動します。

次の表は、サポートされるプラットフォーム別の BEA Tuxedo パッケージのインストール・ファイル名を示します。

プラットフォーム	ファイル
AIX	tvtux_800_aix_power.tar.gz
HP-UX	tvtux_800_hpux_parisc.tar.gz
Solaris	tvtux_800_sol_sparc.tar.gz

インストールの実行

- 1 ディレクトリの場所を TransactionVision インストール・ファイル (DVD デバイスまたはダウンロード・ディレクトリ) に変更します。注: Solaris および HP-UX の場合、代わりにインストール・ファイルを DVD デバイスからホストのハード・ドライブの一時ディレクトリにコピーする必要があります。

- 2 次のようにスーパーユーザとしてログインします。

```
su
```

- 3 パッケージをユーザのプラットフォームに展開して、`untar` します。235 ページ「インストールの準備」を参照してください。次に例を示します。

```
gunzip tvtux_800_hpux_parisc_tar.gz
```

- 4 次のコマンドを入力して、インストール手順を開始します。

```
./tvinstall_800_unix.sh
```

パッケージが展開されると、使用可能なコンポーネントを示すメニューが表示されます。次に例を示します。

The following TransactionVision packages are available for installation:

1. TransactionVision Analyzer
2. TransactionVision UI/Job Server
3. TransactionVision WebSphere MQ Agent
4. TransactionVision BEA Tuxedo Agent
5. TransactionVision User Event Agent

99. All of above

q. Quit install

Please specify your choices (separated by,) by number/letter:

- 5 単独のコンポーネントをインストールするには、TransactionVision コンポーネント・パッケージに関連する番号を入力して、**Enter** キーを押します。

複数の、ただしすべてではないコンポーネントをインストールするには、インストールするコンポーネントに関連する番号をカンマで区切って入力し、

Enter キーを押します。たとえば上記のメニューのすべての **Sensor** をインストールするには、次の番号を入力して、**Enter** キーを押します。

```
3,4
```

使用可能なコンポーネントをインストールするには、**99**を入力して、**Enter** キーを押します。

指定されたパッケージがインストール・スクリプトによってインストールされ、メニューが再表示されます。

- 6 インストール手順を終了するには、**q**を入力して、**Enter** キーを押します。追加のコンポーネントをインストールするには、そのコンポーネントのインストール手順に従ってください。

Tuxedo Sensor の再バインド

WMQ Sensor とは異なり、Tuxedo Sensor では、インストールにより再バインド・スクリプトは自動的に呼び出されません。

Sensor ライブラリを再リンクするには、**rebind_tux_sensor** スクリプトを実行します。このスクリプトの詳細については、付録 A 「ユーティリティ・リファレンス」を参照してください。

Sensor のアンインストール

ホスト上の Sensor またはその他の TransactionVision コンポーネントをアンインストールするには、次の手順を実行します。

- 1 次のようにスーパーユーザとしてログインします。

```
su
```

- 2 次のコマンドを入力します。

```
./tvinstall_800_unix.sh -u
```

次のメニューが表示されます（実際のオプションは、ユーザのコンピュータにインストールされた TransactionVision パッケージによって異なります）。

The following TransactionVision packages are available for installation:

1. TransactionVision Analyzer
2. TransactionVision UI/Job Server
3. TransactionVision WebSphere MQ Agent
4. TransactionVision BEA Tuxedo Agent
5. TransactionVision User Event Agent

99. All of above

q. Quit install

Please specify your choices (separated by,) by number/letter:

- 3 アンインストールする TransactionVision パッケージに関連する番号を入力して、**Enter** キーを押します。

すべての TransactionVision コンポーネントをアンインストールするには、**99** を入力して、**Enter** キーを押します。

指定されたパッケージがインストール・スクリプトによってアンインストールされ、メニューが再表示されます。

- 4 アンインストールを終了するには、**q** を入力して、**Enter** キーを押します。共通パッケージが、インストールされている TransactionVision パッケージのみの場合、このパッケージは自動的にアンインストールされます。

BEA Tuxedo Sensor の設定

TransactionVision を使用してアプリケーションに関するイベント・データを記録するには、事前に、標準 BEA Tuxedo Sensor ライブラリの代わりに Sensor ライブラリを読み込むようにアプリケーション環境を設定する必要があります。

必要な設定の内容は次のとおりです。

- ▶ Sensor ライブラリをロードするためのアプリケーション環境の設定
 - ▶ Analyzer Configuration Service URL の設定
- 以下は任意設定です。
- ▶ 機密データのフィルタ処理

Sensor ライブラリをロードするためのアプリケーション・ライブラリの設定

BEA Tuxedo Sensor ライブラリは、そのライブラリ検索パスを標準 BEA Tuxedo Sensor ライブラリ検索パスの前に含めることによって、実行時に動的にロードされます。標準 BEA Tuxedo Sensor ライブラリは、Sensor ライブラリによって動的にロードされます。

アプリケーションを実行する前に、監視対象のアプリケーションが実行されるコンピュータの環境変数設定に Sensor ライブラリのディレクトリの場所を追加します。

次の表は、32 ビット・アプリケーションを使用する場合に各プラットフォームに設定する適切な環境変数とディレクトリの場所を示します。標準 BEA Tuxedo Sensor ライブラリを含むパスが環境値に含まれる場合、TransactionVision を使用するためには、そのパスの前に Sensor エントリが必要です。

プラットフォーム	環境変数	標準のディレクトリ
Sun Solaris	LD_LIBRARY_PATH	/opt/HP/TransactionVision/lib
HP-UX	SHLIB_PATH	/opt/HP/TransactionVision/lib
IBM AIX	LIBPATH	/usr/lpp/HP/TransactionVision/lib

この表に示すすべてのディレクトリの場所は、標準の Sensor インストール先です。Sensor が標準以外の場所にインストールされている場合は、Sensor 実行ファイルのディレクトリの場所を指定してください。

64 ビット・アプリケーションを使用する場合は、次の表に示すライブラリ・パスを設定してください。

プラットフォーム	環境変数	標準のディレクトリ
Sun Solaris	LD_LIBRARY_PATH	/opt/HP/TransactionVision/lib64:/opt/mqmq/lib64
HP-UX	SHLIB_PATH	/opt/HP/TransactionVision/lib64:/opt/mqmq/lib64
IBM AIX	LIBPATH	/usr/lpp/HP/TransactionVision/lib64:/usr/lpp/mqmq/lib64

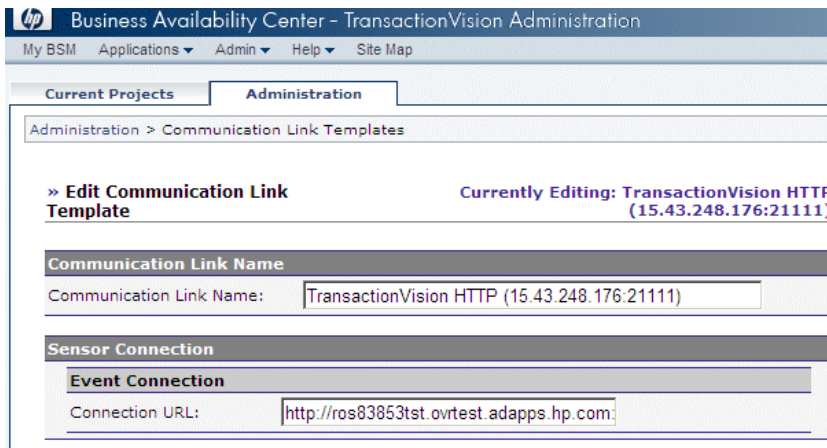
AIX プラットフォームでは、既存のライブラリと同名の新規ライブラリを取得する前に、`/usr/sbin/slibclean` を実行して元の共有ライブラリをメモリからクリアする必要があります。

HP-UX プラットフォームでは、`SHLIB_PATH` 環境変数が使用できないか、またはアプリケーションの動的ライブラリ検索パスの最初のパスでない場合は、`chatr` コマンドを使用して、アプリケーションが Sensor ライブラリを取得できるようにしておくことが必要な場合があります。`chatr` コマンドを使用して、アプリケーションの現在の設定をチェックすることができます。状況に関係なく、`SHLIB_PATH` が使用可能で、標準の BEA Tuxedo ライブラリ・パスの前にあることを確認してください。『[Using TransactionVision](#)』（英語版）の「[Troubleshooting](#)」を参照してください。

Sensor Connection URL の設定

BEA Tuxedo Sensor プロパティ・ファイル (`TVISION_HOME/config/TuxedoSensor.properties`) を変更し、`[static configuration]` セクションの `[transport]` オプションを Sensor Connection URL に設定する必要があります。

Sensor Connection URL は、通信リンク定義の一部として定義されます。この URL を取得するには、通信リンク・テンプレートにアクセスして `[Connection URL]` フィールドからコピーします。



機密データのフィルタ処理

BEA Tuxedo Sensor は、標準では監視対象の呼び出しからペイロード（ユーザ）データを収集するように設定されます。ユーザ・データが機密情報を含む場合、この機能を無効にできます。それには、BEA Tuxedo Sensor プロパティ・ファイル（`TVISION_HOME/config/TuxedoSensor.properties`）を変更し、`[static configuration]` セクションの `[collectUserData]` オプションを「no」に設定する必要があります。

第 22 章

.NET Agent のインストールと設定

.NET Agent は、Diagnostics .NET Probe と TransactionVision .NET Sensor の機能を組み合わせて 1 つのコンポーネントにしたものです。 .NET Agent は、 .NET ホスト上で TransactionVision Sensor および Diagnostics Probe として同時に機能します。

.NET Agent は、 HP ソフトウェアの Business Availability Center アプリケーションで機能する低オーバーヘッドのキャプチャ・ソリューションを提供します。 .NET Agent は、 .NET アプリケーションからイベントをキャプチャし、 TransactionVision Analyzer または Diagnostics Server あるいはその両方にイベント測定値を送信します。

本章の内容

- ▶ .NET Agent Installer について (244 ページ)
- ▶ .NET Agent のインストール (245 ページ)
- ▶ .NET Agent の設定 (250 ページ)
- ▶ IIS の再起動 (258 ページ)
- ▶ .NET Agent のバージョンを確認する (258 ページ)
- ▶ .NET Agent のアンインストール (259 ページ)
- ▶ .NET Agents 対応の SSL 設定 (259 ページ)

.NET Agent Installer について

.NET Agent のインストール時には、次のセットアップと設定が自動的に行われます。

- ▶ ASP.NET アプリケーションのディスカバリ。エージェントがインストールされているシステムで、インストーラにより ASP.NET アプリケーションの自動検索が試みられます。
- ▶ エージェントの標準設定。
 - ▶ インストーラは、検出された各 ASP.NET アプリケーションの ASP.NET/ADO/WCF 基本作業負荷をキャプチャするようにエージェントを設定します。エージェントの設定は、**probe_config.xml** ファイルを使用して制御されます。詳細については、250 ページ「.NET Agent の設定」を参照してください。
 - ▶ .NET アプリケーションの標準測定 `aspnet.points` および `adonet.points` の標準ポイント・ファイルがインストールされ、標準測定が有効化され、ユーザは ASP.NET アプリケーションを監視できるようになります。このポイント・ファイルによって、エージェントがアプリケーションについてキャプチャする作業負荷が制御されます。

標準ポイントである ASP.NET.points および Ado.Points は、TransactionVision イベントを生成できるように有効化されなければなりません。これらのポイントは標準設定で有効化されます。

.NET Remoting イベントを生成するには `Remoting.points` も必要なので、測定用のアプリケーションをセットアップする必要があります。`Remoting` のドキュメントを参照してください。

- ▶ 任意設定 `probe_config.xml` ファイルにエージェントの設定を修正します。詳細については、250 ページ「.NET Agent の設定」を参照してください。

.NET Agent のインストール

次の項で、.NET Agent のインストール方法について詳しく説明します。

本項の内容

- ▶ 245 ページ「インストールの準備」
- ▶ 246 ページ「インストーラの起動」
- ▶ 246 ページ「インストールの実行」
- ▶ 249 ページ「Agent が TransactionVision 環境で機能するようにインストールする」

インストールの準備

.NET Agent は、監視対象のアプリケーションのホスト・マシンにインストールします。監視されるシステムで .NET (.NET Probe) の Sensor によって生ずるオーバーヘッドは、極めて低いものです。

以下は、Sensor の処理をサポートするための推奨空きメモリ容量とディスク容量です。

プラットフォーム	すべてのプラットフォーム
メモリ	60 MB の追加 RAM
ハードディスクの空き容量	200 MB の追加容量
.NET Framework	1.1 以降

重要： .NET Agent には SOAP Extension Handler も含まれています。 .NET Agent をインストールすると、SOAP を使用する既存の Web アプリケーションの再起動が必要となる場合があります。

インストーラの起動

インストーラは、Business Availability Center 製品のインストール・ディスクまたは Business Availability Center のダウンロード・ページから起動します。

製品インストール・ディスクからインストーラを起動するには

- 1 プラットフォームのインストール・パッケージ・ファイルを検索します。

プラットフォーム	ファイル
32 ビット .NET	DotNetAgentSetup_x86_8_00.msi
64 ビット .NET	DotNetAgentSetup_x64_8_00.msi

- 2 インストール・ディスクのルート・ディレクトリにあるファイルを実行します。
246 ページ「インストールの実行」に進みます。

Download Center - HP - BTO Software ページからインストーラを起動するには

- 1 リソース・タイプ・ボックスから [Keyword] フィールドに **TransactionVision** を、[検索条件を絞る] フィールドに **Trial software** を入力します。
- 2 適切なリンクをクリックして、プラットフォームに .NET Agent インストーラをダウンロードします。
246 ページ「インストールの実行」に進みます。

インストールの実行

インストーラを起動したら、メイン・インストール手順を開始できます。

Windows マシンに .NET Agent をインストールするには

- 1 エンド・ユーザ使用許諾契約書に同意します。
使用許諾契約書を読み、[I accept the terms of the License Agreement] を選択します。
[Next] をクリックして先に進みます。

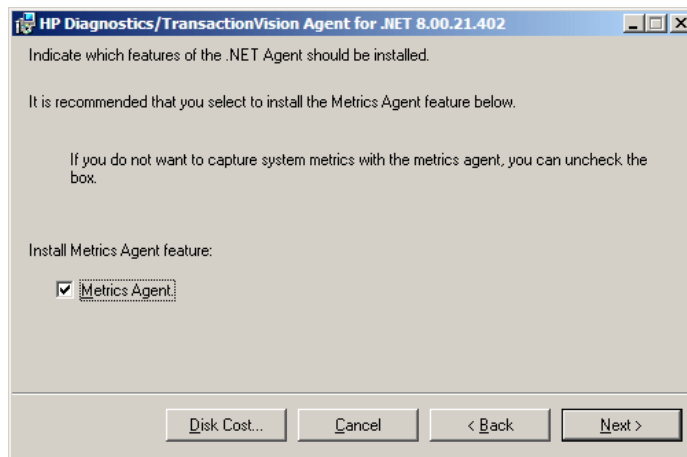
2 Agent のインストール先を指定します。

標準では、Agent は **C:\MercuryDiagnostics\NET Probe** にインストールされます。

標準のディレクトリを受け入れるか、**[Installation HP Diagnostics/TransactionVision Agent for .NET to]** ボックスにインストール・ディレクトリのパスを入力するか、または **[Browse]** をクリックしてインストール・ディレクトリを指定して、ほかの場所を選択します。

[Next] をクリックして次に進みます。

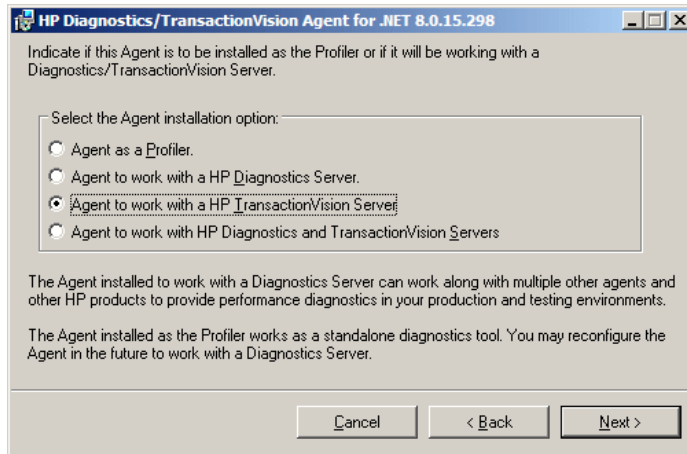
3 インストールする .NET Agent の機能を選択します。



アプリケーションを監視せずに、ホスト・マシンのシステム測定値をキャプチャする場合は、**[Metrics Agent]** だけを選択します。

ホストのドライブ上の空きディスク容量を確認するには、**[Disk Cost]** をクリックします。この機能を使って、Agent のインストールに十分な空き容量があることを確認します。

- 4 Agent を Diagnostics Profiler としてのみインストールするのか、または Diagnostics Server に報告するプローブとして、あるいは TransactionVision Analyzer へ報告する Agent としてインストールするのを選択します。



➤ [Agent to work with an HP TransactionVision Server] を選択します。

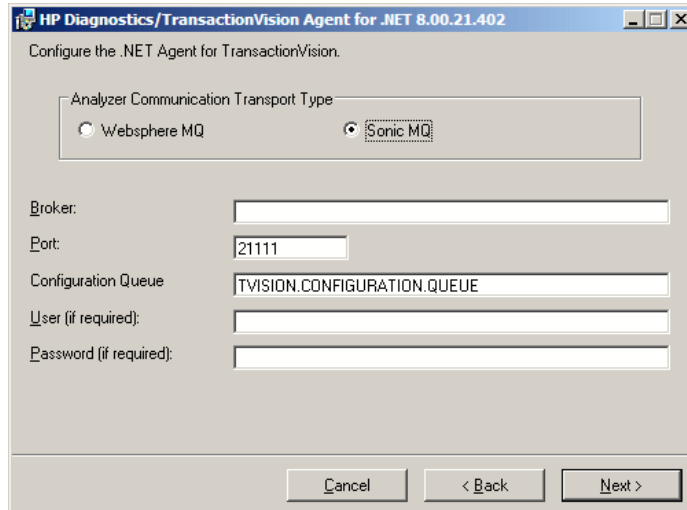
[Next] をクリックして次に進みます。

注： Agent を Diagnostics Profiler としてインストールするか、または Diagnostics Server を使用するようにインストールする場合、『**Diagnostics インストールおよび設定ガイド**』を参照してください。

Agent が TransactionVision 環境で機能するようにインストールする

Agent が TransactionVision 環境で機能するようにインストールする場合は、次の手順に進みます。

- 1 [Configure the .NET Agent for TransactionVision] ダイアログが表示されます。



- 2 Messaging Middleware Provider を選択します。次の選択肢があります。
WebSphere MQ と SonicMQ。

SonicMQ は、.NET Agent に含まれます。このオプションを指定すると、Sonic MQ .NET クライアント (Sonic.Client.dll - Progress SonicMQ .NET Client, バージョン 7.6.0.112) が Agent インストールの一部としてインストールされます。

あるいは、サードパーティの WebSphere MQ インストールを使用することもできます。この場合、監視するホストに MQ シリーズ .NET クライアント (amqmdnet.dll - WebSphere MQ Classes for .NET, バージョン 1.0.0.3) をインストールする必要があります。

標準設定では、SonicMQ が選択されます。

- 3 SonicMQ については、次のように入力します。

[Broker]。Sonic ブローカが実行されるホスト名。普通は Analyzer ホスト名です。

[Port]。ブローカが通信を行うポート。標準設定では 21111 です。

[Configuration Queue]。構成キューの名前。標準設定では TVISION.CONFIGURATION.QUEUE です。

[**User**]。SonicMQ インストールで接続に必要な場合は、ユーザ ID。標準設定では、ユーザ名は不要です。

[**Password**]。SonicMQ インストールで必要とされる場合は、パスワード。これは、PassGen ユーティリティを使用して作成される難読化形式になります。標準設定では、パスワードは不要です。**PassGen** の詳細については、331 ページ「ユーティリティ・リファレンス」を参照してください。

4 WebSphere MQ については、次のように入力します。

[**Host**]。WebSphere MQ キュー・マネージャが常駐するホスト。

[**Port**]。WebSphere MQ キュー・マネージャのポート番号。

[**Configuration Queue**]。構成キューの名前。

[**User**]。WebSphere インストールで接続に必要な場合は、ユーザ ID。

[**Password**]。WebSphere MQ インストールで接続に必要な場合は、パスワード。これは、PassGen ユーティリティを使用して作成される難読化形式になります。**PassGen** の詳細については、331 ページ「ユーティリティ・リファレンス」を参照してください。

[**Websphere MQ channel**]。WebSphere MQ キュー・マネージャのチャンネル名。

[**Websphere MQ Q Manager**] WebSphere の CCSID。

[**Next**] をクリックして次に進みます。

5 プレインストール・サマリ・ダイアログが開きます。

[**Install**] をクリックして、インストールを続けます。

インストールが完了したら、IIS を再起動する必要があります (258 ページ「IIS の再起動」を参照してください)。必要であれば、次項で説明するように特別な設定を行うこともできます。

.NET Agent の設定

.NET Agent の標準設定を使用して、監視されるアプリケーションのパフォーマンス測定値のキャプチャを開始できます。環境の設定や、診断するパフォーマンスの問題に合わせて Agent の設定をカスタマイズすることもできます。

標準の設定をオーバーライドするには、`< sensor_install_dir > /etc/ probe_config` ファイルにアクセスします。

次の要素を使用します。

- ▶ <tv>, <time skew>, <transport>

これらの要素は、TransactionVision についてのみサポートされます。

- ▶ <logging>, <modes>

これらの要素は、TransactionVision と Diagnostics についてサポートされます。

probe_config ファイルの詳細については、『**HP Diagnostics インストールおよび設定ガイド**』を参照してください。

<tv> 要素

目的

TransactionVision で使用する .NET Agent の設定

属性

属性	有効な値	標準設定	説明
eventthreads	数値	3	(起動時に読み取り) Agent によって生成され、Analyzer へ送信されるスレッド数。
eventthreadsleep	数値	100	(動的) メッセージ (イベント・パッケージ) 送信後のイベント・スレッドのスリープ時間 (単位ミリ秒)。
eventmemorythreshold	数値	250,000,000	(動的) 内部バッファ (Q) により消費されるメモリ容量。その後で Agent がアプリケーション・スレッドについてメッセージの送信を試みます。
configthreadsleep	数値	10,000	(動的) 構成キューをブラウジングした後のイベント・スレッドのスリープ時間 (単位ミリ秒)。

要素

発生数	1 (1 回)
親要素	ProbeConfig
子要素	transport, timeskew

使用例

```
<tv eventthreads="3" eventthreadsleep="80"
eventmemorythreshold="25000000" configthreadsleep="10000" >
  <timeskew historysize="24" checkinterval="300000" latencythreshold="100"

    retrythreshold="8"/>
    <transport type="sonicmq"
connectionstring="broker=myhost.mydomain.com;
port=21111; user=; password=;

configurationqueue=TVISION.CONFIGURATION.QUEUE"/>
</tv>
```

<timeskew> 要素

目的

タイム・サーバと .NET Agent が実行されているホスト間の時間差を計算します。タイム・サーバによるチェック頻度を設定できます。

属性

属性	有効な値	標準設定	説明
historysize	数値	24	(起動時に読み取り) 保存して最良のサンプルと比較するタイム・スキュー・サンプル数。
checkinterval	数値	300,000 ミリ秒	(動的) タイム・サーバをチェックしてスキュー・タイムを計算する前の待機時間 (単位ミリ秒)。

属性	有効な値	標準設定	説明
latencythreshold	数値	100 ミリ秒	(動的) 有効なタイム・スキュー値としてタイム・サーバからの応答に認められる最大時間 (単位ミリ秒)。
retrythreshold	数値	8	(動的) タイム・サーバのリクエストが失敗したときの試行回数。

要素

発生数	1 (1 回)
親要素	tv
子要素	なし

使用例

```
<timeskew historysize="24" checkinterval="300000" latencythreshold="100"
  retrythreshold="8"/>
```

<transport> 要素

目的

TransactionVision が使用するイベント・チャネルを設定します。

属性

属性	有効な値	標準設定	説明
type	mqseriessoni cmq	sonicmq	Agent が使用するイベント配送プロバイダ
connectionString	下記参照。		イベント配送プロバイダに関する接続情報。

type=sonicmq の場合の connectionString 構文

```
broker = <broker>; port = <port>; user = <user>; password =<password>;
configurationQueue = <configurationQueue>
```

パラメータ	説明
broker	Sonic ブローカが実行されるホスト名 普通は Analyzer ホスト名です。
port	ブローカが通信を行うポート。標準設定では 21111 です。
user	SonicMQ インストールで接続に必要な場合は、ユーザ ID。標準設定では、ユーザ名は不要です。
password	SonicMQ インストールで必要とされる場合は、パスワード。これは、PassGen ユーティリティを使用して作成される難読化形式になります。標準設定では、パスワードは不要です。 PassGen の詳細については、331 ページ「ユーティリティ・リファレンス」を参照してください。
configurationQueue	.NET TransactionVision Agent に関する構成メッセージを含むキュー名。

type=mqseries の場合の conectionString 構文

```
host= <host>; queuemanager=<queuemanager>; port= <port>; channel=,channel>
```

パラメータ	説明
host	TransactionVision 構成キューをホスティングするホスト。
queuemanager	キューマネージャ名。
port	QueueManager が通信を行う MQSeries ポート。
channel	通信に使用される MQSeries チャンネル。
configurationQueue	.NET TransactionVision Sensor に関する構成メッセージを含むキュー名。

要素

発生数	1 (1 回)
親要素	tv
子要素	なし

使用例

SonicMQ の場合

```
<transport type="sonicmq" connectionstring="broker=brokerHost;
port=21111; user=; password=;
configurationqueue=TVISION.CONFIGURATION.QUEUE"/>
```

MQ Series の場合

```
<transport type="mqseries" connectionstring="host=mqHost;
queuemanager=; port=1414; channel=TRADING.CHL;
configurationqueue=TVISION.CONFIGURATION.QUEUE"/>
```

<logging> 要素

"TVDEBUG" タグは, .NET Agent で TransactionVision 固有のコードを追跡するために指定できます。

次に例を示します。

```
<logging level="TVDEBUG"/>
```

<logging> 要素の詳細については、『**HP Diagnostics インストールおよび設定ガイド**』を参照してください。

<modes> 要素

"tv" モード・タグは, Agent を TransactionVision モードに指定します。

次に例を示します。

```
<modes tv="true"/>
```

これが指定される唯一のモードである場合, Agent は "TV only" モードで機能します。すなわち, Diagnostics プロファイラと Diagnostics プローブが無効となり, TransactionVision イベントのみが生成されます。

その他のモードが指定される場合, TransactionVision と Diagnostics のイベントが両方生成されます。

<modes> 要素の詳細については、『**HP Diagnostics インストールおよび設定ガイド**』を参照してください。

.NET Remoting クライアントおよびサーバ・アプリケーション

次の例は、.NET Remoting 環境での TransactionVision Transport に必要な設定変更を示しています。

probe_config.xml への変更

```
<tv eventthreads="5" eventthreadsleep="100"
eventmemorythreshold="25000000" configthreadsleep="60000" >
  <timeskew historysize="20" checkinterval="300000" latencythreshold="100"

      retrythreshold="6"/>
  <transport type="sonicmq"
connectionstring="broker=myhost.mydomain.com;
port=21111; user=; password=;

      configurationqueue=TVISION.CONFIGURATION.QUEUE"/>
</tv>
<process name="TVRemotingClient">
  <points file="Remoting.points"/>
  <points file="TVRemotingClient.points"/>
  <modes enterprise="true" pro="true" tv="true"/>
</process>
```

.NET Events の関連の有効化

標準設定では、.NET Events は関連されていません。ユーザの関連ビーンを有効にするには、Analyzer ホストの設定を変更します。

- ▶ EventCorrelationDefinition.xml ファイルの .NET Agent セクションのコメントアウトを解除する
- ▶ Beans.xml の次のタグのコメントアウトを解除する

```
<Attribute name="UserCorrelationBean"
value="com.bristol.tvision.services.analysis.eventanalysis.XMLRuleCorrelationBean"/>
```

EventCorrelationDefinition.xml ファイルの詳細については、『**TransactionVision Advanced Customization Guide**』（英語版）を参照してください。

IIS の再起動

.NET Agent をインストールし、必要に応じて設定を変更するか、またはカスタム測定を作成した後、ASP.NET アプリケーションで .NET Agent を使用するには、その前に IIS を再起動します。

コマンド・ラインまたは [スタート] > [ファイル名を指定して実行] メニューから IIS を再起動するには、次のように入力します。

```
iisreset
```

このコマンドによって、Web パブリッシング・サービスが再起動しますが、Sensor はすぐに起動しません。次回、アプリケーションで Web ページが要求されると、Sensor が起動し、アプリケーションが測定され、Sensor が Analyzer から構成キュー・メッセージを読み取ります。

注：ASP.NET は、アプリケーションが再デプロイされたことを検出した場合や、アプリケーションが設定されているリソースしきい値を超えている場合など、さまざまな状況でアプリケーションを自動的に再起動します。

.NET Agent のバージョンを確認する

サポートを依頼する際、インストールしているコンポーネントのバージョンを把握しておくとう便利です。

.NET Agent のバージョンを確認するには

- ▶ ファイル < **.net_agent_install_dir** > ¥bin¥HP.Profiler.dll を右クリックし、メニューから [プロパティ] を選択して、コンポーネント・バージョン情報を表示します。

.NET Agent のアンインストール

.NET Agent をアンインストールするには、次の手順を実行します。

- 1 SOAP を使用している Web アプリケーションをすべて停止します。
- 2 Windows の [プログラムの追加と削除] ユーティリティを使用します。
HP TransactionVision/Diagnostics Agent for .NET プログラムを削除します。
- 3 Web アプリケーションを再起動します。

.NET Agents 対応の SSL 設定

.NET Agent でメッセージ生成ミドルウェアに SonicMQ が使用されている場合、SSL を有効化できます。詳細については、『**HP Business Availability Center Hardening Guide**』の PDF を参照してください。

第 23 章

NonStop TMF への Sensor のインストールと設定

本章の内容

- ▶ NonStop TMF Sensor について (262 ページ)
- ▶ インストールの準備 (262 ページ)
- ▶ NonStop TMF Sensor のインストール (263 ページ)
- ▶ 起動と終了 (264 ページ)
- ▶ NonStop TMF Sensor の設定 (265 ページ)

NonStop TMF Sensor について

HP NonStop Guardian システムにあるすべての監査対象トランザクションは、監査証跡ファイルに記録されます。これには、インデックス付き (b-tree) ファイルから SQL データベースまで、あらゆる種類のデータベース・アクセスが含まれます。監査証跡データを生成する製品を Transaction Monitoring Facility (TMF) といい、TMF は NonStop システム上の監査ファイルを保護するので、すべての変更のリポジトリです。On Line Transaction Processing (OLTP) アプリケーションをサポートするため、TMF は数百のユーザが監査証跡データベースに送信した数千の複雑なトランザクションを監視できます。TMF は、多量のトランザクション処理に不可欠のトランザクション保護、データベースの一貫性、およびデータベースの復旧機能も提供します。

NonStop TMF Sensor は、Enscribe データベースの変更 (追加、削除、修正) について TMF 監査証跡データベースを監視します。NonStop TMF Sensor では、SQL リレーショナル・データベースの変更は監視されず、Enscribe ファイルの監視のみがサポートされます。

TMF は、監査証跡データベースのすべての監査対象の変更を記録するので、NonStop TMF Sensor は、対象の監査証跡レコードを読み取って、TransactionVision Analyzer に転送されて解析される TransactionVision ユーザ・イベントを作成できます。

インストールの準備

Sensor のインストーラは、Business Availability Center 製品のインストール・ディスクまたは Business Availability Center のダウンロード・ページから起動します。

次の表は、サポートされるプラットフォーム別の Nonstop TMF パッケージのインストール・ファイル名を示します。:

プラット フォーム	ファイル
Guardian	tvtmf_800_nonstop_guardian.zip

NonStop TMF Sensor のインストール

- 1 パッケージを一時ディレクトリに展開します。
- 2 FTPSCRIPT.txt を開き、次のように変更します。
 - ▶ < NonStop Username >を有効な NonStop ユーザに変更
 - ▶ < NonStop Userpassword >をユーザのパスワードに変更
 - ▶ < NonStop destination volume.subvolume >を宛先ボリュームおよびサブボリュームに変更
- 3 修正された FTP スクリプトを使用して、FTP します。

次に例を示します。

```
ftp -s:FTPSCRIPT.txt < NonStop host DNS/IP >
```

- 4 NonStop ホストに super.super としてログオンし、一時インストール \$volume.subvol に進みます。
- 5 "INSTALL" TACL マクロを実行し、宛先インストール・ボリューム / サブボリュームを入力し、ホスト DNS 名または IP アドレスを入力します。次に例を示します。

```
135> run install
This program collects configuration information in order to set up the TransactionVision
sensor for the NonStop environment. This includes:
- Installation volume and subvolume where to install the sensor
- Location of the TransactionVision Analyzer Configuration Service

You will be prompted to input required configuration parameters.
If the previous value is provided in (), pressing <Enter> will set
the parameter to the previous value.

Enter the installation volume (): $data02.tvision

Enter the analyzer configuration service DNS or IP address (): 15.178.196.101

Sensor installation complete!
```

起動と終了

NonStop TMF Sensor には、起動と終了に使用される次の 3 つの TACL マクロがあります。

- ▶ **COLDMON** : Sensor を初めて起動します。
- ▶ **STRTMON** : 停止後に Sensor を起動します。
- ▶ **STOPMON** : Sensor を停止します。

COLDMON

このマクロは、Sensor が使用した一時データ（キューに置かれたイベント）をすべて削除し、ユーザ・イベント・フィルタ条件に一致するイベントの検索を開始するように最初の監査ファイルを設定します。

NonStop Sensor がインストールされている \$volume.subvol に移動してコールドスタート・マクロを起動し、マクロを実行します（たとえば、"COLDMON" を実行します）。

"COLDMON" の起動後、監査証跡ファイルを入力して、イベントの検索を開始するよう指示されます。

次に例を示します。

```
Select TMF Audittrail file to commence scan from:
File 1: $AUDIT.ZTMFAT.AA000006
File 2: $AUDIT.ZTMFAT.AA000007
File 3: $AUDIT.ZTMFAT.AA000008
Select number from list above :
```

Sensor に特定の監査証跡ファイルで検索を開始させたいときは、そのファイルを入力し、そうでない場合は、最初に表示されるファイルで検索を開始します（「1」を選択します）。

Sensor は、フィルタ条件に一致する、現在までのイベントの検索を開始します。Sensor が初めて起動される場合、フィルタが設定されるまで、イベントの収集は行われません。Sensor は、監査ファイルをすべてスキャンして、新規監査証跡レコードを待機します。

STRTMON

Sensor が停止されると、レコードの最後の場所が、読み取られていた監査証跡ファイルに保存されます。そのため、Sensor が再起動されると、停止された場所からイベントのスキャンが開始されます。NonStop TMF Sensor を停止後に再起動するには、次の例のように、TACL マクロ STRTMON を実行します。

```
run STRTMON
```

STOPMON

NonStop Sensor を停止するには、次に示す例のように TACL マクロ STOPMON を実行します。

```
run STOPMON
```

NonStop TMF Sensor の設定

NonStop TMF Sensor は、次の 2 つのソースから設定されます。

- ▶ INSTALL TACL マクロ。起動マクロ (COLDMON/STRTMON) の値を設定します。
- ▶ Analyzer から受け取ったデータ。

TACL INSTALL マクロが COLDMON/STRTMON マクロに必要なすべての値を設定するので、一般にユーザは、起動マクロに変更を追加する必要はありません。Sensor のインストール後は、INSTALL マクロによって設定されたすべての値が Sensor によって処理されます。

NonStop Sensor は、Analyzer Configuration Service を読み取り、特にイベント送信のためのトランスポート詳細と実行するデータ収集フィルタリングを決定します。Analyzer では、トランスポート詳細は [通信リンク] の設定の構成要素です。データ収集フィルタリング情報は、[データ収集フィルタ] の設定から取得されます。『Using TransactionVision』(英語版) の「Communication Links」を参照してください。

第 24 章

WebSphere MQ Sensor の設定

TransactionVision には、次の 2 種類の WebSphere MQ Sensor があります。

- ▶ WebSphere MQ Sensor ライブラリ
- ▶ WebSphere MQ API Exit

本章の内容

- ▶ WebSphere MQ Sensor ライブラリの設定 (267 ページ)
- ▶ WebSphere MQ API Exit Sensor の設定 (278 ページ)
- ▶ WebSphere MQ Sensor と FASTPATH_BINDING (286 ページ)
- ▶ WebSphere MQ サンプルでの Sensor の使用 (286 ページ)
- ▶ WebSphere MQ クライアント・アプリケーションの監視 (287 ページ)
- ▶ WebSphere MQ-IMS Bridge Sensor の使用 (291 ページ)
- ▶ WebSphere Business Integration Sensor の使用 (297 ページ)

WebSphere MQ Sensor ライブラリの設定

WebSphere MQ Sensor ライブラリは、そのライブラリ検索パスを標準 WebSphere MQ ライブラリ検索パスの前に置くことによって、実行時に動的にロードされます。標準 WebSphere MQ ライブラリは、Sensor ライブラリによって動的にロードされます。TransactionVision を使用してアプリケーションに関するイベント・データを記録するには、事前に、標準 WebSphere MQ ライブラリの代わりに Sensor ライブラリをロードするようにアプリケーション環境を設定する必要があります。

重要： Sensor ライブラリを 64 ビット Java を含む 64 ビット・アプリケーションに使用する場合、 WebSphere MQ 32 ビット・ライブラリと 64 ビット・ライブラリの間で競合が起こることがあります。 Sensor ライブラリを 64 ビット・アプリケーションに使用しようとする発生する問題を解決するには、『**WebSphere MQ Quick Beginnings**』ドキュメントの「Implications of a 64-bit queue manager」の項を参照してください。

分散プラットフォーム

分散プラットフォームには、 z/OS や i5/OS とは別にすべてのプラットフォームが含まれます。分散プラットフォームでは、アプリケーションを実行する前に、監視対象のアプリケーションが実行されるコンピュータの環境変数設定に Sensor ライブラリのディレクトリの場所を追加する必要があります。

次の表は、 32 ビット・アプリケーションを使用する場合に各プラットフォームに設定する適切な環境変数とディレクトリの場所を示します。標準 WebSphere MQ ライブラリを含むパスが環境値の一部として存在する場合、 TransactionVision を使用するためには、そのパスの前に Sensor エントリが必要です。

プラットフォーム	環境変数	標準のディレクトリ
Windows	PATH	C:\Program Files\HP\TransactionVision\lib
Sun Solaris	LD_LIBRARY_PATH	/opt/HP/TransactionVision/lib
HP-UX	SHLIB_PATH	/opt/HP/TransactionVision/lib
IBM AIX	LIBPATH	/usr/lpp/HP/TransactionVision/lib
Redhat Linux	LD_LIBRARY_PATH	/opt/HP/TransactionVision/lib

この表に示すすべてのディレクトリの場所は、標準の Sensor インストール先です。 Sensor が標準以外の場所にインストールされている場合は、 Sensor 実行ファイルのディレクトリの場所を指定してください。

64 ビット・アプリケーションを使用する場合は、次の表に示すライブラリ・パスを設定してください。

プラット フォーム	環境変数	標準のディレクトリ
Sun Solaris	LD_LIBRARY_PATH	/opt/HP/TransactionVision/lib64:/opt/mqm/lib64
HP-UX	SHLIB_PATH	/opt/HP/TransactionVision/lib64:/opt/mqm/lib64
IBM AIX	LIBPATH	/usr/lpp/HP/TransactionVision/lib64:/usr/lpp/mqm/lib64
RedHat Linux x86-64	LD_LIBRARY_PATH	/opt/HP/TransactionVision/lib64:/opt/mqm/lib64

AIX プラットフォームでは、既存のライブラリと同盟の新規ライブラリを取得する前に、`/usr/sbin/slibclean` を実行して元の共有ライブラリをメモリからクリアする必要があります。

HP-UX プラットフォームでは、`SHLIB_PATH` 環境変数が使用できない場合、またはアプリケーションの動的ライブラリ検索パスの最初のパスでない場合は、`chatr` コマンドを使用して、アプリケーションが `Sensor` ライブラリを取得できるようにすることが必要な場合があります。`chatr` コマンドを使用して、アプリケーションの現在の設定をチェックできます。状況に関係なく、`SHLIB_PATH` が使用可能で、標準の `WebSphere MQ` ライブラリ・パスの前にあることを確認してください。詳細については、『**Using TransactionVision**』（英語版）の「**Troubleshooting**」を参照してください。

重要： `WebSphere MQ` または `UNIX` システムでマルチスレッド・アプリケーションを使用する場合、アプリケーションのスタック・サイズがスレッドに十分であることを確認してください。IBM は、マルチスレッド・アプリケーションが `MQI` 呼び出しを行う場合、256 KB 以上のスタック・サイズの使用をお勧めしています。`TransactionVision WebSphere MQ Sensor` を使用する場合は、これ以上のスタック・サイズが必要となることがあります。推奨されるスタック・サイズは 512 KB 以上です。詳細については、『**WebSphere MQ Application Programming Guide**』（英語版）の Chapter 7、「**Connecting to and disconnecting from a queue manager**」を参照してください。

ライブラリ環境をグローバルに設定せずに、センサーによる監視対象のアプリケーションを実行するには、以下のように、TransactionVision の提供する wmqsensor スクリプトを実行します。

UNIX プラットフォームの場合

```
installation_directory/wmqsensor application_command_line
```

Windows プラットフォームの場合

```
installation_directory¥wmqsensor application_command_line
```

たとえば、WMQ アプリケーションを amqsput として実行している場合、次のスクリプトを使用します。

```
installation_directory¥wmqsensor amqsput...
```

z/OS Batch, IMS, および WebSphere MQ-IMS ブリッジ

z/OS バッチおよび IMS プラットフォームでは、サンプル・プロシージャ・ライブラリ thlqual.SSLMPROC の SLMLKSTB メンバに、Sensor を WebSphere MQ アプリケーション・プログラムにバインドするサンプル・ジョブが含まれます。WebSphere MQ バッチ・サブセクション（バッチ対応の CSQBSTUB, CSQBRRSI, または CSQBRSTB, または IMS 対応の CSQQSTUB）は、アプリケーション・ロード・モジュールで対応する Sensor バッチまたは IMS スタブ（SLMBSTUB, SLMBRRSI, SLMBRSTB, または SLMQSTUB）に置き換えられます。このバインドの後、アプリケーションは WebSphere MQ を直接呼び出さず、Sensor を呼び出します。

thlqual は、TransactionVision のインストール時にシステム管理者が選択する高度な修飾子です。たとえば、高度な修飾子が TVISION である場合、サンプル・プロシージャ・ライブラリは TVISION.SSLMPROC です。

注：現在のアプリケーションがバッチに z/OS Resource Recovery Services (RRS) を使用している場合、SRRCMIT および SRRBACK の呼び出しは Sensor によって記録されず、WebSphere MQ へ渡されるだけです。

アプリケーションの実行時に、LNKLST、STEPLIB、または JOBLIB 連結に Sensor を含むライブラリが指定されていることを確認してください。UNIX System Services で、環境変数 STEPLIB を定義して、Sensor を含むライブラリを指定します。

z/OS CICS

z/OS CICS プラットフォームでは、Sensor は z/OS 対応 WebSphere MQ の CICS アダプタが提供する API 交差出口メカニズムを使用します。

i5/OS

i5/OS プラットフォームでは、メインの Sensor サービス・プログラムは、WebSphere MQ サービス・プログラムと同じ名前を持ちます。非スレッド・プログラムが LIBMQM で、スレッド・プログラムが LIBMQM_R です。この 2 つの TransactionVision サービス・プログラムは、WebSphere MQ のサービス・プログラムと同じ署名と（同じ順序で）エクスポートされたシンボルを持ちます。

TransactionVision も 次の 2 つのユーティリティ・サービス・プログラムを提供します。

- ▶ QMQM/LIBMQM にバインドされる MQMUTL5
- ▶ QMQM/LIBMQM_R にバインドされる MQMUTL5_R

メインの Sensor サービス・プログラムは、この 3 つのサービス・プログラムの 1 つにバインドされるので、プログラムの MQI 呼び出しは WebSphere MQ サービス・プログラムに渡されます。

次のいずれかのメソッドを使用して、Sensor サービス・プログラムを使用します。

- ▶ ユーザ・プログラムは、パラメータ "BNDSRVPGM(QMQM/LIBMQM)" または "BNDSRVPGM(QMQM/LIBMQM_R)" または "BNDSRVPGM(QMQM/AMQZSTUB)" を使用して、CRTPGM によって作成されます。

"ALWUPD(*YES)" および "ALWLIBUPD(*YES)" パラメータを CRTPGM に指定します。標準設定値は ALWUPD が *YES で、ALWLIBUPD が *NO です。これらのパラメータを指定しないでユーザ・プログラムが作成される場合は、いずれかのメソッドでユーザ・プログラムを再バインドします。

プログラムのバインド後、UPDPGM を使用して、WebSphere MQ および Sensor により提供されるサービス・プログラムを切り替えます。このコマンドのパラメータは SRVPGMLIB であり、QMQM または TVSENSOR のいずれかに設定できます。

- ▶ ユーザ・プログラムは、パラメータ "BNDSRVPGM(*LIBL/LIBMQM)" または "BNDSRVPGM(*LIBL/LIBMQM_R)" を使用して CRTPGM によって作成されます。プログラムのバインド後、ADDLIBLE または CHGLIBL を使用して、WebSphere MQ ライブラリ QMQM と Sensor ライブラリ TVSENSOR を切り替えます。TransactionVision を使用するためには、ライブラリ・リストで Sensor ライブラリが WebSphere MQ ライブラリ QMQM の前に置かれている必要があります。

注：ILE RPG コンパイラによって作成された RPG プログラムは、ILE C コンパイラによって作成された C プログラムと同じ WebSphere MQ サービス・プログラムを使用します。TransactionVision は、ILE RPG プログラムで MQI を使用する次のシナリオでテスト済みです。

- ▶ MQM 呼び出しに MQI を使用
 - ▶ プロトタイプ MQI 呼び出しを使用
-

Sensor ロギングの設定

オペレーティング・システムの中には、エラーの取得と WebSphere MQ Sensor のロギングの追跡に追加設定が不要のものがあります。ただし、UNIX プラットフォームでは、WebSphere MQ Sensor によって使用されるロギング機能を記録するように syslogd を設定する必要があります。UNIX プラットフォームでの WebSphere MQ センサー・ロギングの詳細については、第 26 章「Agent および Sensor ロギングの設定」を参照してください。

構成キュー名の設定

Sensor は標準設定では、WebSphere MQ API 呼び出しで指定されたキュー・マネージャで TVISION.CONFIGURATION.QUEUE という構成キューを検索します。ただし、通信リンクを作成する場合は、別の構成キュー名を指定することができます。非標準設定のキュー名を指定する通信リンクを使用する場合、TVISION.CONFIGURATION.QUEUE の代わりにそのキューで構成メッセージを検索するように Sensor を設定する必要があります。

UNIX, Windows, i5/OS

UNIX, Windows, および i5/OS プラットフォームで、Sensor を使用するすべてのプロセスについて、通信リンクに指定された Sensor 構成キューに TVISION_CONFIGURATION_QUEUE 環境変数を設定します。

IBM z/OS バッチ, IMS および WebSphere MQ-IMS ブリッジ

IBM z/OS バッチ, IMS, および WebSphere-IMS ブリッジで、SLMBCNFG というユーザ・インストール可能なロード・モジュールを生成して、Analyzer からの構成メッセージを検索するため Sensor がどのキューを読み取るかを制御できます。

- 1 thlqual.SSLMPROC(SLMBCFGQ) を編集して、次のように変更します。
 - a SET CONFIGQ ステートメントの値を変更して、希望の構成キュー名を指定します。
 - b SET THLQUAL ステートメントを変更して、TransactionVision Sensor ライブラリに高度の修飾子を指定します。
 - c システムに IBM が供給する HLASMCL プロシージャが使用できない場合、必要に応じて JCL を変更して、含まれているソース・コードを整理してバインドします。ソース・コードは変更しないでください。
- 2 thlqual.SSLMPROC(SLMBCFGQ) を送信して、変更を有効にします。

WebSphere MQ-IMS ブリッジ・センサーの詳細については、291 ページ「WebSphere MQ-IMS Bridge Sensor の使用」を参照してください。

IBM z/OS CICS で、SLMCNFQ というユーザ・インストール可能なプログラムを書いて、Analyzer からの構成メッセージを検索するために Sensor がどのキューを読み取るかを制御できます。

プログラム SLMCNFQ が Sensor によって EXEC CICS LINK から実行される場合、Sensor は、構成キュー名を格納するのに十分なサイズ（48 バイト）の CICS カンマ領域を SLMCNFQ に渡します。カンマ領域には最初、標準構成キュー名（TVISION.CONFIGURATION.QUEUE）が含まれます。Sensor が SLMCNFQ を実行すると、SLMCNFQ は、渡されるカンマ領域に希望の構成キュー名を書き込んで返します。その後、Sensor はカンマ領域を読み取り、正しい構成キュー名を検索します。

注： z/OS CICS Sensor のインストール手順では、SLMCNFQ プログラムは生成されません。このプログラムの作成方法については、この後の「SLMCNFQ の作成」を参照してください。

SLMCNFQ の実行の試みが失敗すると、z/OS CICS WebSphere MQ Sensor はプログラム SLMBCNFG のロードを試み、構成キュー名を取得します。このプログラムの生成方法については、275 ページ「SLMBCNFG の生成」を参照してください。

SLMBCNFG のロードが失敗すると、Sensor は TVISION.CONFIGURATION.QUEUE から読み取られます。

注： CICS 領域ごとに異なる構成キュー名を使用するには、275 ページ「CICS 領域ごとに異なる構成キューの使用」を参照してください。

SLMCNFQ の作成

SLMCNFQ プログラムは、CICS 領域にサポートされるどの言語でも作成できます。以下は、構成キュー名を MY.CONFIGURATION.QUEUE に設定する、C で作成されたプログラム例です。

```
#include <cmqc.h>
#include <string.h>
#include <stdlib.h>
int main(int argc, char * argv[])
{
    void *pCommArea = NULL;
    EXEC CICS ADDRESS COMMAREA(pCommArea) EIB(dfheiptr);
    if (pCommArea && (dfheiptr->eibcalen >= sizeof(MQCHAR48)))
    {
        memset(pCommArea, 0, sizeof(MQCHAR48));
        strcpy(pCommArea, "MY.CONFIGURATION.QUEUE");
    }
    EXEC CICS RETURN;
}
```

SLMBCNFG の生成

SLMBCNFG は、次のように、z/OS バッチおよび IMS と同じ方法で生成されます。

- 1 thlqual.SSLMPROC(SLMBCFGQ) を編集して、次のように変更します。
 - a SET CONFIGQ ステートメントの値を変更して、希望の構成キュー名を指定します。
 - b SET THLQUAL ステートメントを変更して、TransactionVision Sensor ライブラリに高度の修飾子を指定します。
 - c システムに IBM が供給する HLASMCL プロシージャが使用できない場合、必要に応じて JCL を変更して、含まれているソース・コードを整理してバインドします。ソース・コードは変更しないでください。
- 2 thlqual.SSLMPROC(SLMBCFGQ) を送信して、変更を有効にします。

CICS 領域ごとに異なる構成キューの使用

複数の CICS 領域がある場合、これらの CICS 領域間で Sensor ライブラリを共有して、CICS 領域ごとに異なる構成キュー名を使用できます。これを行うには、次の手順を実行してください。

- 1 CSD 定義に SLMBCNFG 表を追加していない場合は、追加してください。
SLMBCNFG 定義を以下に示します。

```
DEFINE PROGRAM(SLMBCNFG) GROUP(BTITV)
  DESCRIPTION(TVISION SENSOR CONFIGQ TABLE)
  LANGUAGE(ASSEMBLER) RELOAD(NO) RESIDENT(YES) USAGE(NORMAL)
  USELPACOPY(NO) STATUS(ENABLED) DATALOCATION(ANY)
```

- 2 &TVISION.SSLMSAMP に CONFIGQ という新規メンバを作成します。供給されるサンプル TVISION.SSLMPROC(SLMBCFGQ) から抽出されるこのメンバの内容は、次のとおりです。

```
.
*-----
* SLMCONFIG - TVision configuration macro - PLEASE DO NOT CHANGE
*-----
.
  MACRO
    SLMCONFIG &CONFIGQ=TVISION.CONFIGURATION.QUEUE
  SLMBCNFG  AMODE 31
  SLMBCNFG  RMODE ANY
  SLMBCNFG  SECT
  CONFIGQ   DC  CL48'&CONFIGQ'
  MEND
*
  SLMCONFIG CONFIGQ=&SYSPARM
  END
```

- 3 CICS 起動 JCL で、次の更新を行います。

DFHSTART の PROC ステートメントで、パラメータ CONFIGQ を 1 つ追加します。次に例を示します。

```
//DFHSTART,PROC,START=AUTO,
// INDEX1='CICSTS22',
...
// SIP=0,
// CONFIGQ='TVISION.CONFIGURATION.QUEUE',
...
```

実際の CICS 実行ステップの前にステップを追加します。

```

/*-----
// EXEC HLASMCL,
// PARM.C='NORLD,NOXREF,NORXREF,SYSPARM(&CONFIGQ)',
// PARM.L='MAP,REFR'
//C.SYSIN DD DISP=SHR,DSN=TVISION.SSLMSAMP(CONFIGQ)
//L.SYSLMOD DD
DSN=&&CONFIGQ(SLMBCNFG),DISP=(,PASS),SPACE=(TRK,(1,,1))
/*-----

```

上記で作成された一時 PDS を参照する DD ステートメントを TVISION ライブラリに先行する DFHRPL 連結に追加します。次に例を示します。

```

...
// DD DISP=SHR,DSN=&&CONFIGQ
// DD DISP=SHR,DSN=TVISION.SSLMLOAD
...

```

- 4 この変更を行った後、一意の CONFIGQ パラメータを渡すだけで、各 CICS 領域を異なる構成キュー名で起動できます。次に例を示します。

```

S CICS.CICS1,START=COLD,...,CONFIGQ='TV.CONFIG.Q1'
S CICS.CICS2,START=COLD,...,CONFIGQ='TV.CONFIG.Q2'

```

構成キュー検査間隔の設定

標準設定では、Sensor は新規構成メッセージの構成キューを 5 秒ごとに検査します。ただし、UNIX、Windows、および i5/OS プラットフォームでは、別の構成キュー検査間隔を指定することもできます。Sensor に標準以外の構成キュー検査間隔を指定するには、TVISION_CONFIG_CHECK_INTERVAL 環境変数を希望のミリ秒単位の時間間隔に指定します。

WebSphere MQ メッセージング・システム・プロバイダの設定

TransactionVision は、Analyzer と Sensor 間の通信にキューを使用します。少なくとも構成キュー、イベント・キュー、例外キューの 3 つのキューが必要です。これらのキューの標準名は、それぞれ

TVISION.CONFIGURATION.QUEUE, TVISION.EVENT.QUEUE, および TVISION.EXCEPTION.QUEUE です。これらのキューは、ベンダ固有の方法でメッセージ・システム・プロバイダに設定する必要があります。詳細は、『**Using TransactionVision**』（英語版）の「Communication Links」を参照してください。

IBM WebSphere MQ runmqsc ユーティリティ・プログラムまたは Windows と Linux で使用可能な MQ Explorer グラフィック・ユーザ・インタフェースを使用して、キューをキュー・マネージャに作成できます。クライアント接続タイプを使用する場合、サーバ接続チャンネルとリスナーもキュー・マネージャに定義する必要があります。詳細については、ベンダのドキュメントを参照してください。

WebSphere MQ API Exit Sensor の設定

WebSphere MQ API Exit Sensor は、関連するキュー・マネージャに関して生成されたすべての MQI 呼び出しを検査する出口プログラムです。出口プログラムは、MQI 呼び出しの前後に呼び出される関数を登録します。このプログラムは、次の共有オブジェクトや DLL に実装されます。

- ▶ tvisionapiexit
- ▶ tvisionapiexit_r

Sensor はキュー・マネージャについて登録されますが、実際には、MQI 呼び出しを行うアプリケーションのアドレス・スペース内にロードされて実行されます。たとえば Sensor は、キュー・マネージャ・スペースではなく、amqsput プログラム・アドレス・スペース内で実行されます。

WebSphere MQ API Exit Sensor を使用して、すべての WebSphere MQ サーバ・アプリケーションを監視できます。サーバ接続チャンネル・エージェント（リスナー）で対応する MQI 呼び出しを収集することによって、クライアント・アプリケーションを間接的に監視できます。

WebSphere MQ API Exit Sensor は、次の点が WebSphere MQ Sensor ライブラリと異なります。

- ▶ FASTPATH_BINDING を無効にする必要はありません (詳細については 286 ページ「WebSphere MQ Sensor と FASTPATH_BINDING」章を参照してください)。
- ▶ MQDISC 呼び出しの完了および理由コードは API Exit Sensor によって収集されず、それぞれ MQCC_OK および MQRC_NONE の固定値に設定されます。MQDISC イベントのイベント時間は、MQDISC 関数呼び出し時間の前に設定されます。
- ▶ API Exit Sensor は、Analyzer による WebSphere MQ (WMQ) 呼び出しから生成される TransactionVision 内部イベントを収集し、またリスナーへのクライアント接続を使用して Java Sensor からの内部 WMQ イベントも収集します。

注 : API Exit Sensor と WebSphere MQ Sensor ライブラリが同時に有効化されると、API Exit Sensor は警告をログに記録して、MQI 出口を登録せず、無効の状態になります。その後 WebSphere MQ Sensor ライブラリはイベントの処理を続行します。

分散および i5/OS プラットフォームでの API Exit Sensor の設定

WebSphere MQ API Exit Sensor を分散プラットフォームまたは i5/OS で使用するには、次の手順を実行します。

- 1 適切な WebSphere MQ API Exit Sensor の共有オブジェクトや DLL をユーザ環境でリンクします (分散プラットフォームのみ)。
- 2 必要な stanza を mqs.ini と qm.ini ファイルに追加します。

WebSphere MQ API Exit Sensor のリンク

TransactionVision は 32 ビットおよび 64 ビットの両バージョンの WebSphere MQ をサポートするので、適切な共有オブジェクトや DLL にリンクする必要があります。

WebSphere MQ V5.3 および V6.0 (32 ビット) では、次のコマンドを使用して Sensor をリンクします。

```
In -s < TransactionVision Install Directory > /lib/tvisionapiexit
/var/mqm/exits/tvisionapiexit
```

```
In -s < TransactionVision Install Directory > /lib/tvisionapiexit_r
/var/mqm/exits/tvisionapiexit_r (Solaris には不要)
```

WebSphere MQ V6.0 (64 ビット) では、次のコマンドを使用して Sensor をリンクします。

```
In -s < TransactionVision Install Directory > /lib64/tvisionapiexit  
/var/mqm/exits64/tvisionapiexit
```

```
In -s < TransactionVision Install Directory > /lib64/tvisionapiexit_r  
/var/mqm/exits64/tvisionapiexit_r (Solaris には不要)
```

次の stanza が qm.ini ファイルにあることを確認してください。

ExitPath:

```
ExitsDefaultPath=/var/mqm/exits/  
ExitsDefaultPath64=/var/mqm/exits64
```

qm.ini ファイルの ExitPath: stanza の ExitsDefaultPath パラメータ値または ExitsDefaultPath64 値、あるいはその両方が変更されると、それぞれのリンク・コマンドに記述されるディレクトリ名 /var/mqm/exits または /var/mqm/exits64、あるいはその両方を変更する必要があります。

新規 Stanza

mqs.ini ファイルの新規 stanza に、WebSphere MQ 環境全体に適用される API Exit Sensor と、個々のキュー・マネージャに適用される qm.ini ファイルを定義する必要があります。mqs.ini ファイルは、普通は /var/mqm ディレクトリにあります。qm.ini ファイルは、普通は /var/mqm/qmgrs/ < qmgr_name > ディレクトリにあります。stanza は、セクション・ヘッダとそれに続くコロン、それに続く「=」の文字で区切られた属性と値の対を含む行で構成されます。mqs.ini と qm.ini のどちらにも同じ属性を使用できます。

mqs.ini に次の stanza を追加します。

▶ ApiExitCommon

この stanza の属性は、キュー・マネージャが起動されると読み取られ、その後 qm.ini に定義される API 出口によって上書きされます。

▶ ApiExitTemplate

キュー・マネージャが作成されると、この stanza の属性が、ApiExitLocal stanza の下にある新規作成された qm.ini ファイルにコピーされます。

次の stanza を qm.ini に追加します。

▶ ApiExitLocal

キュー・マネージャが起動されると、そこに定義された API 出口によって mqs.ini の標準定義が上書きされます。

Stanza の属性と値

これらのすべての必須 stanza は次の属性と値を持ちます。

▶ Name=TransactionVisionWMQSensor

MQAXP 構造の ExitInfoName フィールドに渡される API 出口のわかりやすい名前。この属性は文字列 "TransactionVisionWMQSensor" に設定する必要があります。

▶ Function=TVisionEntryPoint

API 出口コードを含むモジュールへの関数エントリ・ポイントの名前。このエントリ・ポイントは MQ_INIT_EXIT 関数です。この属性は文字列 "TVisionEntryPint" に設定する必要があります。

▶ Module=tvisionapiexit

モジュールには API 出口コードが含まれます。この属性を TransactionVision WebSphere MQ API Exit Sensor ライブラリに設定します。個別のスレッド・ライブラリをサポートするプラットフォーム (AIX, HP-UX, および Linux) については、これは非スレッド・バージョンのセンサー・モジュールのパスに設定されます。スレッド・バージョンの WebSphere MQ アプリケーション・スタブは、ロードされる前に `_r` を暗黙的に特定のモジュール名に追記します。

重要：パスは指定しないでください。モジュール・パスは、適切なモジュールへのリンクに使用されたリンク・コマンドによって決定されます (279 ページ「WebSphere MQ API Exit Sensor のリンク」を参照してください)。その場所は、32 ビットと 64 ビットの WebSphere MQ ライブラリのいずれが使用されるかによって異なります。32 ビット・モジュールは < TransactionVision インストール・ディレクトリ > /lib にあり、64 ビット・モジュールは < TransactionVision インストール・ディレクトリ > /lib64 にあります。

▶ Data=TVQ=queue_name

Sensor が WMQ イベントを無視するキュー・オブジェクト名を設定するには、ワイルドカードを使用して TVQ 属性をオブジェクト名またはその一部に指定します。Data セクションが指定されないと、Sensor は TVISION* に一致するオブジェクトのイベントを無視します。この機能を完全にオフにするには、TVQ (Data=TVQ=) に空白の文字列を指定します。

データ・フィールドには最大 24 文字を入力できます。このため、TVQ オブジェクト名の値は最大 20 文字まで入力可能で、文字列の最初または最後、あるいはその両方にワイルドカード文字 * を含めることができます。TVQ 属性には、キュー文字列を 1 つだけ指定できます。詳細については、285 ページ「TransactionVision キューでの WebSphere MQ イベントの破棄」を参照してください。

▶ Sequence=sequence_number

TransactionVision WebSphere MQ API Exit Sensor モジュールが呼び出される、ほかの API 出口と相対的なシーケンス。シーケンス番号下位の出口が、シーケンス番号上位の出口の前に呼び出されます。出口のシーケンス番号は連続している必要はありません。1, 2, 3 のシーケンスは、7, 42, 1096 のシーケンスと同じ結果になります。2 つの出口のシーケンス番号が同じである場合、どちらが最初に呼び出されるかはキュー・マネージャが決定します。この属性は、署名無し数値です。

以下は、キュー・マネージャ (qm.ini) ごとの Sensor 設定の例です。

```
ApiExitLocal
Name=TransactionVisionWMQSensor
Sequence=100
Function=TVisionEntryPoint
Module=tvisionapiexit
```

Windows プラットフォームへの API Exit Sensor の設定

WebSphere MQ サービス・スナップインまたは **amqmdain** コマンドを使用して、Windows プラットフォームで、Windows レジストリを更新するように WebSphere MQ API Exit Sensor を設定します。

WebSphere MQ Services ノード、すなわち API 出口に関する新しいプロパティ・ページに、このノードから管理される **ApiExitCommon** と **ApiExitTemplate** の 2 種類の API 出口の説明があります。個々のキュー・マネージャに関する [Exits] プロパティ・ページで、**ApiExitLocal** を更新できます。[Configure...] ボタンにより、各 stanza 内のエントリを管理するダイアログが起動されます。このダイアログは、該当する stanza 内にすでに定義済みの API 出口の マルチコラム・リストと、出口のプロパティを追加、表示、および変更するボタンと出口を削除するボタンから構成されています。必要な stanza と属性値の説明については、279 ページ「分散および i5/OS プラットフォームでの API Exit Sensor の設定」を参照してください。

出口の定義または変更が終了したら、[OK] ボタンを押してレジストリを更新するか、または [キャンセル] ボタンを押して変更を破棄します。

監視するプログラムの識別

WebSphere MQ API Exit Sensor は、次の 2 つのファイルを使用して監視対象のプログラムを識別します。

- ▶ **exit_sensor.allow** は、監視されるプログラムを定義します。それ以外のすべてのプログラムは監視されません。このファイルが空白であると、プログラムは監視されません。i5/OS では、このファイル名は **ALLOW.MBR** です。
- ▶ **exit_sensor.deny** は、監視されないプログラムを定義します。それ以外のすべてのプログラムは監視されます。i5/OS では、このファイル名は **DENY.MBR** です。このファイルは、WebSphere MQ Sensor に付属して出荷され、WebSphere MQ コマンド・サーバなど、そのイベントが API Exit Sensor によって収集されない標準プログラムを含んでいます。

これらのファイルは、最上位の **TransactionVision** インストール・ディレクトリにあります。たとえば、Solaris で **TransactionVision** が **/opt/HP/TransactionVision** にインストールされると、この 2 つのファイルは **/opt/HP/TransactionVision** ディレクトリに置かれます。i5/OS では、これらのファイルは **/qsys.lib/tvsensor.lib/EXITSENSOR.FILE** にあります。

これらのファイルでは、コメント行は # 文字で始まります。プログラム名の最初または最後、あるいはその両方にワイルドカード文字 * を使用できます。

exit_sensor.allow と exit_sensor.deny が両方存在する場合、Sensor は exit_sensor.deny を無視します。

大部分の WebSphere MQ コマンド (プログラム) は拒否され、それらに API 出口は登録されません。追加のプログラムについては、exit_sensor.deny に名前を指定して、ユーザが拒否することができます。

注 : WebSphere MQ API Exit Sensor が使用されるとき、exit_sensor.allow ファイルが存在して、それが空白の状態だと、プログラムは監視されません。

以下の例は、WebSphere MQ リスナーからのみ収集を行う exit_sensor.allow ファイルを示しています。

```
# File: exit_sensor.allow
# Description: Only collect from WebSphere MQ Listeners
amqcrsta
amqrmppa
runmqlsr
```

次の例は、amq で始まるプログラムを除くすべてのプログラムを収集する exit_sensor.deny ファイルを示しています。

```
# File: exit_sensor.deny
# Description: Collect any program except those that
# start with "amq"
amq*
```

TransactionVision キューでの WebSphere MQ イベントの破棄

標準設定では、Sensor は、"TVISION" で始まる名前を持つキュー・オブジェクトと関連する WebSphere MQ トラフィックをすべて破棄します。別のキュー・オブジェクト名を指定するには、データ属性のオブジェクト名文字列を TVQ に設定します。指定された文字がオブジェクト名のどこに生成されるかを指定するには、次の例に示すように、ワイルドカード文字 * を使用します。

▶ HP_TV*

"HP_TV" は、すべての TransactionVision キュー・オブジェクトの接頭辞です。

▶ *HP_TV

"HP_TV" は、すべての TransactionVision キュー・オブジェクトの接尾辞です。

▶ *HP_TV*

すべての TransactionVision キュー・オブジェクトには文字列 "HP_TV" が含まれます。

注：ワイルドカードは TVQ 値の前または後、あるいはその両方に使用できますが、中に含めることはできません。たとえば、T*VISION の値は無効です。

どのキュー・オブジェクトを追跡するかを微調整する必要がある場合は、代わりにデータ収集フィルタを使用します。データ収集フィルタの使用方法については、『Using TransactionVision』（英語版）の「Data Collection Filters」を参照してください。

WebSphere MQ Sensor と FASTPATH_BINDING

分散プラットフォームにある標準 WebSphere MQ Library Sensor の場合、監視対象のアプリケーションに FASTPATH_BINDING が設定されると、アプリケーションはキュー・マネージャと同じアドレス・スペースにバインドされ、標準 WebSphere MQ ライブラリにリンクされるセカンダリ DLL のロードが試みられます。この環境は、WebSphere MQ の代わりに Sensor ライブラリをロードするように設定されているので、セカンダリ DLL は、使用できない内部シンボルの呼び出しを試みます。

この潜在的な問題を解決するため、監視対象のアプリケーションが MQCONNX を呼び出すと、Sensor は MQ_CONNECT_TYPE 環境変数を STANDARD に設定して、すべての FASTPATH_BINDING を無効にします。

WebSphere MQ サンプルでの Sensor の使用

センサーを使用して WebSphere MQ サンプル・アプリケーションを監視するには、次の制限事項に注意してください。

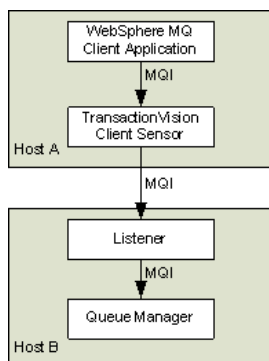
- ▶ Windows では、WebSphere MQ サンプルが 2 つの場所にあります。WebSphere MQ¥bin にあるサンプルを実行する場合、サンプル実行ファイル (amqsput, amqsget など) を別のディレクトリにコピーして、それらが標準 WebSphere MQ ライブラリの代わりに Sensor ライブラリをロードできるようにする必要があります。これは、WMQ ライブラリがこの同じフォルダに置かれており、PATH が適切に設定されていれば、Sensor ライブラリに優先するからです。WebSphere MQ¥TOOLS¥c¥samples¥bin にあるサンプルにはこの問題は起こりません。
- ▶ HP-UX および Linux プラットフォームでは、サンプル実行ファイルには WebSphere MQ ライブラリ・パスがハードコード化されているため、Sensor ライブラリはロードされません。
- ▶ WebSphere MQ サンプル amqsgbr を使用する際は、最初のパラメータに Sensor イベント・キューを使用しないでください。

WebSphere MQ クライアント・アプリケーションの監視

WebSphere MQ クライアント・バインディングを使用するアプリケーションでは、TransactionVision はこれらのアプリケーションのメッセージング活動を分散モードと集中モードのいずれかで監視および追跡することができます。

分散監視

以下の図は、分散監視環境で TransactionVision がどのように機能するかを説明しています。



一般に、WebSphere MQ クライアント・バインディングを使用するアプリケーションは、ターゲットのキュー・マネージャと同じホストで実行される「リスナー」プロセス（チャンネル・レスポンドともいう）と通信します。すべての WebSphere MQ アクティビティ（すなわち MQI 呼び出し）はリスナー・プログラムに転送されて処理され、次にリスナー・プログラムがクライアント・アプリケーションに代わって適切な MQI 呼び出しを対応するキュー・マネージャに発行します。

分散監視モードでは、TransactionVision クライアント Sensor のインスタンスが、クライアント・アプリケーションが実行されるホストにインストールされます。Sensor は、クライアント・アプリケーションによって作成された MQI 呼び出しを傍受して監視し、それに応じた追跡情報を生成し、通常の WebSphere MQ クライアント・バインディングに対応する MQI エントリ・ポイントを呼び出します。

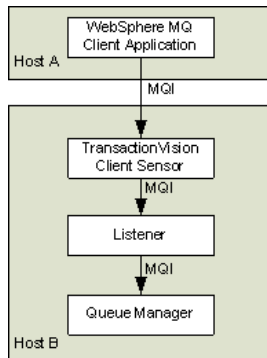
追跡情報は、クライアント・アプリケーションのコンテキストに基づいて生成されます。これは、プログラム名、プログラム・インスタンス、ホスト名などの情報が、クライアント・アプリケーションに直接関連していることを意味します。

この監視スキームには、WebSphere MQ クライアント・アプリケーションが実行されている各マシンにインストールされたクライアント・センサーが必要です。さらにクライアント Sensor は、C 言語 WebSphere MQ クライアント実行時バインディングを使用するアプリケーションの監視もできます。つまりこのクライアント Sensor は、C および C++ 言語で開発されたアプリケーションをサポートするという事です。これに対し WebSphere MQ Java クライアント・クラスは、C 実行時バインディングを使用しません。よってこのアプローチは、WebSphere MQ Java クライアント・アプリケーションまたはアプレットに適用されません。(WebSphere MQ Java Server アプリケーションは、実際は C 言語 TransactionVision Server Sensor によってサポートされます)。

このアプローチは、Windows, Solaris, HP-UX, AIX, および Linux オペレーティング・システムで実行されるクライアント・アプリケーションについてサポートされます。

集中監視

WebSphere MQ リスナー・プログラムの集中監視は、API Exit Sensor を使用する場合のみサポートされ、ライブラリ Sensor を使用する場合はサポートされません。以下の図は、集中監視環境で Sensor がどのように機能するかを説明しています。



この場合 Sensor は、リスナー・プロセスとマネージャが格納されているホストにデプロイされます。Sensor は、クライアント・アプリケーションを直接監視せずリスナー・プログラムを監視します。TransactionVision Server Sensor がデプロイされます。Sensor は、リスナー・プログラムによって発行された MQI 呼び出しを傍受してレポートを生成します。つまり、リスナー・プログラムは、クライアント・アプリケーションが呼び出すものと同じ MQI 呼び出しを実行します（後に説明する若干の例外があります）。このため、TransactionVision ユーザは、リスナー・プログラム MQI 呼び出しレコードを調べれば、クライアント・アプリケーションから生成されたメッセージング活動が一目瞭然です。

このアプローチの利点の 1 つは、Sensor のデプロイメントをキュー・マネージャが実行されているホスト・マシンに集中できることです。分散アプローチと異なり、クライアント・マシンにクライアント Sensor は不要です。このため、クライアント・アプリケーションが、さまざまな物理機能に分散された数千のマシンで実行される環境では、インストールおよび管理作業は大幅に削減されます。

さらに、このアプローチは WebSphere MQ Java クライアント・アプリケーションまたはアプレットの監視もサポートできます。このような監視は、分散モードではサポートされません。

この監視モードをデプロイする場合は、次の点に注意してください。

- ▶ Sensor はクライアント・アプリケーションを直接監視する代わりにリスナー・プログラムを監視するので、プログラム名、プログラム・インスタンス識別子、ホスト名などの報告される特定のコンテキスト情報はリスナー・プログラムに対応します。ただし、各クライアント接続はリスナー・プログラムの個々のスレッドまたはプロセス・インスタンスの中で処理されるので、同じアプリケーションと同じ接続からの MQI 呼び出しは、同じリスナー・プログラム・インスタンスにリストされます。
- ▶ リスナー・プログラムは、クライアント接続要求では MQCONN 呼び出しも MQCONNX 呼び出しも行いません。このため、各接続イベントについては、対応する TransactionVision 追跡情報は報告されません。
- ▶ 代わりにリスナー・プログラムは追加の MQI 呼び出しを行うことができます。たとえば、新しい接続を処理する場合、キュー・マネージャ情報をクエリするため、MQOPEN-MQINQ-MQCLOSE 呼び出しシーケンスが作成されます。また、クライアントからの接続解除要求を処理する場合は、MQCMIT-MQBACK 呼び出しシーケンスも作成されます。

- ▶ クライアント・アプリケーションからの MQPUT/MQPUT1/MQGET 呼び出しとリスナー・プログラムからの MQPUT/MQPUT1/MQGET 呼び出しの間には 1 対 1 の対応があります。このため、リスナーのメッセージング活動は、リスナーが対処するクライアントのメッセージング活動を正確に反映しています。
- ▶ 前述のとおり、報告されるコンテキスト情報はリスナー・プログラムと関連付けられています。ただし、クライアント・アプリケーションの発信元のコンテキスト情報は、呼び出しパラメータから MQPUT/MQPUT1/MQGET フィールドバック・データに含まれるメッセージ・ヘッダ (MQMD) 構造によって間接的に取得できます。
- ▶ リスナーからキュー・マネージャへのクライアント接続に Servlet, JMS, または EJB Sensor を使用し、これが TransactionVision WebSphere MQ Sensor によって監視される場合、内部 TransactionVision イベントがキャプチャされます。Servlet, JMS, または EJB Sensor に個別の監視されないリスナーを使用するか、またはこれらの Sensor からのサーバ・バインディング接続を使用することをお勧めします。これができない場合、データ収集フィルタを変更して、構成キューとイベント・キューを除外します。

以下の表に、2 つのアプローチの特徴をまとめます。

分散監視	集中監視
クライアント MQI による直接追跡	リスナー MQI 呼び出しによる間接追跡
クライアント・コンテキスト情報の直接アクセス	呼び出しパラメータによるクライアント・コンテキスト情報
各クライアント・ホスト上のクライアント Sensor	キュー・マネージャ・ホスト別のサーバ Sensor
WebSphere MQ C バインディングを使用したアプリケーションの監視	キュー・マネージャ・ホスト別のサーバ Sensor
Windows, Solaris, HP-UX, AIX, および Linux で実行されるクライアント・アプリケーションのサポート	Windows, Solaris, HP-UX, および AIX で実行されるキュー・マネージャへ接続するクライアントのサポート

インストールと設定の注意事項

分散監視では、WebSphere MQ クライアント・アプリケーションが実行されるホストごとにクライアント Sensor をインストールします。149 ページ「Windows での Java Agent のインストールと設定」の Sensor インストールの標準手順に従います。

集中監視では、いくつかのリスナー・プログラムが監視されるホストごとにサーバ Sensor をインストールします。

注：集中監視メカニズムは、分散監視メカニズムと排他的です。一般に、リスナー・プログラムを監視するサーバ・プログラムは、クライアント・ホストに常駐するクライアント Sensor によって監視されるクライアントを含め、リスナー・プログラムが対処するすべてのクライアントで発生する活動を報告します。集中監視アプローチを選択した場合、レポートが冗長になるのを避けるため、クライアントが監視対象のリスナーのキュー・マネージャに接続されているすべてのホストで、クライアント Sensor が無効となっていることを確認します。

WebSphere MQ-IMS Bridge Sensor の使用

WebSphere MQ-IMS ブリッジは、WebSphere MQ アプリケーションが IMS トランザクションを呼び出し、その応答メッセージを受信できるようにする WebSphere MQ コンポーネントです。このアプリケーションは IMS トランザクション・コードとそれに続くトランザクション・データで構成されるメッセージを使用して、WebSphere MQ-IMS ブリッジ入力キューに MQPUT を実行し、MQPUT のメッセージ記述子に指定される返信先キューに MQGET を実行して、IMS 出力メッセージを受信します。このインタフェースに対応するために、IMS トランザクションを変更する必要はありません。

TransactionVision WebSphere MQ-IMS ブリッジ Sensor は、呼び出し側アプリケーションが生成する WebSphere MQ API 呼び出しではなく、WebSphere MQ-IMS ブリッジ・メッセージを監視します。

Sensor の設定

WebSphere MQ-IMS ブリッジ Sensor を使用する前に、次の設定タスクを実行します。

- 1 thlqual.SSLMPROC でサンプル TVISIONB 起動プロシージャをカスタマイズして、適切な PROCLIB にコピーします。TVISIONB には、プロシージャまたは START コマンドに指定される 4 つの起動パラメータが必要です。
 - ▶ QMGR パラメータは、TVISIONB がその構成キューとイベント・キューにアクセスするために接続する必要がある WebSphere MQ キュー・マネージャの名前を指定します。このキュー・マネージャは、センサーへの通信リンクを設定する際の Analyzer の接続先キュー・マネージャであり、必ずしも WebSphere MQ-IMS ブリッジの接続先のキュー・マネージャではありません。これは、インストール時に構成キューとイベント・キューを定義するとき使用されるキュー・マネージャでなければなりません (thlqual.SSLMINST (SLMCRTQS) のサンプル・ジョブを参照してください)。
 - ▶ MAXQ パラメータは、TVISIONB がそのバッファ・キューに割り当てる最大格納領域をメガバイト単位で指定します。詳細については、294 ページ「TVISIONB バッファ・キュー」を参照してください。
 - ▶ EDPROC パラメータは、TVISIOND アドレス空間を起動するプロシージャ名を指定します。
 - ▶ IMSJOB パラメータは、監視される IMS システムの IMS 管理領域のジョブ名を指定します。
- 2 TransactionVision WebSphere MQ-IMS ブリッジ監視が要求されている IMS 管理領域ごとに thlqual.SSLMAUTH を STEPLIB 連結に含めるか、または DFSYIOE0 モジュールを既存の修飾ライブラリにコピーします。

WebSphere MQ-IMS ブリッジ Sensor の操作

WebSphere MQ-IMS ブリッジ Sensor を操作するには、次の手順を実行します。

- 1 IMS 管理領域が、STEPLIB 内でアクセス可能な TransactionVision DFSYIOE0 出口ルーチンで起動されていることを確認します。
- 2 起動プロシージャにオーバーライドされるパラメータを指定して、システム・オペレータのコンソールから TVISIONB アドレス空間を起動します。次に例を示します。

```
S TVISIONB[.jobname],IMSJOB=IMS71CR1,QMGR=CSQ1,MAXQ=10
```

Sensor の複数のインスタンスを実行する場合、インスタンスごとに一意のジョブ名を指定する必要があります。さもないと、ジョブ名はプロシージャ名に標準設定され、すべての MVS の変更および停止コマンドがすべてのインスタンスに適用されます。あるいは、監視対象の IMS システムごとに一意の名前を持つ個別の起動プロシージャが作成されます。

IMSJOB が省略されると（たとえば NUL として指定されると）、起動される Sensor インスタンスは、DFSYIOE0 出口ルーチンが起動されており、Sensor のほかのインスタンスによって明示的に監視されていない各 IMS を監視します。monitor-all Sensor が実行中に IMS システム固有の Sensor が起動されると、ターゲットの MS システムの監視は、特定の Sensor インスタンスに切り替えられます。反対に特定の Sensor が停止すると、ターゲットの IMS システムの監視は、monitor-all Sensor が実行されていれば、それに切り替えられます。混乱を避けるために、特定の Sensor だけを実行するか、または monitor-all Sensor を実行して、特定の Sensor は実行しないことをお勧めします。monitor-all Sensor を 1 つだけ許可し、特定の各 IMS システムを監視する Sensor を 1 つだけ許可します。

TVISIONB は TVISIOND を自動的に起動します。

- 3 接続されたワークステーションで TransactionVision UI/Job サーバからブリッジ監視を要求します。詳細については、『**TransactionVision User's Guide**』（英語版）を参照してください。
- 4 通常は、ブリッジ Sensor の活動は TransactionVision UI/Job サーバから制御されます。ただし、MODIFY コマンドを F TVISIONB,DISABLE MQIMSBGD のように使用して、システム・コンソールから Sensor を無効化できます。TransactionVision 出口ルーチンを無効にすると、DFSYIOE0 は IMS 管理領域で引き続き実行されますが、イベントは TVISIONB サーバ・コンポーネントに送信されません。Sensor を F TVISIONB,ENABLE MQIMSBGD コマンドで再度有効にします。
- 5 P TVISIONB のコマンドで TVISIONB アドレス空間を停止します。これにより、Sensor は暗黙的に無効化され、出口ルーチンは実行を続けますが、イベントは TVISIONB に送信されません。TVISIOND は自動的に停止されます。

バッファ・キューのイベントは、シャットダウンの完了前にイベント・ディスパッチャ・コンポーネントに送信されます。この静止機能を回避するため、P TVISIONB IMMED を使用して、即時シャットダウンを要求できます。この場合、バッファ・キューにあるすべてのイベントは破棄されます。

TVISIONB バッファ・キュー

Sensor サーバ・コンポーネントは、ストレージ内キューを保守して、既存のルーチンから TVISIONB を経て TVISIOND へフローするイベントをバッファに入れます。既存ルーチンからのイベント速度は、TVISIOND によるイベント配送速度より 7 倍も速くなることがあります。キューは、これらの各フローに応じて展開または折りたたまれます。キューの最大サイズは、REGION の指定とは無関係に制御されます。

TVISIONB 起動コマンドまたは起動プロシージャで、MAXQ=nn を指定します。この nn は、メガバイト単位のキュー最大サイズを示しています。最小サイズは 3 です。可能な最大値は 2046 です。これにより、TVISIONB は、2GB アドレス空間のすべてを使用することができます。

TVISIONB は、そのキュー・ストレージを 1 メガバイトのブロック単位で割り当て、解放します。TVISIONB が、MAXQ の制限または REGION サイズの制約のために、追加のブロックを要求されても割り当て不可の場合、警告メッセージが発行され、現在のブロックがいっぱいならば、新規ブロックが割り当て可能になるまで新規イベントは破棄されます。すでにキューに置かれているイベントは、引き続き収集されます。

ユーザの環境にとって最適の MAXQ 指定を定義するには、ある程度経験が必要です。ただし、余裕のある指定にしておいて、結果的に不要であったとしても、超過分が不要なときには MAXQ の設定に関係なく、キューが 2 メガバイトまで収縮するので、コストが高くつくことはありません。

イベント・データ

WebSphere MQ-IMS ブリッジ Sensor は、各 WebSphere MQ-IMS ブリッジ・イベントについて次のイベント・データを収集します。

- ▶ 入力および出力フラグ
- ▶ セグメント・シーケンス・インジケータ
- ▶ トランザクション・コード
- ▶ IMS メッセージ（またはメッセージ・セグメント）
- ▶ ユーザ ID
- ▶ キュー・マネージャのクロス・システム・カップリング機能（XCF）メンバ名

- ▶ 発信アプリケーションの MQPUT に指定されるメッセージ記述子 (MQMD)。

Sensor がキュー・マネージャとキュー・オブジェクトを WebSphere MQ-IMS ブリッジ入口イベント・データに追加するには、Analyzer にイベント修飾子ビーンが必要です。TransactionVision に用意されているビーンは単純なアプローチを提供します。WebSphere MQ キュー・マネージャとキュー・オブジェクトが個別の XML 構成ファイルに定義され、また定義を取り出して WebSphere MQ-IMS ブリッジ入口イベントに挿入する、特別イベント修飾子が定義されます。< **TVISION_HOME** > /config/services に置かれている次の 2 つのファイルは、WebSphere MQ-IMS ブリッジ入口イベント修飾子の設定に使用されます。

- ▶ Beans.xml
- ▶ IMSBridgeObject.xml

Beans.xml

このファイルは、ビーン処理のチェーンを定義して、イベント解析フレームワークを設定します。標準設定では、EventModifierCtx に `com.bristol.tvision.services.analysis.event-modifier.IMSBridgeEntryModifierBean` がすでに定義されており、これにより `IMSBridgeObject.xml` からオブジェクト定義が読み取られ、そのイベントが WebSphere MQ-IMS ブリッジ入口イベントである場合、定義は (XML ドキュメント・フラグメントの形式で) イベント XML ドキュメントに挿入されます。

```
<Module type="Context" name="EventModifierCtx">
<!--
  This bean read MQObject definition for IMS bridge
  entry event from $TVISION_HOME/config/service/
  IMSBridgeObject.xml
-->
<Module type="Bean" class="com.bristol.tvision.services.analysis.eventmodifier
.IMSBridgeEntryModifierBean"/>
</Module>
```

IMSBridgeObject.xml

このファイルは、以下の例に示すように、WebSphere MQ-IMS ブリッジ・イベントを生成する WebSphere MQ キュー・オブジェクトを定義します。

```
<?xml version="1.0" encoding="UTF-8"?>
<IMSBridgeMQObject>
  <MQObject objectName="IMS.BRIDGE.QUEUE" queueManager="MQS1"
  objectType="Q_LOCAL"/>
</IMSBridgeMQObject>
```

属性	説明
objectName	WebSphere MQ キュー名を定義する
queueManager	キュー・マネージャ名を定義する
objectType	キューの種類を定義する 有効値は Q_LOCAL, Q_ALIAS, Q_REMOTE, Q_CLUSTER, Q_LOCAL_CLUSTER, Q_ALIAS_CLUSTER, Q_REMOTE_CLUSTER, および DISTRIBUTION_LIST

ルート要素 IMSBridgeMQObject の下に定義されるのは MQOBJECT 要素だけです。複数の MQObject 要素が定義されていると、イベント変更ビーンは最初の要素だけを取り出します。

オブジェクトの種類に基づいて、XML ドキュメントは構造を拡張して、より詳細な情報を提供します。次の例はリモート・キュー・オブジェクトを定義します。

```
<?xml version="1.0" encoding="UTF-8"?>
<IMSBridgeMQObject>
  <MQObject objectName="REMOTE.BRIDGE.QUEUE" queueManager="MQS1"
  objectType="Q_REMOTE">
    <MQObject objectName="IMS.BRIDGE.QUEUE" queueManager="MQS2"
    objectType="Q_LOCAL">
  </MQObject>
</IMSBridgeMQObject>
```

XML スキーマは < TVISION_HOME > /config/xmlschema/IMSBridgeObj.xsd にあります。

データ収集フィルタとクエリ

ユーザ名, IMS PSB 名, IMS 領域の種類, IMS 識別子, プログラム, 入口イベント・キュー, キュー・マネージャまたはリターン・コードなどの一部のイベント属性には (データ収集フィルタやクエリでの) フィルタリングが提供されません。

WebSphere MQ-IMS ブリッジ入口または出口イベントについてフィルタリングするには, WebSphere MQ API 条件ページ (クエリ) または MQ IMS ブリッジ API 条件ページ (データ収集フィルタ) のいずれかで適切な API を選択します。

API	説明
MQIMS_BRIDGE_ENTRY	WebSphere MQ-IMS ブリッジ入口イベント
MQIMS_BRIDGE_EXIT	WebSphere MQ-IMS ブリッジ出口イベント

WebSphere Business Integration Sensor の使用

TransactionVision は, さまざまなメッセージ・フローを識別して, WBI 内の個々の論理トランザクション・パスを特定できる WebSphere Business Integration (WBI) Sensor を提供します。WBI Sensor は, 通常実行パスに挿入される追跡ノード (TransactionVisionTrace) と, 失敗パスに挿入される失敗ノード (TransactionVisionFailure) をサポートする WBI プラグインです。

処理ノードは, 既存のメッセージ・フローにどこでもいくつでも挿入できます。各処理ノードは, 現在のメッセージ・フローの状態を収集し, それを Analyzer に報告するチェックポイントです。報告されるイベントには, ブローカ名, メッセージ・フロー名, メッセージ・データなどの情報が含まれます。各ノードに一意のラベルを割り当てることができます。このラベルは, ノード・インスタンスに関連する TransactionVision イベントに報告されます。

WBI Sensor をインストールして設定するには, 次の手順を実行する必要があります。

- ▶ TransactionVision プラグインを Message Brokers Toolkit for WebSphere Studio と統合します。
- ▶ TransactionVision WBI Sensor を WBIMB プラットフォームにインストールします。

Message Brokers Toolkit for WebSphere Studio の統合

次の手順は、標準 Message Brokers Toolkit および Eclipse Technology プラグイン・インストール手順に基づくものです。

- 1 Message Brokers Toolkit が実行されていないことを確認します。
- 2 `sensor_install_directory\mqsi\TransactionVisionWBIPlugin.zip` を `WBIMB_install_directory\eclipse\plugins` に展開します。

Message Brokers Toolkit が起動すると、メッセージ・フローを編集する際に、TransactionVision 追跡ノードが、ほかのビルトイン・ノードとともに表示されます。

WBIMB 対応の TransactionVision ユーザ定義ノード統合

TransactionVision WBI Sensor を WBIMB プラットフォームにインストールするには、次の手順を実行します。

- 1 WBI メッセージ・ブローカを停止します。
- 2 WBI Sensor ユーザ定義ノード・ライブラリを対応する WBIMB インストール・サブディレクトリにコピーします。

Windows の場合

ライブラリ < `sensor_install_directory` > `mqsi\visiontrace.lil` をディレクトリ < `WBIMB_install_directory` > `bin` にコピーします。

UNIX の場合

ライブラリをディレクトリ < `WBIMB_install_directory` > `/lib` にコピーします。

- 3 WBI メッセージ・ブローカを再起動します。

ノードの挿入

ここで任意の数の TransactionVision Sensor の追跡ノードと失敗ノードを Message Brokers Toolkit を使用してメッセージ・フローに挿入できます。設定リポジトリの変更は適切なブローカにデプロイする必要があります。

TransactionVision データ収集および解析に WBI Sensor を使用する方法については、『**TransactionVision User's Guide**』（英語版）を参照してください。

第 25 章

Proxy Sensor の設定

本章の内容

- ▶ Proxy Sensor について (299 ページ)
- ▶ アプリケーションの要件 (300 ページ)
- ▶ Proxy Sensor について (300 ページ)
- ▶ プロキシ定義ファイルの設定 (300 ページ)
- ▶ ユーザ・インタフェースの設定 (303 ページ)

Proxy Sensor について

TransactionVision Proxy Sensor により、TransactionVision は、TransactionVision Sensor の使用では監視されないビジネス・プロセス・トランザクションの基本レベルの収集を行うことができます。Proxy Sensor の適切なアプリケーションの例を以下に挙げます。

- ▶ 監視されるアプリケーションが要求メッセージをキューに入れ、その後で TransactionVision Sensor (Tandem など) によりサポートされないプラットフォームで実行されているアプリケーションが、そのメッセージを検索し、処理し、応答をキューに入れ、それが監視されるアプリケーションによって検索されるトランザクション。
- ▶ 監視されるアプリケーションが要求メッセージをキューに入れ、その後で (TransactionVision がインストールされていない) ビジネス・パートナーの場所にあるアプリケーションがメッセージを検索し、処理して、応答をキューに入れ、それが監視されるアプリケーションによって検索されるトランザクション。

センサーにより監視されないアプリケーションがビジネス・トランザクションに関わっているこれらのシナリオでは、TransactionVision は Proxy Sensor を使用して、ビジネス・トランザクション全般に関する限定情報を提供できます。

TransactionVision Sensor とは異なり、Proxy Sensor は、Analyzer 内で実行される Java ビーンです。Proxy Sensor は、センサーにより監視されないアプリケーションに移動するトランザクションを認識し、特別のプロキシ・オブジェクトを作成して、トランザクションに関わる、センサーにより監視されないアプリケーションを特定します。

アプリケーションの要件

Proxy Sensor が、センサーにより監視されないアプリケーションに関わるビジネス・トランザクションと関連するためには、アプリケーションは次の要件を満たしていなければなりません。

- ▶ Sensor により監視されるアプリケーションは、メッセージ ID と 相関 ID を MQMD で保守しなければなりません。
- ▶ Sensor により監視されるアプリケーションは、返信先キューを要求に指定する必要があります。
- ▶ センサーにより監視されないアプリケーションは、応答イベントの MQMD にわかりやすいプログラム名を指定する必要があります。

Proxy Sensor について

Proxy Sensor は、TransactionVision ライセンス・コードによって有効化されます。

プロキシ定義ファイルの設定

WebSphere MQ イベントが特定のキューに置かれ、メッセージ ID と相関 ID が一致する要求応答 MQPUT-MQGET 対に属していると、Analyzer はプロキシ・オブジェクトを生成します。プロキシ定義ファイルは、プロキシ・オブジェクトの属性を定義する XML ファイルです。

このファイルは、`< TVISION_HOME > /config/services/ProxySensorDef.xml` にあります。

TransactionVision 解析に含める、センサーにより監視されないアプリケーションごとにプロキシ要素を定義する必要があります。

注：このファイルを変更するときは、変更を有効にするために Analyzer を再起動する必要があります。

次の例は、プロキシ要素をプログラム P2 用に定義しています。

```
<ProxySensor>
  <Proxy matchMsgIdToCorrelId="true">
    <Request queue="Q1" queueManager="QM1"/>
    <Reply queue="Q2" queueManager="QM2" />
    <Retrieve queue="INPUT_LQ" queueManager="DWMQ11"/>
    <Program name="P2" path="/usr/local/bin/P2path" />
    <Host name="P2_host_name" os="SOLARIS" />
  </Proxy>
</ProxySensor>
```

サブ要素

各プロキシ要素に次のサブ要素を指定します。

要素	必須か	属性	説明
Request	はい	queue queueManager	プロキシ・プログラムがイベントを取得する元の WebSphere MQ キューとキュー・マネージャ
Reply	はい	queue queueManager	プロキシ・プログラムが要求イベントと同じメッセージ ID と相関 ID を持つ応答イベントを置く WebSphere MQ キューとキュー・マネージャ
Program	はい	namepath	プロキシ・プログラムの名前とパス
ホスト	はい	nameos	プロキシ・プログラムが実行されるホストの名前とオペレーティング・システム
Retrieve	いいえ	queue	Proxy Sensor は、<応答>に定義されるオブジェクトではなく、その定義に対して特定のキュー・オブジェクトをチェックします。この要素によって、Sensor は緩い組み合わせで監視します。

要素	必須か	属性	説明
z/OSBatch	いいえ	jobID jobName stepName tcbAddr	プロキシ・プログラムが z/OS Batch ジョブである場合、そのジョブ ID、ジョブ名、手順名、および TCB アドレスを指定します。
z/OSCICS	いいえ	regionName transactionID taskNum	プロキシ・プログラムが z/OS CICS タスクである場合、領域名、トランザクション ID、およびタスク番号を指定します。
z/OSIMS	いいえ	psbName transactionName regionID jobName imsID imsType	プロキシ・プログラムが z/OS IMS ジョブである場合、PSB 名、トランザクション名、領域 ID、ジョブ名、IMS ID および IMS タイプを指定します。

プロキシ要素のオプション属性

上述のサブ要素のほかに、プロキシ要素として次のオプション属性を指定できます。

属性	説明
matchMsgIdToCorrelId	Proxy Sensor が MQPUT のメッセージ ID を MQGET の関連 ID と比較します。
matchCorrelIdToMsgId	Proxy Sensor が MQPUT の関連 ID と MQGET のメッセージ ID を比較します。
swapMsgCorrelID	true に設定すると、TransactionVision が、ルックアップ・キーの生成時に MQPUT/MQPUT1 イベントのメッセージ ID と関連 ID を切り替えます。この属性は、 matchMsgIdToCorrelId または matchCorrelIdToMsgId には使用できません。

ユーザ・インタフェースの設定

標準設定では、プロキシ関連リンクは [コンポーネント・トポロジ・アナリシス] ビューに動的モードでは表示されません。プロキシ・ノードをこのビューで有効にするには、`UI.properties` ファイルの `hasProxySensor` 属性を `true` に設定します。この構成ファイルの変更の詳細については、付録 B 「設定ファイル」を参照してください。

第 26 章

Agent および Sensor ロギングの設定

本章の内容

- ▶ ログ・ファイル (305 ページ)
- ▶ 循環ロギング (306 ページ)
- ▶ トレース・ログ (308 ページ)
- ▶ 複数の Sensor インスタンスについての個別のログ・ファイルの設定 (308 ページ)
- ▶ Windows および UNIX システム・ログの使用 (310 ページ)

ログ・ファイル

Java Agent

標準設定では、TransactionVision Java Agent は、エラーと警告メッセージを Java Agent がインストールされているログ・ディレクトリに記録します。この記録場所は、Sensor.Logging.xml ファイルに保存されます。

Java Sensor がその実行中にバナーを印刷出力できるようにするには、com.bristol.tvision.sensor.banner Java システム・プロパティを true に設定します。バナーは標準に印刷出力されます。

WebSphere MQ Sensors

標準設定では、WebSphere MQ Sensor は、エラー、警告、および追跡メッセージを UNIX システム・ログ (syslogd)、Windows イベント・ログ、z/OS オペレータ・コンソール・ログ、または i5/OS ユーザ・ジョブ・ログの local0 機能に記録します。

UNIX プラットフォームでは、`TVISION_SYSLOG` 環境変数を `user`, `local0`, `local1`, `local2`, `local3`, `local4`, `local5`, `local6`, `local7` のいずれかの値に設定することで、ログ機能を指定できます。`TVISION_SYSLOG` が設定されないか、または上記の値以外の値に設定されると、`local0` の値が使用されます。`syslogd` がログファイルに記録されるためには、ターゲット・ログ・ファイルがすでに存在していなければなりません。システム・ログ機能を設定する上で必要であれば、システム管理者に問い合わせてください。

UNIX および Windows プラットフォームでは、アプリケーションを起動する `TVISION_BANNER` 環境変数を設定します。アプリケーションが `Sensor` をロード中であることを示すバナーが表示されます。この動作を無効にするには、`TVISION_BANNER` の設定を解除します。この環境変数は、任意の値に設定できます。Windows では、この値を空文字列以外の値に設定する必要があります。i5/OS では、`TVISION_BANNER` は、UNIX の場合のようにライブラリ・パスを表示しません。

循環ロギング

標準設定では、Java Agent は循環的なログ記録の形式を使用します。ログ・ファイルが設定された最大サイズに達すると、名前を変更されてバックアップ・ファイルとなり、新しい空のログ・ファイルが作成されます。標準設定では、最大ログ・サイズは 10 MB で、バックアップ・ログ・ファイルは 1 つです。

標準設定を使用した場合、たとえば `Sensor` ログ・ファイル `sensor.log` のサイズが 10 MB に達すると、その名前が `sensor.log.1` に変更され、新しい `sensor.log` ファイルが作成されます。設定を変更してバックアップ・ファイルの数を 2 個にすると、`sensor.log` が 10 MB に達したときに以下のことが起こります。

- ▶ `sensor.log.2` が存在する場合は、それが削除されます。
- ▶ `sensor.log.1` の名前が `sensor.log.2` に変更されます。
- ▶ `sensor.log` の名前が `sensor.log.1` に変更されます。
- ▶ 新しい `sensor.log` が作成されます。

循環的なログ記録を望まない場合は、直線的なログ記録を使用するように設定を変更できます。その場合は、単一のログ・ファイルが生成されます。

< TVISION_HOME > /config/logging/*.Logging.xml ファイルは、使用されるログ記録の種類、最大ログ・ファイル・サイズ、およびバックアップ・ログ・ファイルの数を、各コンポーネントごとに指定します。たとえば、Sensor.Logging.xml には、Servlet Sensor と JMS Sensor に関する設定を指定します。このファイルには、次の例のようなエントリが含まれます。

```
<appender class="tvision.org.apache.log4j.RollingFileAppender"
name="SENSOR_LOGFILE">
  <param name="File" value="c:/Program Files/HP/TransactionVision/logs/sensor.log"/>
  <param name="Append" value="true"/>
  <param name="MaxBackupIndex" value="2"/>
  <param name="MaxFileSize" value="10MB"/>
  <layout class="tvision.org.apache.log4j.PatternLayout">
  <param name="ConversionPattern" value="%d [%t] %-5p %c %x - %m%n"/>
  </layout>
</appender>
```

ログ・ファイルの最大サイズ

ログ・ファイルの最大サイズを変更するには、MaxFileSize パラメータの値を希望するサイズに変更します。指定する値の最後には、メガバイトとキロバイトを区別するために、MB か KB を付けます。

バックアップ・ログ・ファイルの最大数

バックアップ・ファイルの最大数を変更するには、MaxBackupIndex パラメータの値を希望するバックアップ・ファイル数に変更します。

循環的なログ記録から直線的なログ記録への変更

循環的なログ記録ではなく直線的なログ記録を使用するには、次の手順を実行します。

- 1 appender class の値を RollingFileAppender から FileAppender に変更します。たとえば、上記の例では、最初の行を次のように変更します。

```
<appender class="tvision.org.apache.log4j.FileAppender"
name="SENSOR_LOGFILE">
```

- 2 MaxBackupIndex パラメータと MaxFileSize パラメータのエントリを削除します。

トレース・ログ

トレース・ロギングは、TransactionVision Sensor が内部で何を行っているかについて詳細な情報を提供します。トレース・ログは、主に問題のトラブルシューティングに使用されるものであり、実運用環境ではオンにするべきではありません。

TransactionVision Sensor でトレース・ログを有効にして、センサーの設定問題をデバッグできます。WebSphere MQ Sensor の追跡情報は、UNIX システム・ログ、Windows イベント・ログ、z/OS オペレータ・コンソール・ログ、または i5/OS ユーザのジョブ・ログに記録されます。Java Agent の場合、追跡メッセージは Agent の log4j TraceLog に送信されます。

通信リンクでの Sensor トレース・ログを有効または無効にするには、『Using TransactionVision』（英語版）の「Communication Links」を参照してください。

複数の Sensor インスタンスについての個別のログ・ファイルの設定

同一マシン上の複数のアプリケーション・サーバまたは JVM プロセスで Agent が有効化される場合、Windows または UNIX システム・ログに、あるいは各アプリケーション・サーバまたは JVM ごとに異なるログ・ファイルに記録するように Agent を設定する必要があります。これを行わないと、ログ・ファイル・エントリが破壊されたり上書きされたりします。

各 Sensor インスタンスを異なるログ・ファイルに記録されるように設定するには、次の手順を実行します。

- 1 < TVISION_HOME > /config/sensor/Sensor.properties ファイルを
< TVISION_HOME > /config/sensor ディレクトリ内に別の名前でコピーします。たとえば、serverA と serverB の 2 つのサーバに Sensor を設定する場合、Sensor.properties を Sensor_serverA.properties と Sensor_serverB.properties にコピーします。
- 2 < TVISION_HOME > /config/logging/Sensor.Logging.xml ファイルを
< TVISION_HOME > /config/logging ディレクトリ内に別の名前でコピーします。たとえば、各サーバ (serverA と serverB) について、このファイルを、Sensor.Logging.serverA.xml および Sensor.Logging.serverB.xml ファイルにコピーします。

- 3 各 Sensor.properties ファイルの logging_xml プロパティを変更します。たとえば、Sensor_serverA.properties で、プロパティ行を logging_xml=Sensor.Logging.serverA.xml に変更します。同様に、Sensor_serverB.properties で、プロパティ行を logging_xml=Sensor.Logging.serverB.xml に変更します。
- 4 JVM property com.bristol.tvision.sensor.properties を適切な Sensor.property ファイルに設定します。たとえば、serverA については JFM プロパティを次のように設定します。

```
com.bristol.tvision.sensor.properties=sensor/Sensor_serverA.properties
```

serverB については、JVM プロパティを次のように設定します。

```
com.bristol.tvision.sensor.properties=sensor/Sensor_serverB.properties
```

スタンドアロン・プログラムについては、この JVM プロパティは、JVM の起動時に次のようにコマンド・ラインに設定されます。

```
java -Dcom.bristol.tvision.sensor.properties=sensor/Sensor_serverA.properties
```

WebSphere の場合、この JVM プロパティは、[サーバ] > [アプリケーション・サーバ] > [プロセス定義] > [Java 仮想マシン] > [ユーザ定義プロパティ] にある特定のアプリケーション・サーバの管理コンソールを使用して設定されます。

WebLogic の場合、この JVM プロパティは WebLogic 起動スクリプトを使用して設定します。テキスト・エディタを開いて、スクリプトを編集します。たとえば、startWebLogic.cmd では、JAVA_OPTIONS 環境変数を次のように com.bristol.tvision.sensor.properties を含めて設定します。

```
SET JAVA_OPTIONS=%JAVA_OPTIONS% -Dcom.bristol.tvision.  
sensor.properties=<TVISION_HOME>/config/sensor/<custom properties file>
```

Windows および UNIX システム・ログの使用

UNIX および Windows プラットフォームで、TransactionVision が出力をシステム・イベント・ロギング機能（Windows ではイベント・ログ，UNIX では syslog）に記録するように設定できます。この設定に必要なロギング設定ファイル例は，`TVISION_HOME/config/logging/system/*/Sensor.Logging.xml` にあります。

Windows でも UNIX でも，専用のイベント・アペンダを定義する必要があります。

Windows イベント・アペンダ

次の例は，イベント・ログを使用するように Windows イベント・アペンダを設定する方法を示しています。

```
<appender name="NT_EVENT_LOG" class="tvision.org.apache.log4j.nt
.NTEventLogAppender">
  <layout class="tvision.org.apache.log4j.PatternLayout">
    <param name="ConversionPattern" value="%d [%t] - %m%n"/>
  </layout>
</appender>
```

これにより，ユーザが選択したカテゴリ定義で `NT_EVENT_LOG` を参照できます。次に例を示します。

```
<category additivity="false" class="com.bristol.tvision.util.log.XCategory"
name="sensorLog">
  <priority class="com.bristol.tvision.util.log.XPriority" value="info"/>
  appender-ref ref="NT_EVENT_LOG"/>
</category>
```

Windows では，特殊 DLL もパスに追加する必要があります。この DLL，すなわち `NTEventLogAppender.dll` は，`config¥logging¥system¥bin` ディレクトリにあります。次に例を示します。

```
set path=%TVISION_HOME%¥config¥logging¥system¥bin;%PATH%
```

UNIX イベント・アペンダ

次の例は、syslog を使用する UNIX イベント・アペンダを示しています。

```
<appender name="SYSLOG" class="tvision.org.apache.log4j.net.SyslogAppender">  
  <param name="SyslogHost" value="localhost"/>  
  <param name="Facility" value="local0"/>  
  <layout class="tvision.org.apache.log4j.PatternLayout">  
    <param name="ConversionPattern" value="[%t] %-5p %c %x  
- %m%n"/>  
  </layout>  
</appender>
```

SyslogHost および Facility パラメータをユーザの環境に合わせて指定します。

第 V 部

セキュリティ

第 27 章

セキュリティ

本章では、TransactionVision のセキュリティ設定手順について説明します。

本章の内容

- ▶ TransactionVision におけるセキュリティについて (315 ページ)
- ▶ TransactionVision 対応の SSL 設定 (316 ページ)
- ▶ TransactionVision ユーザ権限の管理 (318 ページ)
- ▶ 標準ロール (321 ページ)
- ▶ TransactionVision 認証と Light Weight Single Sign On (322 ページ)
- ▶ TransactionVision 設定ファイルのセキュリティ保護 (323 ページ)
- ▶ TransactionVision Analyzer のセキュリティ保護 (325 ページ)
- ▶ TransactionVision データベースのセキュリティ保護 (327 ページ)

TransactionVision におけるセキュリティについて

TransactionVision におけるセキュリティは、次のタスクによって管理されます。

- ▶ TransactionVision コンポーネント間の SSL の有効化
- ▶ ユーザ権限の制御
- ▶ Analyzer, 設定ファイル, およびデータベースのセキュリティ保護

手順 2 : TransactionVision SonicMQ Server の設定

TransactionVision Analyzer は、そのインストール時にインストールされる Progress SonicMQ 7.5 にバンドルされます。

手順 1 で生成した証明書を使用するように TransactionVision SonicMQ Server を設定します。

手順 3 : Analyzer の設定

- a HTTPS 構成メッセージ・サービスを (NonStop TMF Sensors とともに) 使用する場合は、手順 1 で生成される証明書を設定します。
- b TransactionVision SonicMQ Server と通信するには、手順 2 で設定された証明書を使用するように Analyzer を設定します。

手順 4 : UI/Job サーバの設定

- a 受け取った SSL 要求をサポートするには、手順 1 で生成された証明書を使用して TransactionVision UI/Job サーバを設定します。
- b Business Availability Center へ送信される SSL 通信をサポートするには、TransactionVision UI/Job Serve が Business Availability Center 公開証明書を使用し、また https プロトコルを使用して Business Availability Center と通信するように設定します。

手順 5 : Sensor と Agent の設定

- a TransactionVision SonicMQ Server と直接通信する場合、手順 2 で設定された証明書を使用して Sensor と Agent を設定します。
- b HTTPS 構成メッセージ・サービスを (NonStop TMF Sensors に) 使用する場合は、手順 3 で設定された証明書を使用するように Sensor を設定します。

各手順の詳細については、『**HP Business Availability Center Hardening Guide**』(英語版) PDF を参照してください。

TransactionVision ユーザ権限の管理

ユーザ権限は、特定のユーザがどの TransactionVision ページを見ることができるか、ならびにユーザが管理上の変更や特定データの表示ができるかどうかを管理します。TransactionVision 内の特定のセッションにおけるユーザ権限は、Business Availability Center でのユーザのログインと認証によって決定されます。詳細については、『**HP Business Availability Center Hardening Guide**』（英語版）PDF を参照してください。

ユーザ権限の設定

TransactionVision リソースへのユーザまたはグループ権限の設定は、Business Availability Center の [ユーザと権限] タブで TransactionVision コンテキストを選択して行います。ユーザまたはグループ権限は、リソースに事前定義されているロールを割り当てるか、または個別の操作を適用して設定できます。プロジェクトまたはユーザが作成したレポートのリソース・コンテナに権限を付与すると、同じ権限が子リソース・インスタンスと自動的に追加される後続のインスタンスにも付与されます。

標準設定では、TransactionVision の権限の変更は、ユーザが次にログインしたときに有効となります。ユーザのログイン・セッションで権限を有効にするには、`< TVISION_HOME > /config/ui/UI.properties` の TransactionVision `PermissionsCacheMaxAge` プロパティを TransactionVision が Business Availability Center から権限の更新を取得するまでの時間（分）に更新します。

注： この値を設定すると、TransactionVision が、そのユーザ権限について Business Availability Center に再クエリします。TransactionVision UI/Job サーバが別のマシンにインストールされている場合、特にそれが Business Availability Center サーバとは異なるネットワーク上にある場合、権限が過度に頻繁に再ロードするよう設定されていると、Business Availability Center での TransactionVision アプリケーションの応答が遅延する傾向があります。

TransactionVision コンテキスト内では、次のリソースが使用可能です。

注：プロジェクトとユーザが作成したレポートは、TransactionVision で使用可能なプロジェクトまたはユーザが作成したレポートまたはその両方に基づいて動的に読み込まれるコンテナです。プロジェクトの子リソース・インスタンスは、プロジェクト・ウィザードを使用して新規プロジェクトが作成または削除されると、TAS ツリーに追加または TAS ツリーから削除されます。プロジェクト・ウィザードで指定されるプロジェクト名は TAS ツリーに表示され、プロジェクトの作成者には、新規プロジェクト・リソース・インスタンスに FULLCONTROL が付与されます。

ユーザが作成したレポートの子リソース・インスタンスは、
 < TVISION_HOME > /config/ui/Reports.xml ファイルがカスタム・レポート定義を使用して更新されると、TAS ツリーに追加または TAS ツリーから削除されます。レポート要素のタイトル属性は、TAS ツリーに表示されます。

名前	操作 (権限)
プロジェクト	<p>操作が選択されないと、ユーザはプロジェクトにアクセスできません。</p> <p>VIEW (ユーザはプロジェクトの表示だけを許可され、それによりすべての組み込みレポートと表示および変更クエリへのアクセス権も与えられます)。</p> <p>CHANGE (ユーザはプロジェクトを編集、削除、作成することができ、プロジェクトのジョブを起動または停止できます)。</p> <p>FULLCONTROL (ユーザはすべての使用可能な操作を許可され、操作を付与および無効にすることもできます)。</p>
ユーザが作成したレポート	<p>VIEW (ユーザはレポートを表示できます)。</p> <p>FULLCONTROL (ユーザはすべての使用可能な操作を許可され、操作を付与および無効にすることもできます)。</p>

名前	操作（権限）
Analyzer	EXECUTE（ユーザはアナライザを起動または停止できます）。 FULLCONTROL（ユーザはすべての使用可能な操作を許可され、操作を付与および無効にすることもできます）。
ユーザ・データ	VIEW（ユーザはユーザ・データを表示できます）。 FULLCONTROL（ユーザはすべての使用可能な操作を許可され、操作を付与および無効にすることもできます）。

ユーザ・データ権限

TransactionVision により収集された一部のトランザクションとイベントの内容には、ビジネス上の機密データとみなされるものが含まれることがあります。CC 番号や社会保障番号などの項目がそうしたデータに当たります。

TransactionVision に、特定の個人だけが表示を許可される機密データが含まれることが判明した場合、これはユーザ・データ・リソースによって制御できます。ユーザ・データへのアクセスを禁止することにより、TransactionVision により表示される情報は、すべてのイベントまたはトランザクションまたはその両方のデータに使用可能な汎用データ・フィールドに限定されます。

ユーザ・データ表示権限が拒否されると、次のような影響があります。

- ▶ イベントの [イベント詳細] ウィンドウに「ユーザ・データを表示する権限がありません」と表示されます。
- ▶ SQL 統計情報レポートに完全な SQL ステートメントが表示されません。
- ▶ トランザクション追跡レポートにカスタム・カラムが表示されません。
- ▶ ビジネストランザクションの詳細で [サマリ] セクションにカスタム・データが表示されません。
- ▶ トランザクション定義エディタの [照合条件の編集] ページで txn を選択できず、分類に既存の関連 txn 情報が含まれていても、値表示枠に txn 値が読み込まれません。

注： Business Availability Center 管理者ロールには、すべての TransactionVision 権限を付与および無効化する権限があるので、Business Availability Center 管理者には、すべての TransactionVision へのアクセス権が有効化されます。ユーザ・データ・アクセスをロックする特別の保護が必要で、ただしユーザに [TransactionVision 管理] タブへのアクセスを許可する必要がある場合、ユーザに Business Availability Center 管理者ロールではなく、TransactionVision 管理者ロールを割り当てることができます。

標準ロール

標準ロールの数は、TransactionVision へのアクセスを割り当てるために定義されます。以下の表で、これらのロールについて説明します。

ロール	操作	TransactionVision アプリケーションのアクセス可能タブ
TransactionVision 管理者	プロジェクト, Analyzer, およびユーザが作成したレポートでの FULLCONTROL	[現在のプロジェクト] および [管理]
TransactionVision オペレータ	プロジェクトおよびユーザが作成したレポートでの VIEW Analyzer での EXECUTE	[現在のプロジェクト]
TransactionVision ユーザ	プロジェクトおよびユーザが作成したレポートでの VIEW	[現在のプロジェクト]

また、事前定義された Business Availability Center ロールである管理者とシステム・ビューアは、それぞれ TransactionVision 管理者と TransactionVision ユーザについて定義された操作を含めて更新されました。TransactionVision 管理者および Business Availability Center 管理者は、Business Availability Center の TransactionVision アプリケーションの [管理] タブへアクセスできる唯一のロールです。

ユーザ・データの機密的な性格のため、またうっかりアクセスを許可してしまう可能性を最小化するため、Business Availability Center 管理者だけがユーザ・データ・アクセス (FULLCONTROL 権限) を付与されています。

TransactionVision 認証と Light Weight Single Sign On

TransactionVision への安全なアクセスを保証するため、Light Weight Single Sign On (LWSSO) を有効にする必要があります。LWSSO に使用されるセキュリティ・トークンが TransactionVision と Business Availability Center の間で送受信され、正しく認証されたユーザだけが、アクセスの許可されているページにアクセスできます。Business Availability Center で LWSSO を使用する場合は、TransactionVision の追加の設定は不要です。自動的に用意され機能します。

LWSSO の要件は、完全に修飾されたドメイン名を使用して、TransactionVision および Business Availability Center サーバにアクセスしなければならないことです。デプロイメントによっては、完全に修飾されたホスト名を使用して Business Availability Center または TransactionVision あるいはその両方にアクセスすることが望ましくない場合もあります。その場合、Business Availability Center での LWSSO は無効になります。TransactionVision のセキュリティを適切に保護するためには、ユーザ認証と承認が正しく判定されるように、LWSSO と完全修飾ドメイン名が必要です。要があります。

TransactionVision を稼働させながら、何らかの理由で LWSSO を無効にしたい場合があります。この場合、TransactionVision UI/Job サーバに送信される資格情報はセキュリティで保護されずに送信されます。したがって、LWSSO を無効にすることは、認証がバイパスされるという脆弱性の原因となり、ユーザが機密データを持っているか、また正規に承認されたユーザだけに

TransactionVision データおよび管理へのアクセスを許可する必要がある場合、推奨されるアプローチではありません。Business Availability Center で LWSSO を無効にする場合、< TVHOME > /config/ui/UI.properties の bac_lwssso_enabled 設定を変更して、false に設定する必要があります。bac_lwssso_enabled を false に設定すると、TransactionVision UI/Job サーバがセキュリティの低いユーザ認証を許可します。

TransactionVision 設定ファイルのセキュリティ保護

すべての TransactionVision 設定ファイルは < TVISION_HOME > /config ディレクトリにあります。< TVISION_HOME > /bin/ TVisionSetupInfo.[sh|bat] スクリプトが実行されると、TransactionVision ログ・ファイルが存在するログ・ディレクトリの場所が指定されます。

ファイルへの書き込みアクセスを許可される唯一のユーザである TransactionVision アプリケーション管理者がすべての TransactionVision 設定ファイルを所有します。一般に、読み取り権限だけはすべての関連する TransactionVision グループに付与されます。

次の表に、TransactionVision 設定ファイルの権限を示します。

ディレクトリ	Transaction Vision UI/Job Server	Transaction Vision Analyzer	説明
< TVISION_HOME > /config/codepage/*	読み取り	読み取り	コードページ設定
< TVISION_HOME > /config/datamgr/*	読み取り	読み取り	データベース接続設定
< TVISION_HOME > /config/job/*	読み取り	なし	ジョブ・マネージャ設定
< TVISION_HOME > /config/license/*	読み取り	読み取り	TransactionVision ライセンス
< TVISION_HOME > /config/logging/*	読み取り	読み取り	ログ記録ポリシー
< TVISION_HOME > /config/rmi/*	なし	読み取り	RMI ポリシー
< TVISION_HOME > /config/services/*	なし	読み取り	Analyzer 設定
< TVISION_HOME > /config/setup/*	読み取り	読み取り	TransactionVision Setup 設定
< TVISION_HOME > /config/technologyconst/*	読み取り	なし	さまざまな定数定義

ディレクトリ	Transaction Vision UI/Job Server	Transaction Vision Analyzer	説明
< TVISION_HOME > /config/typeconversion/*	読み取り	なし	イベント詳細型変換の設定
< TVISION_HOME > /config/ui/*	読み取り	なし	レポートの設定を含む ユーザ・インタフェース の設定
< TVISION_HOME > /config/usermgr/*	読み取り	なし	ユーザ・データの設定
< TVISION_HOME > /config/xdm/*	読み取り	読み取り	さまざまな xdm の定義
< TVISION_HOME > /config/xmlschema/*	読み取り	読み取り	さまざまな xsl の定義
< user defined dir > /logs/*	書き込み	書き込み	TransactionVision ログ

注： < TVISION_HOME > /config/datamgr/Database.properties には、平文の DB2 ユーザ名とパスワードが含まれることがあるので、このファイルに読み取り権限を付与する際は特に慎重にする必要があります。セキュリティを最大にするには、ファイル所有者とシステム管理者だけに読み取り権限を割り当てます。

Windows NT のファイルへのファイル・アクセス権限を定義するには、次の手順を実行します。

- 1 システム管理者としてログインします。
- 2 Windows エクスプローラを開きます。
- 3 プロパティまたはポリシー・ファイルを右クリックして、[プロパティ] を選択します。ファイルの [プロパティ] ダイアログ・ボックスが開きます。

- 4 [セキュリティ] タブをクリックして、さまざまなユーザ・アカウントまたはセキュリティ・グループにファイルの権限を割り当てます。

UNIX で設定ファイルの権限を設定するには、次の手順を実行します。

- 1 システム管理者としてログインします。
- 2 `chmod` を実行して、適切なアクセス権をさまざまな TransactionVision グループに付与します。`/datamgr/database.properties` には `rw--r--` を、その他のすべての設定ファイルには `rw-r--r--` を設定することをお勧めします。
- 3 必要に応じて、`chgrp` を使用してファイル・グループを変更し、`chown` を使用してファイルの所有者を変更します。

TransactionVision Analyzer のセキュリティ保護

TransactionVision Analyzer は、TransactionVision Sensor および Agent と通信して、収集されたイベント・データを処理して有用な解析を行います。Analyzer は Windows では Windows サービスとして実行され、UNIX ではデーモンとして機能します。

TransactionVision は、RMI (Remote Method Invocation) を使用して、指定されたポート (標準設定は 21100) で Analyzer と通信します。次に、TransactionVision UI/Job サーバとコマンド・ライン・ユーティリティが RMI を使用して Analyzer と通信し、サービス・シャットダウンを開始し、収集を開始および終了できるようにし、ステータス情報を取得して、Analyzer の動作を制御します。Analyzer で使用されるユーティリティについては、『**Using TransactionVision**』(英語版)の「Administration Utilities」を参照してください。

TransactionVision では、ファイル < TransactionVision Install Directory > /config/rmi/RMI.policy に RMI セキュリティ・ポリシーが定義されます。標準設定では、`permission java.security.AllPermission` と定義されます。この定義で最も重要なことは、TransactionVision インストールへのアクセス権を持つユーザに Analyzer へのリモート・アクセスが許可されることです。TransactionVision の管理者は、このことを認識しておく必要があります。Analyzer へのより安全なアクセスが望ましい場合は、その他のオプションを組み合わせることでアクセスを制御することができます。

Analyzer が実行されているホストから Analyzer へのアクセスを制御する最も単純なメカニズムは、TransactionVision インストールのファイル・アクセス権限です。これらのアクセス権限は、必要に応じて調整できます。たとえば UNIX では、読み取り、書き込み、または実行権限をインストール・ファイルの所有者、または TransactionVision の管理権限を持つ指定グループのユーザのみに許可することです。

リモート・マシンから Analyzer へのアクセスを制御するため、前述の RMI.policy ファイルを使用できます。java.net.SocketPermission を付与することにより、どのマシンから Analyzer へのリモート・アクセスが許可されるかを制御できます。これは以下のテンプレートを使用して実行できます。少なくとも RMI.policy ファイルは、UI/Job サーバが実行されているマシンにソケット権限を与えなければなりません。さもないと、Analyzer を Web インタフェースから制御することができません。

```
grant {
  permission java.net.SocketPermission "webserver:*", "accept, connect, listen, resolve";
  permission java.net.SocketPermission "remotepc:*", "accept, connect, listen, resolve";
  permission java.net.SocketPermission "localhost:*", "accept, connect, listen, resolve";
  permission java.io.FilePermission "<<ALL FILES>>", "read, write, delete, execute";
  permission java.net.NetPermission "***";
  permission java.awt.AWTPermission "***";
  permission java.util.PropertyPermission "***, "read, write";
  permission java.lang.reflect.ReflectPermission "***";
  permission java.lang.RuntimePermission "***";
  permission java.security.SecurityPermission "***";
  permission java.io.SerializablePermission "***";
};
```

この例では、マシン "webserver" および "remotepc", ならびに Analyzer が実行されているローカル・マシンが、Analyzer へのコマンドが承認される唯一の発信元マシンです。

"java.net.SocketPermission" を除いて、ほかの権限は、Analyzer 自身が必要とする最小セットの権限であり、これらの変更はお勧めできません。これらの権限の詳細については、<http://java.sun.com/j2se/1.3/docs/guide/security/permissions.html> に追加情報があります。

TransactionVision データベースのセキュリティ保護

TransactionVision データベースの安全保護の目的は次のとおりです。

- ▶ ビジネス上のニーズがないユーザによる分類されたデータへの未承認アクセスの禁止
- ▶ 悪意によるデータの削除やデータの改ざんなど未承認ユーザによる被害の防止
- ▶ 監査技術によるデータのユーザ・アクセスの監視

データベース・アクセスは一般に、ユーザ認証とユーザ承認によって制御されます。認証は、ユーザの身元を確認するプロセスであるのに対し、承認は、要求されたリソースにアクセスする権限がユーザにあるかどうかを決定するプロセスです。データベース認証を実行するパラメータは、TransactionVision で設定されます。TVisionSetupInfo で、データベース・ユーザ名とパスワードの入力を求められます。ユーザ名とパスワードは、設定ファイル < TVISION_HOME > /config/datamgr/Database.properties に（パスワードは暗号化されて）保存されます。ユーザ認証の基礎となるメカニズムは、データベース製品ごとに異なることがあります。TransactionVision は、DB2 にはオペレーティング・システム・ベースの認証を、Oracle にはデータベース認証を、また SQL Server には 'SQL Server Authentication' をサポートします。

TransactionVision のテーブルとスキーマ

TransactionVision では、2 種類のデータベース・スキーマが使用されます。すなわち、TVISION という名前のシステム・スキーマとユーザが定義するプロジェクト・スキーマです。TVISION スキーマは、イベントに無関係の情報を保存するのに対し、プロジェクト・スキーマは TransactionVision センサーが収集したプロジェクト・イベントを保存します。

TVISION システム・スキーマは、TVisionSetupInfo で自動的に作成することも、スクリプト " < TVISION_HOME > /bin/CreateSqlScript.[sh|bat] -c -e -system を実行して手動で作成することもできます。その後で、TransactionVision を起動して、Web アプリケーションまたは CreateSqlScript から TransactionVision プロジェクト・スキーマを作成できます。

データベース権限

TransactionVision を特定のデータベース環境で実行するには、TransactionVision に設定されたデータベース・ユーザがすべての必須データベース・オブジェクトにアクセスするのに必要な権限を持っていることを確認する必要があります。これらの権限には次のものがあります。

- ▶ データベース・システム・カタログ・テーブルの SELECT 権限
- ▶ スキーマ TVISION のすべてのテーブルの SELECT, INSERT, UPDATE, および DELETE 権限
- ▶ プロジェクト・スキーマのすべてのテーブルの SELECT, INSERT, UPDATE, および DELETE 権限
- ▶ インストール時に TVISION システム・テーブルを作成し、TransactionVision UI プロジェクト作成ウィザードからプロジェクト・スキーマ・テーブルを作成する CREATE TABLE 特権

TransactionVision に専用のデータベースが使用される場合、データベースにアクセスするには、普通は、事前に定義された必要なデータベース権限を持つユーザ（たとえば DBA ロールを持つユーザなど）を選択すれば十分です。これが不可能な場合、上記の特権を明示的に付与する必要があります。いったんテーブルが作成されると、TransactionVision の実行に CREATE TABLE 権限は不要になるので、データベース DBA が TVISION およびプロジェクト・テーブルを手動で作成して、これらの権限を付与する必要がなくなります。ユーティリティ 'CreateSqlScript' を使用して、必要な SQL スクリプトを生成し、TVISION およびプロジェクト・テーブルを作成し、また設定されたユーザが TVISION およびプロジェクト・テーブルにアクセスするのに必要な権限を付与する次のスクリプトを作成できます。

- ▶ CreateSqlScript -create -system
- ▶ CreateSqlScript -create -schema SCHEMA
- ▶ CreateSqlScript -grant -system
- ▶ CreateSqlScript -grant -schema SCHEMA

設定されたデータベース・ユーザが CREATE TABLE 権限を持っていない場合、Business Availability Center の TransactionVision アプリケーション内から新規プロジェクトのテーブルを作成することはできません。代わりに、[新規プロジェクト] ウィザードを実行する前に、手動でテーブルを作成する必要があります。

第 VI 部

付録

付録 A

ユーティリティ・リファレンス

本章の内容

- ▶ CreateSqlScript (332 ページ)
- ▶ DB2RunStats (335 ページ)
- ▶ DB2Test (337 ページ)
- ▶ MigrateDB (338 ページ)
- ▶ nanny (339 ページ)
- ▶ OracleRunStats (342 ページ)
- ▶ OracleTest (344 ページ)
- ▶ PassGen (346 ページ)
- ▶ rebind_sensor (347 ページ)
- ▶ rebind_tux_sensor (349 ページ)
- ▶ runSupportSnapshot (350 ページ)
- ▶ ServicesManager (353 ページ)
- ▶ SetupModule (356 ページ)
- ▶ SQLServerTest (357 ページ)
- ▶ TVisionSetupInfo (358 ページ)

CreateSqlScript

保管場所

TVISION_HOME/bin/CreateSqlScript.[sh|bat]

目的

ユーザが SQL スクリプトを作成し、任意で実行して、TransactionVision システム・テーブルまたはプロジェクト・テーブルを作成、削除、インポート、またはエクスポートできるようにします。

注：データベースのインポートまたはエクスポートを行う前に、TransactionVision Analyzer および UI/Job サーバを停止する必要があります。これは、Analyze または UI/Job サーバが実行するデータベース・ロックによってデータベースのインポートまたはエクスポートが失敗するのを回避するためです。スキーマが削除される場合、有効なプロジェクトまたはそのプロジェクトにログインしているユーザがそのスキーマを使用していないことを確認してください。

構文

```
CreateSqlScript
{-create(-c) | -drop(-d) | -import(-i) | -export(-ex) | {-system(-sys) |
-schema(-s)SCHEMA | -table(-t) TABLE SCHEMA}
[[-noscript(-n)] -execute(-e)]
    [-noprompt(-np)] [-noinsert(-ni)] (
    [-tablespace(-ts) TABLESPACE] [-dbMove(-m)]
    [-fileType(-f) IXF|DEL] [-lobPath(-lp) PATH]
    [-noLob(-nl)] [-dbproperties(-db) FILE]
```

オプション

オプション	説明
-drop (-d)	テーブルを削除する
-create (-c)	テーブルを作成する
-execute (-e)	スクリプトを実行する

オプション	説明
-noscript (-n)	スクリプトを生成しない
-system (-sys)	システム・テーブル (スキーマ TVISION) を作成または削除する
-schema (-s) SCHEMA	スキーマ SCHEMA のプロジェクト・テーブルを作成または削除する
-table (-t) TABLE SCHEMA	スキーマ SCHEMA のテーブル TABLE を作成または削除する
-tablespace (-ts) TBSPC	表領域 TBSPC を使用する
-dbproperties (-db) FILE	データベース・プロパティ・ファイル FILE を使用する
-import (-i)	データベース・インポート・スクリプトを生成します。データベース・スクリプトを実行するには、コマンド <code>db2 -n -t -f < sql script filename ></code> を使用します。DB2 データベースについては、このオプションを <code>-fileType</code> および <code>-lobPath</code> オプションと組み合わせて、LOB のデータ形式と保管場所をカスタマイズできます。このオプションを <code>-noscript</code> または <code>-execute</code> オプションと組み合わせることはできません。
-export (-ex)	データベース・エクスポート・スクリプトを生成します。データベース・スクリプトを実行するには、コマンド <code>db2 -n -t -f < sql script filename ></code> を使用します。DB2 データベースについては、このオプションを <code>-fileType</code> および <code>-lobPath</code> オプションと組み合わせて、LOB のデータ形式と保管場所をカスタマイズできます。このオプションを <code>-noscript</code> または <code>-execute</code> オプションと組み合わせることはできません。
-resetseq (-r)	照合するシーケンス開始番号をリセットして、インポートされたデータと一致させます。
-dbMove (-m)	IMPORT/EXPORT スクリプトを生成する代わりに、スキーマをインポートまたはエクスポートする対応する <code>db2move</code> コマンドを表示します (<code>-import/-export</code> との併用に限定)。

オプション	説明
-noprompt (-np)	データベース・テーブルが削除される時確認のプロンプトを表示しません。
-noinsert (-ni)	データベース・テーブルの作成時に、DMX ファイルに定義された最初のテーブル行を挿入しません。
-fileType (-f) IXF DEL	データのインポートおよびエクスポートに使用される DB2 データ出力ファイル形式を指定します。標準設定のタイプは IXF です。IXF ファイル・タイプでは、インポートおよびエクスポート・スクリプトは、32K 以上のデータを含む行に LOBFILE オプションを使用します。DEL タイプでは、インポートおよびエクスポート・スクリプトは、LOBFILE を 1 つのファイル (FixPack 8 が必要) にエクスポートします。
-lobPath (-lp) PATH	DB2 LOBFILE のディレクトリを指定します。標準値はカレント・ディレクトリです。
-noLob (-nl)	LOBINFILE オプションで DB2 export/import SQL を生成しません。このオプションは、LOB データを最初の 32K バイトに切り詰めます。

使用例

- 1 TVISION のスキーマを使用してシステム・テーブルを作成し、SQL スクリプトを生成せずにプロシージャを実行します。

```
CreateSqlScript -e -n -c -sys
```

- 2 PROJECT のスキーマを使用して、プロシージャを実行せずにプロジェクト・テーブルを作成する SQL スクリプトを生成します。

```
CreateSqlScript -c -s PROJECT
```

- 3 スキーマ PROJECT のテーブル EVENT を削除し、SQL スクリプトを生成せずにプロシージャを実行します。

```
CreateSqlScript -e -n -d -t EVENT PROJECT
```

DB2RunStats

保管場所

TVISION_HOME/bin/DB2RunStats.[sh|bat]

説明

テーブルおよび関連インデックスの物理特性についての統計情報を更新します。これらの特性には、レコード数、ページ数、平均レコード長などがあります。オプティマイザは、データへのアクセス・パスを決定する際にこれらの統計情報を使用します。

データが TransactionVision Analyzer によって継続的に DB2 に収集される場合など、テーブルが大幅に更新されると、このコマンドが呼び出されます。その結果、TransactionVision のビューとレポートにより生成されるクエリと、TransactionVision Analyzer がイベントを相関するために内部で生成するクエリのパフォーマンスが大幅に向上します。

- ▶ このスクリプトは、UNIX **cron** 機能または Windows スケジューラを使用して、定期バッチ・ジョブとして実行するように設定できます。
- ▶ このスクリプトの実行時には、TransactionVision Analyzer の処理速度が低下します。
- ▶ メッセージ比率が高くなれば実行頻度が高くなりますが、このスクリプトは、通常毎日実行されます。
- ▶ DB2 のインストール先と TVISION_HOME の保管場所を適切に設定するには、このスクリプトをユーザのシステムに基づいてカスタマイズする必要があります。
- ▶ プロジェクト・スキーマ用にさらにテーブルが定義されている場合、追加のテーブルをカバーする新しい RUNSTATS ステートメントを追加する必要があります。

構文

```
DB2RunStats username passwd database-name schema-name [-v7]
```

オプション

オプション	説明
-username	RUNSTATS を実行してデータベース接続を行う権限を持つユーザ・アカウント
-passwd	user_name と関連するパスワード
-database-name	接続先のデータベース名
-schema-name	プロジェクトがデータを読み取りおよび書き込みするスキーマの名前
-v7	DB2 7.x 環境で使⽤します。標準操作は、DB2 8.1 用の操作です。

DB2Test

保管場所

com.bristol.tvision.admin.DB2Test

説明

DB2 データベースの INSERT パフォーマンスを測定します。

このユーティリティはサンプル・イベント・データを指定されたスキーマの RAW_EVENT テーブルに挿入します。テストの実行前に、CreateSqlScript（テスト実行後に削除できます）を使用して新規プロジェクト・スキーマを作成します。必要なテスト環境の設定方法の詳細については、『**TransactionVision Planning Guide**』（英語版）を参照してください。

構文

```
java com.bristol.tvision.admin.DB2Test databaseName user passwd schema
eventCount eventSize threadCount {-commit n}{-jdbcBatch}
```

オプション

オプション	説明
-databaseName	テストに使用されるスキーマを含む DB2 データベースの名前
-user	DB2 ユーザ名
-passwd	DB2 パスワード
-schema	サンプル・イベント・データが保存される TransactionVision スキーマ
-eventCount	生成されるイベントの数
-threadCount	イベントの生成に使用されるスレッドの数
-commit n	単一のバッチ操作の一部として n 回挿入ステートメントを実行します。標準設定では、各挿入を個別にコミットします。
-jdbcBatch	JDBC 標準バッチ操作を使用

MigrateDB

保管場所

TVISION_HOME/bin/MigrateDB.[sh|bat]

説明

プロジェクト・データベース・ファイルを旧バージョンの TransactionVision から現在のバージョンに移行します。このスクリプトは、インストール・プロセスのアップグレード後に実行する必要があります。このスクリプトは設定済みの TransactionVision 環境で実行する必要があります。データベースとの通信には、**Database.properties** を適切に設定する必要があります。

構文

```
MigrateDB
```

nanny

保管場所

TVISION_HOME/bin/nanny.[sh|bat]

説明

Nanny Manager によって制御されるサービスを管理します。SonicMQ ドメイン・マネージャ、SonicMQ Broker、Analyzer、および UI/Job サーバなどの TransactionVision コンポーネントは、Nanny Manager によって管理（起動、停止、監視）されるサービスです。

Nanny Manager は、Windows の（HP Business Availability Center という）サービスであり、UNIX のプロセス（nannyManager）です。この nanny ユーティリティを使用して、Nanny Manager により管理されているコンポーネントを特定できます。

構文

```
nanny -s <hostname> -p <port> -disableService
<servicename> -enableService <servicename> -getServiceInfo <servicename>
-isServiceRunning <servicename> -listAllDisabledServicesNames
-listAllServicesNames -listDeadServicesNames -listLiveServicesNames
-listServiceInfo -listStartingServicesNames -restartService <servicename>
-retrieveNannyManagerHTMLAdapterPort -showStackTrace
<servicename> -showStackTrace <servicename> -startAllServices -startService
<servicename> -stopAllServices -stopService <servicename>
```

オプション

オプション	説明
-s < hostname >	Analyzer がインストールされるホストのホスト名。標準設定ではローカルホストです。
-p < port >	nanny サーバのポート。標準設定では 11020 です。
-disableService < servicename >	< servicename > で指定されるサービスを無効にします。
-enableService < servicename >	< servicename > で指定されるサービスを有効にします。

オプション	説明
-getServiceInfo < servicename >	< servicename > についてのサービス情報を表示します。
-isServiceRunning < servicename >	< servicename > で指定されたサービスが実行されているかどうかをチェックします。
-listAllDisabledServicesNames	無効化されたすべてのサービス名を一覧表示します。
-listAllServicesNames	すべてのサービス名を一覧表示します。
-listDeadServicesNames	実行されていないすべてのサービスを一覧表示します。
-listLiveServicesNames	実行中のサービスを一覧表示します。
-listServiceInfo	すべてのサービスの属性を一覧表示します。
-listStartingServicesNames	起動状態のすべてのサービスを一覧表示します。
-restartService < servicename >	< servicename > で指定されるサービスを停止および起動します。
-retrieveNannyManagerHTMLAdapterPort	nanny jmx ポートを表示します。
-showStackTrace < servicename >	< servicename > で指定されるサービスのスタックトレースを表示します。
-startAllServices	すべてのサービスを起動します。
-startService < servicename >	引数で指定された特定のサービスを起動します。
-stopAllServices	すべてのサービスを停止します。
-stopService < servicename >	引数で指定された特定のサービスを停止します。

使用例

Nanny Manager によって管理されるすべてのサービスを一覧表示するには、次のコマンドを入力します。

```
nanny.bat listAllServiceNames
```

このコマンドにより、次の出力が生成されます。

```
Executing: listAllServicesNames Service: on (localhost,11020)
```

Result = [tv_as, tv_message_broker, tv_domain_manager, tv_analyzer]

nanny ユーティリティを使用してコンポーネントを有効または無効化できません。コンポーネントがいったん無効化されると、有効化されるまで Nanny Manager によって管理されません。

Analyzer を無効化するには、次のコマンドを実行します。

```
nanny.bat disableService tv_analyzer
```

Analyzer を有効化するには、次のコマンドを実行します。

```
nanny.bat enableService tv_analyzer
```

nanny ユーティリティを使用して、コンポーネントを有効化した後に起動することもできます。

Analyzer コンポーネントを起動するには、次のコマンドを実行します。

```
nanny.bat startService tv_analyzer.
```

OracleRunStats

保管場所

TVISION_HOME/bin/OracleRunStats.[sh|bat]

説明

注： Oracle 10g には、統計情報を自動収集する組み込みのジョブ・スケジューラがあります。バージョン 10g には、このスクリプトを実行する必要はありません。

テーブルおよび関連インデックスの物理特性についての統計情報を更新します。これらの特性には、レコード数、ページ数、平均レコード長などがあります。オプティマイザは、データへのアクセス・パスを決定する際にこれらの統計情報を使用します。

データが TransactionVision Analyzer によって継続的に Oracle に収集される場合など、テーブルが大幅に更新されると、このコマンドが呼び出されます。その結果、TransactionVision のビューとレポートにより生成されるクエリと、TransactionVision Analyzer がイベントを相関するために内部で生成するクエリのパフォーマンスが大幅に向上します。

このスクリプトは、UNIX **cron** 機能または Windows スケジューラを使用して、定期バッチ・ジョブとして実行するように設定できます。

注： このスクリプトの実行時には、TransactionVision Analyzer の処理速度が低下します。

メッセージ比率が高くなれば実行頻度が高くなりますが、このスクリプトは、通常毎日実行されます。

このスクリプトは、データベース操作を実行する権限を持つユーザ・アカウントから実行します。このスクリプトは、Oracle ユーザ・アカウントから実行されなければなりません。

注： .profile や .bashrc などの適切な環境の初期化スクリプトを呼び出し、Oracle インストールの場所や TVISION_HOME の保管場所を適切に設定するには、このスクリプトをユーザのシステムに基づいてカスタマイズする必要があります。また、このスクリプトの実行時には、ファイル OracleRunStats.sql が現在のディレクトリに置かれていなければなりません。

構文

```
OracleRunStats user_name passwd database_name schema_name
```

オプション

オプション	説明
-user_name	スクリプトが実行される Oracle ユーザ・アカウント
-passwd	user_name と関連するパスワード
-database_name	接続先のデータベース名
-schema_name	プロジェクトがデータを読み取りおよび書き込みするスキーマの名前

OracleTest

保管場所

com.bristol.tvision.admin.OracleTest

説明

Oracle データベースの INSERT パフォーマンスを測定します。

このユーティリティはサンプル・イベント・データを指定されたスキーマの RAW_EVENT テーブルに挿入します。テストの実行前に、CreateSqlScript (テスト実行後に削除できます) を使用して新規プロジェクト・スキーマを作成します。必要なテスト環境の設定方法の詳細については、『**TransactionVision Planning Guide**』(英語版) を参照してください。

構文

```
java com.bristol.tvision.admin.OracleTest databaseName host port user passwd
schema eventCount eventSize threadCount [-VARCHAR | -BLOB |
-LONGRAW] {-commit n} {-jdbcBatch} {-OracleBatch} {-thin} {-parallel} {-url URL}
```

オプション

オプション	説明
-databaseName	テストに使用されるスキーマを含む Oracle データベースの名前
-host	Oracle サーバが存在するホスト・システムの名前
-user	Oracle ユーザ名
-passwd	Oracle パスワード
-schema	サンプル・イベント・データが保存される TransactionVision スキーマ
-eventCount	生成されるイベントの数
-eventSize	イベント・ユーザ・データ・バッファのサイズ (標準設定では 1024 バイト)
-threadCount	イベントの生成に使用されるスレッドの数
-VARCHAR	VARCHAR カラム定義に RAW_EVENT テーブルが作成されている場合は、このオプションを使用します。

オプション	説明
-BLOB	BLOB カラム定義に RAW_EVENT テーブルが作成されている場合は、このオプションを使用します。
-LONGRAW	LONGRAW カラム定義に RAW_EVENT テーブルが作成されている場合は、このオプションを使用します。
-commit n	単一のバッチ操作の一部として <i>n</i> 回挿入ステートメントを実行します。標準設定では、各挿入を個別にコミットします。
-jdbcBatch	JDBC 標準バッチ操作を使用します。
-oracleBatch	Oracle 更新バッチ操作を使用します。
-thin	ユーザ軽量クライアント・ドライバ。標準設定では、oci クライアント・ドライバを使用します。
-parallel	標準 INSERT INTO の代わりに Oracle INSERT PARALLEL オプションを使用します。
-url URL	標準設定では、OracleTest は軽量または oci クライアント・ドライバに適切な JDBC URL を使用します。ただし、ここで JDBC URL を指定して、標準設定を上書きできます。

PassGen

保管場所

TVISION_HOME/bin/PassGen

説明

パスワードは、PassGen ユーティリティを使用して難読化できます。

構文

```
PassGen /system <password>
```

オプション

オプション	説明
-password	最大 128 文字の英数字からなる文字列。

使用例

```
cd < installdir > /bin  
$ PassGen /system TheLazyFoxJumpedHigh
```

```
password:OBF:3q6r3xxz3y3r3xjs3wx03yc63n0r3lbr3vc03wd745893wre44u041  
3j3kn93zwy40vi432i44fr3m453m894493439040pc40303kjd419r44na3wx0451h  
3wir3v6m3lfr3mwj3yi03wre3xpi3xxz3y3r3q23
```

rebind_sensor

保管場所

TVISION_HOME/bin/rebind_sensor.sh

説明

このスクリプトは、AIX プラットフォームで TransactionVision WebSphere MQ センサーを再バインドします。

WebSphere MQ サポート・パックに、AIX プラットフォームで TransactionVision WebSphere MQ センサーからエクスポートされる内部シンボルを変更できます。センサー・ライブラリからエクスポートされた内部シンボルが WebSphere MQ ライブラリで使用できなくなると、アプリケーションは起動できなくなり、失敗してさまざまなシンボル解決エラーを表示します。

このため、WebSphere MQ サポートまたは修正パックのインストール時には、rebind_sensor スクリプトを実行する必要があります。

これにより TVISION_HOME/lib の TransactionVision Sensor ライブラリが修正されます。

WebSphere MQ 6.0 以降では、このユーティリティを 2 回実行して、次のように 32 ビット・ライブラリと 64 ビット・ライブラリをインストールメント化する必要があります。

`$TVISION_HOME/bin/rebind_sensor.sh` (32 ビット・ライブラリをバインドしなおす)

`$TVISION_HOME/bin/rebind_sensor.sh -64` (64 ビット・ライブラリをバインドしなおす)

構文

```
rebind_sensor.sh [-v|-s|-h][-64]
```

オプション

オプション	説明
-v	エラーをコンソールに書き込みます。標準動作では、エラーを TVISION_HOME/logs/mqsensorbind.log に書き込みます。
-s	サイレント・モードが使用され、実行前にポップアップ・ウィンドウは表示されません。
-h	使用法のメッセージが表示されます。
-64	64 ビット・センサー・ライブラリを再バインドします。このオプションが表示されないときは、標準設定で 32 ビット・ライブラリが再バインドされます。

rebind_tux_sensor

保管場所

TVISION_HOME/bin/rebind_tux_sensor.sh

説明

このスクリプトは、TransactionVision BEA Tuxedo Sensor を AIX, Solaris, および HP-UX プラットフォームで再バインドします。

構文

```
rebind_tux_sensor.sh [-v|-s|-h]
```

オプション

オプション	説明
-v	エラーをコンソールに書き込みます。標準動作では、エラーを TVISION_HOME/logs/tuxsensorbind.log に書き込みます。
-s	サイレント・モードが使用され、実行前にポップアップ・ウィンドウは表示されません。
-h	使用法のメッセージが表示されます。

使用上の注意

BEA Tuxedo Sensor ライブラリは、このフル・パスを使用して BEA Tuxedo ライブラリにリンクされなければなりません。このリンクが実行されないと、TransactionVision Sensor が監視する BEA Tuxedo アプリケーションは失敗し、未解決のライブラリ依存または未解決のシンボルを示すエラーが表示されます。

よって、rebind_tux_sensor.sh スクリプトは、TransactionVision Sensor のインストール後、BEA Tuxedo がアップグレードされるか、またはそのインストールの場所が変更されるときに実行されなければなりません。

このスクリプトを実行する前に、TVISION_HOME および TUXDIR 環境変数を設定する必要があります。TUXDIR は、BEA Tuxedo Sensor のインストール・ディレクトリの完全絶対パスに設定する必要があります。このスクリプトにより、TVISION_HOME/lib または TVISION_HOME/lib64 の TransactionVision Sensor ライブラリが必要に応じて修正されます。

runSupportSnapshot

保管場所

< Diagnostics_probe_install_dir > /contrib/JASMutilities/snapins/
runSupportSnapshot.[sh|cmd]

注 : < TransactionVision_sensor_install_dir > は,
< Diagnostics_probe_install_dir > にあります。

説明

runSupportSnapshot ユーティリティは、Diagnostics または TransactionVision デプロイメント環境での Java エージェントの 1 つ以上のインスタンスのトラブルシューティングに関連する完全ファイル・セットを含む .zip ファイルを作成します。この .zip ファイルはカスタマ・サポート用です。

.zip ファイルには次のファイルが含まれます。

- ▶ < Diagnostics_probe_install_dir > %etc ディレクトリのファイル
- ▶ < Diagnostics_probe_install_dir > %log ディレクトリのファイル
- ▶ < TransactionVision_sensor_install_dir > %config ディレクトリのファイル
- ▶ < TransactionVision_sensor_install_dir > %logs ディレクトリのファイル
- ▶ Property Scanner レポート。修正されたエージェント・ディレクトリをリリース・バージョンと比較して、プロパティ・ファイル間、ポイント・ファイル間、また TransactionVision Sensors のみについては XML ファイル間の相違を報告します。
- ▶ プロパティ設定を含む Probe または Sensor インスタンス情報。1.5 JVM で実行されるエージェントの場合、環境変数、スタック・ダンプ、およびクラス・ローダ情報も含まれます。

注 : Diagnostics プローブが SSL をサポートするように設定されている場合、このユーティリティはインスタンス情報を収集できません。

基本構文

このユーティリティは、大部分のエージェント環境に適用される標準設定を使用します。したがって、一般的な構文は次のように単純です。

```
runSupportSnapshot.[sh|cmd] -console
```

完全構文

このユーティリティは、ユーザが標準設定をオーバーライドするオプションを指定すれば、一般的でないエージェント環境でも使用できます。このオプション構文は次のとおりです。

```
runSupportSnapshot.[sh|cmd] -console [-Zipfile] [-AddFiles] [-JAUser] [-JAPass] [-JAPort] [-JACount]
[-FTPSite] [-User] [-Pass] [-RemoteFile]
[-Source1] [-Source2] [-DiffOnly] [-SubDirs] [-Sort]
```

オプション構文のオプション

オプション	説明
-ZipFile	作成される .zip ファイル名です。標準設定 = HPCustomerSupport.zip
-AddFiles	.zip ファイルに含まれる追加のディレクトリです。ディレクトリのすべてのサブディレクトリを含めるには、ディレクトリ名に "+" を付け加えます。
-JAUser	エージェントが実行されているホストにアクセスするために使用されるユーザ名です。標準設定 = admin
-JAPass	対応するパスワードです。標準設定 = admin
-JAPort	Agent インスタンスのポーリングを起動するポートです。標準設定 = 35000
-JACount	そのデータが .zip ファイルに含まれるこのホストの Java Agent インスタンスの最大数です。標準設定 = 20
-FTPSite	FTP サイト名です。
-User	FTP サイトにアクセスするために使用されるユーザ名です。
-Pass	対応するパスワードです。

オプション	説明
-Remote File	リモート・ファイル名です。必要であれば、ファイル名にディレクトリ・パスの接頭辞を付けます。たとえば、 customer/diag/AgentSnapshot.zip のようにします。
-Source1	現在のエージェントがインストールされている etc ディレクトリです。標準設定 = < Java_probe_install_dir > ¥etc 。
-Source2	ディレクトリ内の用意済みの etc ディレクトリのアーカイブされたコピーです。標準設定 = エージェントがインストールされる内部ディレクトリ
-DiffOnly	違いのあるプロパティのみを表示します（「はい」または「いいえ」）。標準設定 = はい
-SubDirs	サブディレクトリも比較されるかどうかを指定します（「はい」または「いいえ」）。標準設定 = いいえ
-Sort	出力をプロパティ名でソートするかどうかを指定します（「はい」または「いいえ」）。標準設定 = いいえ

使用例

```
> cd C:¥MercuryDiagnostics¥JavaAgent¥DiagnosticsAgent¥contrib¥JASMUilities
¥Snapins
> java -Dcom.hp.javaagent.diagnostics.home="..¥..¥.." -jar
"..¥..¥..¥lib¥setupModule.jar"
-launchClass com.mercury.opal.javaprobe.setupModule.SetupModule -launchMethod
launchSetupModule -importJarList
probe.jar,org.mortbay.jetty-jdk1.2.jar,javax.servlet.jar,mail.jar,activation.jar
-importJarsFrom "..¥..¥..¥lib,lib" -customerSnapshot -console
INFORMATION-> [Looking for Java Agent Instances]: Looking for Java Agent
Instances...please wait...
INFORMATION-> [Java Agent found at http://localhost:35000/inst/
customerSupportSnapshot]: Java Agent Discovery
INFORMATION-> [Creating Probe Information file]: Java Agent Information
INFORMATION-> [Creating Configuration Information file]: Java Agent Configuration
Information
INFORMATION-> [Zip File Created]: Zip file
C:¥HPCCode¥diag_head¥javaprobe¥build¥contrib¥JASMUilities¥Snapins¥HPCCustom
erSupportFile.zip Created with length 159215
```

.zip ファイルをカスタマ・サポートに送信する前に、そこに機密データが含まれていないか必要に応じて確認します。

ServicesManager

保管場所

TVISION_HOME/bin/ServicesManager.[sh|bat]

目的

TransactionVision Analyzer サービスを管理します。Analyzer は、埋め込まれた RMI レジストリを使用して、リモート・ホストで実行される TransactionVision Web ユーザ・インタフェースで管理可能になります。

Analyzer が ServicesManager ユーティリティの `-keepcollect` オプションを使用して停止または終了される場合、Sensor を完全に停止する唯一の方法は、Analyzer とプロジェクトを起動してから、プロジェクトを通常どおり停止することです。さもないと Sensor は、Analyzer によって送信される最後の構成メッセージが期限切れになるまで、イベント・キューに収集されるイベントの生成を続行します。『**Using TransactionVision**』（英語版）の「**Configuration Message Expiry**」を参照してください。

構文

```
ServicesManager
{(-start [-project (-proj) PROJECTNAME]) |
 (-stop [-quiesce][[-project (-proj) PROJECTNAME]
 [-keepcollect]) |
 (-exit [-quiesce][[-keepcollect]) |
 (-status [-project (-proj) PROJECTNAME]) |
 (-killserver) |
 (-reconfig {classification | logging | analyzer}) |
 (-versioninfo)
 ([-host HOST][[-rmiregp PORTNUMBER][[-debug])
```

オプション

オプション	説明
-start	まだ実行されていないければ、Analyzer プロセスを起動します。
-stop	Analyzer はイベント・データの収集を停止します。
-exit	Analyzer はイベント・データの収集を停止し、プロセスは終了します。

オプション	説明
-quiesce	停止または終了した Analyzer の標準動作は、収集の即時停止です。このフラグが設定されると、Analyzer は、停止または終了の前にイベント・キューで保留中のイベントをクリアします。大量のイベント・バックログがあると、Analyzer の停止または終了に時間がかかることがあります。
-keepcollect	Sensor に停止メッセージを送信しません。-stop または -exit オプションと組み合わせて使用されると、センサーが収集を続行します。
-status	現在の Analyzer のステータスを報告します。
-reconfig	<p>イベントの収集を停止せずに、Analyzer の設定（分類、ログ記録、すべての設定）を再ロードします。</p> <ul style="list-style-type: none"> ▶ logging – ディレクトリ <TVISION_HOME>/config/logging にある log4j XML 設定ファイルを再ロードします。 ▶ analyzer – Analyzer.properties ファイル、Beans.xml ファイルに記述されるすべてのビーン、再初期化ロギングファイル、ならびにすべての XML ベースのルール・ファイルの大部分の Analyzer 設定値を再ロードして、Analyzer を再初期化します。 <p>Analyzer.properties ファイルの [Collection Properties] セクションの設定値は、write_to_buffer_table property を除いて、-reconfig オプションを使用して Analyzer によって再ロードされません。</p>
-versioninfo	Analyzer のバージョン情報を返します。
-killserver	Analyzer を即時シャットダウンします。このフラグの設定は推奨されません。Analyzer をエラーなくシャットダウンするには、-exit の使用をお勧めします。
-project (-proj) PROJECTNAME	<p>指定されたコマンドのプロジェクト名。プロジェクト名にスペースを含めるときは、プロジェクト名を（たとえば -proj "Project Name" のように）二重引用符で囲んでください。</p> <p>Analyzer がイベントを処理するためには、プロジェクトに通信リンクが必要です。-start、-stop、および -status コマンドのみ単一のプロジェクトで実行できます。このオプションは、-host または -rmiregp オプションと組み合わせて使用することはできません。-project オプションを使用するときは、Analyzer のホストとポートはデータベースから検索されます。</p>

オプション	説明
-host HOST	Analyzer が実行されるホスト名。名前でも IP アドレスでも指定できます。指定されないと、ローカル・ホストが使用されます。
-rmiregp PORTNUMBER	RMI 接続で Analyzer がリッスンするポート番号。標準設定値は、Analyzer.properties ファイルの analyzer_port property で指定される値です。このオプションは、すでに実行されている Analyzer、すなわち起動時に必ず Analyzer.properties で指定される値を使用する Analyzer と通信する場合にかぎり有効となります。
-debug	Analyzer プロセスをデバッグ・モードで起動します。

使用例

- ▶ プロジェクト PROJECT を起動します。
ServicesManager -start -proj PROJECT
- ▶ ホスト HOST 上の Analyzer を停止します。
ServicesManager -stop -host HOST

SetupModule

保管場所

< java_agent_install_dir > ¥DiagnosticsAgent¥lib

この < java_agent_install_dir > は、Java Agent のインストール先のディレクトリのパスです。標準設定のパスは、Windows では C:¥MercuryDiagnostics¥JavaAgent, UNIX では ¥opt¥MercuryDiagnostics¥JavaAgent です。

説明

このスクリプトは、Java Agent セットアップ・モジュールを起動します。

構文

```
SetupModule -recordFile name.rec -installFile name.rec -console
```

オプション

オプション	説明
-recordFile name.rec	レコード
-installFile name.rec	記録を再生するか自動化します。
-console	コンソール・モードで起動します。

SQLServerTest

保管場所

com.bristol.tvision.admin.SQLServerTest

説明

SQL Server データベースの INSERT パフォーマンスを測定します。

このユーティリティはサンプル・イベント・データを指定されたスキーマの RAW_EVENT テーブルに挿入します。テストの実行前に、CreateSqlScript（テスト実行後に削除できます）を使用して新規プロジェクト・スキーマを作成します。

構文

```
java com.bristol.tvision.admin.SQLServerTest databaseName user passwd schema
eventCount eventSize threadCount {-commit n}{-jdbcBatch}
```

オプション

オプション	説明
-databaseName	テストに使用されるスキーマを含む SQL Server データベースの名前
-user	SQL Server ユーザ名
-passwd	SQL Server パスワード
-schema	サンプル・イベント・データが保存される TransactionVision スキーマ
-eventCount	生成されるイベントの数
-threadCount	イベントの生成に使用されるスレッドの数
-commit n	単一のバッチ操作の一部として n 回挿入ステートメントを実行します。標準設定では、各挿入を個別にコミットします。
-jdbcBatch	JDBC 標準バッチ操作を使用します。

TVisionSetupInfo

保管場所

TVISION_HOME/bin/TVisionSetupInfo.[sh|bat]

目的

TransactionVision の実行前に CLASSPATH およびライブラリ・パスが正しく設定できるように、TransactionVision が使用する外部ツールに関する情報を管理者から収集します。このユーティリティは、次の操作を行います。

- ▶ ソフトウェア・ツールのインストール・パスを TVISION_HOME/config/setup/DefaultInstallPath.xml に保存します。
- ▶ ユーザ入力に基づき、TVISION_HOME/config/datamgr/Database.properties を修正します。
- ▶ ユーザ入力に基づき、TVISION_HOME/config/ldap/Ldap.properties を修正します。
- ▶ TVision に必要な環境変数の最小セット (JAVA_HOME, CLASSPATH, さまざまなプラットフォームの共有ライブラリ・パスなど) を設定する TVISION_HOME/bin/SetupEnv.[sh|bat] を生成します。

構文

```
TVisionSetupInfo.[sh|bat] - cleanweb
```

オプション

オプション	説明
-cleanweb	TransactionVision Web アプリケーションを削除します。

注意事項

TVISION_HOME/config/setup/Setup.properties は、標準設定のソフトウェア・インストール・パスとセットアップ・ロギング・ファイルを保存する XML ファイルの指定に使用されます。

ユーザは、修正および生成されたファイルを後で自在に修正できます。

このユーティリティは TransactionVision の設定ファイルを修正するので、ユーザにはこのユーティリティを実行するファイル修正権限 (Unix ではルート、Windows では管理者) が必要です。

TVISION_HOME/config/setup/DefaultInstallPath.xml のフォーマットは次のとおりです。

```
<?xml version="1.0"?>
<DefaultInstallPath>
  <OS name="Windows">
    <DB2>C:\Program Files\IBM\sqllib</DB2>
    <WebSphereMQ>C:\Program Files\IBM\WebSphere MQ</WebSphereMQ>
    <WebSphereMQJava>C:\Program Files\IBM\WebSphere MQ\Java
  </WebSphereMQJava>
    <WebSphereAppServer>C:\Program Files\Websphere\AppServer
  </WebSphereAppServer>
    <WebLogic>C:\bea\weblogic81</WebLogic>
  </OS>
  <OS name="SunOS">
    <DB2>/opt/IBMdb2/V7.1</DB2>
    <Oracle></Oracle>
    <WebSphereMQ>/opt/mqm</WebSphereMQ>
    <WebSphereMQJava>/opt/mqm/java</WebSphereMQJava>
    <WebSphereAppServer>/opt/WebSphere/AppServer </WebSphereAppServer>
  <WebLogic>/usr/local/BEA/weblogic81</WebLogic>
  </OS>
  <OS name="AIX">
    <DB2>/usr/lpp/db2_07_01</DB2>
    <Oracle></Oracle>
    <WebSphereMQ>/usr/lpp/mqm</WebSphereMQ>
    <WebSphereMQJava>/usr/lpp/mqm/java</WebSphereMQJava>
    <WebSphereAppServer>/usr/WebSphere/AppServer </WebSphereAppServer>
  <WebLogic></WebLogic>
  </OS>
  <OS name="Linux">
    <DB2>/usr/IBMdb2/V7.1</DB2>
    <Oracle></Oracle>
    <WebSphereMQ>/opt/mqm</WebSphereMQ>
    <WebSphereMQJava>/opt/mqm/java</WebSphereMQJava>
    <WebSphereAppServer>/opt/WebSphere/AppServer </WebSphereAppServer>
    <WebLogic></WebLogic>
  </OS>
</DefaultInstallPath>
```


付録 B

設定ファイル

TransactionVision セットアップ・ユーティリティは、Analyzer の設定情報を以下の設定ファイルに保存します。設定を変更する必要がある場合は、これらのファイルに直接変更を加えることもできます。

本章の内容

- ▶ Analyzer.properties (362 ページ)
- ▶ CacheSize.properties (368 ページ)
- ▶ Database.properties (368 ページ)
- ▶ JobManager.properties (372 ページ)
- ▶ Sensor.properties (372 ページ)
- ▶ SensorConfiguration.xml (372 ページ)
- ▶ Setup.properties (374 ページ)
- ▶ StatisticsCache.properties (375 ページ)
- ▶ UI.properties (376 ページ)

注：プロパティ・ファイルに変更を加えた場合は、その変更を有効にするために、関連付けられたアプリケーションを再起動する必要があります。再起動については、84 ページ「Analyzer の管理」を参照してください。

Analyzer.properties

< **TVISION_HOME** > /config/services/**Analyzer.properties** ファイルは、TransactionVision Analyzer の一般、収集、および分析に関する設定情報を提供します。

一般プロパティ

- ▶ **rmi_client_timeout** : このエント리는, RMI クライアントのタイムアウトを分単位で指定します。標準設定の値は 2 です。
- ▶ **logging_xml** : Analyzer が使用するログ設定ファイルの名前を指定します。標準の値は Analyzer.Logging.xml です。
- ▶ **trace** : このプロパティは, Analyzer のトレース・ログのオン/オフを指定します。標準設定の値は ON です。
- ▶ **debug** : このプロパティは, Analyzer のデバッグ・ログのオン/オフを指定します。標準設定の値は OFF です。この設定は, サポート担当者からそうするように助言された場合にのみ有効にしてください。
- ▶ **service_jvm_flags** : Analyzer の実行に使用する追加の JFM フラグを指定します。
- ▶ **service_additional_classpath** : Analyzer とともに使用するカスタム・ビーンのクラスパスを指定します。
- ▶ **jvm_dll service_classpath service_libpath** : Windows では, これらのプロパティは TVisionSetupInfo によって自動的に生成されます。これらを変更する必要はないはずですが。変更する場合は, TVisionSetupInfo が再び実行されるとこれらのプロパティがリセットされる点に注意してください。これらのプロパティは, Windows サービスとして実行される Analyzer 以外の TransactionVision 環境には影響を与えません。
- ▶ **analyzer_port** : このエント리는, Analyzer の実行で使用する標準設定のポート番号を指定します。標準設定の値は 21100 です。Analyzer は, 起動時に常にこのポート番号を使用します。すでに実行中の Analyzer については, ServicesManager.[bat|sh] -rmiregp コマンドを使ってポート番号を設定することができます。

- ▶ **rmi_server_port** : このエント리는, RMI サーバ・オブジェクトが標準設定でリッスンするポートの番号を指定します。

標準設定の値は 0 で, Analyzer の起動時に Java ランタイムが任意のランダムなポートを使用することを意味します。ファイアウォールの背後で Analyzer を実行するため固定ポートが必要な場合は, このプロパティを特定のポート番号に指定します。

- ▶ **time_server** : Analyzer 内でタイム・サーバを実行する場合は on に設定します。タイム・サーバは, TIBCO EMS および汎用 JMS 通信リンク・タイプにとって, タイム・スキュー情報を計算するために必要です。別のホストでタイム・サーバを実行する場合は, TimeServer.[sh|bat] ユーティリティを使用してください。
- ▶ **time_server_port** : time_server が on に設定されている場合, タイム・サーバで使用するリッスン・ポートを指定します。標準設定のポートは 9037 です。

収集プロパティ

Analyzer のイベント収集操作には, 以下のプロパティが使用されます。

- ▶ **batch_commit_count** : いくつかのイベントをバッチにまとめてコミットするかを指定します。標準設定の値は 50 です。
- ▶ **commit_time_threshold** : コミットが強制されるまでの時間を秒数で指定します。標準設定の値は 1 です。
- ▶ **commit_byte_threshold** : コミットが強制されるまでのバイト数を指定します。標準設定の値は 1,000,000 です。
- ▶ **write_to_buffer_table** : 通常の処理に加えて, キューからの未処理イベントのコピーを RAW_EVENT テーブルに書き込むかどうかを指定します。標準設定の値は False です。この設定は, UI の [通信リンク] ページでの対応する設定とは異なり, すべてのプロジェクトにグローバルに適用されます ([通信リンク] ページの場合は, 特定の通信リンクについて未処理イベントの書き込みを有効にできます)。read_from_buffer_table が True に設定されている場合は, このプロパティを True に設定してはなりません。このプロパティを使用するのは, カスタム・ビーンをテストする場合や, 問題をトラブルシューティングするために使用するようにサポート担当者から助言された場合です。実運用環境では, 常に False に設定してください。

- ▶ **read_from_buffer_table** : キューからの読み取りに加えて、RAW_EVENT テーブルからイベントをプルするかどうかを指定します。標準設定の値は False です。write_to_buffer_table が True に設定されている場合は、このプロパティを True に設定してはなりません。このプロパティを使用するのは、カスタム・ビーンをテストする場合や、問題をトラブルシューティングするために使用するようサポート担当者から助言された場合です。実運用環境では、常に False に設定してください。
- ▶ **keep_events** : 未処理のイベントが正常に処理された後で Analyzer で未処理イベントを削除するかどうかを指定します。未処理イベントを RAW_EVENT テーブル内に残す場合は、このプロパティを True に設定します。残されたイベントの event_status カラムは PROCESSED に設定されるので、それらのイベントがもう一度処理されることはありません。それらのイベントをもう一度処理する場合は（たとえば、別のタイプの分析をするためにカスタム Java ビーンを使って処理する場合など）、まず event_status を NEW に設定する必要があります。Analyzer が未処理のイベントを処理した後でそれらのイベントが RAW_EVENT テーブルから削除されるようにするには、このプロパティを False（標準設定値）に設定します。
- ▶ **write_to_jar** : write_to_buffer_table に似ていますが、Analyzer は未処理イベントをデータベース内のテーブルではなく JAR ファイルに保存します。この JAR ファイルは \$TVISION_HOME/logs にあり、そのファイル名は **SCHEMA_raw_events.jar** です。各プロジェクト・スキーマについて、それぞれ個別の JAR ファイルが作成されます。JAR ファイルを正しく閉じるためには、プロジェクトを停止する必要があります。
- ▶ **read_from_jar** : read_from_buffer_table に似ていますが、Analyzer は未処理イベントをデータベース内のテーブルではなく JAR ファイルから読み取ります。データベース・スキーマが SCHEMA のプロジェクトの場合、Analyzer は < TVISION_HOME > /logs ディレクトリ内の **SCHEMA_raw_events.jar** ファイルを検索します。Analyzer の実行中に JAR ファイルをコピーできます。Analyzer は、対応するファイルを探して定期的にディレクトリ内をスキャンします。Analyzer が JAR ファイル内のイベントの処理を開始すると、そのファイルは **SCHEMA_raw_events.jar.processed.TIMESTAMP** という名前に変更されます。write_to_jar が True に設定されている場合は、このプロパティを True に設定してはなりません。
- ▶ **fail_safe_collection_shutdown** : True に設定すると、イベントの収集と処理で重大な障害が発生したときに Analyzer が構成メッセージの送信を停止するようになります。標準設定の値は true です。

- ▶ **jdbc_batching** : jdbc_batching の値が標準設定 (on) になっている場合、TransactionVision はデータベース・ステートメントをバッチ・モードで実行します。
- ▶ **jdbc_batch_count** : JDBC バッチ・モードでバッチ処理する SQL 操作の数を指定します。この数は、batch_commit_count property と等しいか、その倍数でなければなりません。標準設定の値は 50 です。

分析プロパティ

Analyzer のイベント分析操作には、以下のプロパティが使用されます。

- ▶ **save_event_document** : イベント・ドキュメントをデータベースに保存するかどうかを指定します。標準設定の値は on です。このプロパティを off に設定すると、イベントの詳細は提供されません。さらに、非同期的にフラッシュされたトランザクション・データおよび静的トポロジ・データを Analyzer が復元できません。これは、プログラムのクラッシュやほかの異常終了が発生した場合に、静的トポロジ・ビューのトランザクション・データと統計データが不正確になる可能性があることを意味します。
- ▶ **eventmatching_interval** : イベント・マッチングの呼び出しの間隔を秒数で指定します。標準設定の値は 600 です。
- ▶ **partial_event_lifetime** : 部分的なイベント・エントリが partial_event テーブル内に存在できる最大時間 (分)。この制限時間に達すると、その部分イベントはフラッシュされます。標準設定の値は 10 分です。
- ▶ **latency_resolution** : イベント間のレイテンシの計算に使用される分解能。次の値が使用可能です。

値	説明
1	秒
10	1/10 秒
100	1/100 秒 (標準設定)
1000	ミリ秒

- ▶ **generate_api_only_txns** : Analyzer は標準設定では、データ収集フィルタ「API only」で収集されたイベントに対しては、トランザクション分析を実行しません。この動作を変更するには、このプロパティを true に設定します。

- ▶ **correlation_limit** : この設定は、特定の相関キーについて Analyzer によって作成されるイベントの関係の数を制限します。このプロパティは、パフォーマンスの問題につながるカスタム相関における相関ルール設計の欠陥に対する安全手段として利用できます。標準設定の値は -1 (制限なし) です。
- ▶ **local_txn_limit** : この設定は、Analyzer によって 1 つのビジネス・トランザクションに割り当てられるローカル・トランザクションの数を制限します。このプロパティは、パフォーマンスの問題につながるカスタム・ローカル・トランザクション分析におけるローカル・トランザクション・ルール設計の欠陥に対する安全手段として利用できます。標準設定の値は -1 (制限なし) です。
- ▶ **separate_child_thread_txns** : 標準設定では、サブレットが JMS 呼び出しを行うためにスレッドを生成する場合、サブレットは追跡情報を子スレッドに渡します。その結果、サブレット・イベントと JMS イベントは両方とも同じビジネス・トランザクションに属することになります。ただし、これらのイベントを別々のトランザクションに分ける必要がある場合もあります。たとえば、サブレットと同じトランザクションの一部にたくない長時間実行されているスレッドを、サブレットが生成することもできます。サブレットと同じビジネス・トランザクションに含められるスレッドがサブレットによって生成されないようにするには、このプロパティの値としてサブレット・プログラム名をリストします。複数のプログラム名を入力する場合は、次の例のようにカンマで区切ります。

```
separate_child_thread_txns=program1, program2
```

- ▶ **enable_dbcaching** : 失敗モードに加えて、または失敗モードの代わりに、標準モードを有効にします。すべてのプロジェクトが失敗モードしか使用しない場合は、システム・リソースを節約するために標準モードを無効にしてもかまいません。
- ▶ **enable_failure_mode** : 失敗したビジネス・トランザクション (または SLA に違反したトランザクション) のイベント・データだけを保存するために、失敗モードを有効にします。成功したビジネス・トランザクションについては、対応するビジネス・トランザクション行だけが保存されます。このモードは、ビジネス・トランザクションの「result」属性の有意の値に大きく依存するため、「result」カラムに関する適切なルールを持つトランザクションのクラス化が必要です。失敗モードは各プロジェクトごとに有効にすることができるので、特定のプロジェクトを失敗モードで実行する一方で、別のプロジェクトを従来の標準モードで実行することもできます。標準設定の値は off です。失敗モードの詳細については、『Using TransactionVision』(英語版) の 74 ページ「Failure Mode」を参照してください。

- ▶ **dbcache_thread_count** : キャッシュされた分析データをデータベースに書き込むために使用するスレッドの数を指定します。最適の値は **Analyzer** を実行しているハードウェアによって異なり、さまざまな値でパフォーマンスを測定することでしか決定できません。ただし、開始点としては、特定のスキーマに使用されている収集スレッドの半分の数に設定してみるのがいいでしょう。標準設定の値は 2 です。
- ▶ **failure_mode_thread_count** : 失敗モードに使用するフラッシュ・スレッドの数を定義します。この数は、キャッシュ・データを十分高速でフラッシュできる程度に多く、システム・リソースを多く浪費しない程度に少ない数に設定しなければなりません。標準設定の値は 2 です。
- ▶ **failure_mode_process_delay** : エイジアウトのタイムアウトを決定するために SLA 値に追加される時間をミリ秒数で定義します。これは、失敗モードにとって不可欠な値です。この値を小さく設定しすぎると、多くの成功したトランザクションのデータが不必要にエイジアウトしてデータベースに書き込まれるため、パフォーマンスが低下し、データベース・ディスク・スペースが浪費されます。この値を大きく設定しすぎると、メモリ・キャッシュに保持されるデータが多くなりすぎます。キャッシュ・サイズがシステム負荷に十分対応できるだけ大きくないと、キャッシュ・オーバーフローが頻繁に発生し、パフォーマンスが大幅に低下します。標準設定の値は 5 秒です。
- ▶ **failure_mode_discard_overflow** : キャッシュ・オーバーフローが発生すると、キャッシュ内のすべてのデータが正常にデータベースに書き込まれます。何かの理由でオーバーフローが発生すると、システムの処理速度の低下が長く続き、定期的なキャッシュのフラッシュがさらに状況を悪化させて、**Analyzer** の動作がさらに遅くなる可能性があります。このような状況でイベント・キューがいっぱいになるのを防ぐことが主な目的の場合は、このパラメータを **true** に設定すれば、オーバーフロー発生時にすべてのキャッシュ・データが破棄されるようになります。標準設定の値は **false** です。

CacheSize.properties

< **TVISION_HOME** > **/config/services/CacheSize.properties** ファイルは、イベント分析で使用されるキャッシュのサイズを指定します。キャッシュ・サイズを大きくすると、データベース・アクセスが最小限になってパフォーマンスが向上しますが、TransactionVision が使用するメモリの量も増大します。以下のプロパティが設定されます。

- ▶ **system_model_objects** : キャッシュされる PII 以外のシステム・モデル・オブジェクトの数を指定します。
- ▶ **pii_objects** : キャッシュされる PII システム・モデル・オブジェクトの数を指定します。
- ▶ **event_based** : キャッシュされるイベントの数を指定します。
- ▶ **transaction_based** : キャッシュされるトランザクションの数を指定します。

Database.properties

< **TVISION_HOME** > **/config/datamgr/Database.properties** ファイルは、Analyzer と TransactionVision UI/Job サーバが読み取りと書き込みに使用するデータベースを指定します。

必須エントリ

以下の必須エントリが必要です。

- ▶ **jdbc_driver=COM.ibm.db2.jdbc.app.DB2Driver** : このエントリは、使用される JDBC ドライバのクラス名です。IBM DB2 JDBC の場合の標準設定は、上記のとおりです。Oracle の場合の標準設定は、`oracle.jdbc.driver.OracleDriver` です。SQL Server の場合の標準設定は、`com.microsoft.jdbc.sqlserver.SQLServerDriver` です。
- ▶ **database_connection_name** : DB2 の場合、このエントリは使用されるデータベース接続の名前（通常はデータベース別名）です。Oracle の場合、このエントリは `oci` クライアント接続のデータベース名か、シンクライアント接続の SID（システム ID）です。SQL Server では、このエントリは使用されません。

- ▶ **database_name** : DB2 の場合、このエント리는接続先のサーバ上にあるデータベースの名前です。クライアント・データベース接続が使用されている場合は、この名前が `database_connection_name` と異なることがあります。Oracle の場合、このエント리는 SID です。SQL Server の場合、このエント리는データベースの名前です。
- ▶ **database_host** : このエント리는、データベース・サーバが実行されているホストです。
- ▶ **database_port** : Oracle の場合、このエント리는ターゲット・ホスト上の Oracle リスナのポート番号を指定します。標準設定の値は 1521 です。この属性が使用されるのは、Oracle シンクライアントを使用している場合だけです。TransactionVision とともに Oracle を使用する方法については、第 6 章「データベースの設定」を参照してください。SQL Server の場合、このエント리는サーバ上の接続先ポートを指定します。標準設定の値は 1433.21 です。
- ▶ **oracle_client_type** : Oracle thin またはクライアント JDBC ドライバのどちらかを使用するかを指定します。thin (推奨) または oci for Oracle 9 または oci8 for Oracle 8.1.7 に設定してください。標準設定の値は thin です。この属性が使用されるのは、Oracle データベースを使用している場合だけです。TransactionVision とともに Oracle を使用する方法については、第 6 章「データベースの設定」を参照してください。
- ▶ **oracle_user_password** : このエント리는、新しいプロジェクト・スキーマを作成された Oracle ユーザ用に設定されるユーザ・パスワードを指定します。標準設定の値は ABC99DEF です。特定のパスワード・ポリシーに従う必要がある場合は、この値を変更できます。
- ▶ **db2_instance_env** : このエント리는、DB2 環境変数 DB2INSTANCE の値を指定します。

自動検出をオーバーライドする省略可能なエントリ

このファイルでは、以下の省略可能なエント리를設定できます。これらのエント리는、自動検出をオーバーライドします。

- ▶ **connection_type** : このエント리는、JDBC 接続を取得する方法を定義します。次のいずれかの値を使用します。
 - ▶ JDBC : JDBC ドライバから取得した接続をドライバ接続プールなしで使用します (標準設定)。
 - ▶ DB2DataSource : DB2 接続プールを使用します。

- ▶ **OracleDataSource** : Oracle 接続プールを使用します。
- ▶ **JNDI** : JNDI に登録されたデータ・ソースを使用します。JNDI 接続を使用するには、`jndi_url` プロパティも設定する必要があります。
- ▶ **database_url** : このエントリは、JDBC ドライバ・マネージャで使用するカスタム URL を定義します。
- ▶ **jndi_url** : このエントリは、JNDI 接続を使用する場合のデータベースの JNDI 名 (WebSphere で定義されたもの) を定義します。
- ▶ **unicode_db** : このプロパティが設定されると、属性 `unicode=true` が設定された文字ベースの XDM カラムはすべて、指定された数の文字をデータベースに保存できるようにするために、バイト・サイズの 2 倍で生成されます。
- ▶ **unicode_bytes_per_character** : このプロパティが設定されると、属性 `unicode=true` が設定された文字ベースの XDM カラムはすべて、この値を基本値の乗数として使用して生成されます。標準設定値の 2 は、`unicode_db=true` を設定してこの値を設定しないという動作と一致します。3 より大きい値は、データベース作成の問題を引き起こす場合があります。このプロパティは、`unicode_db=true` が設定されていないと効果がありません。
- ▶ **jdbc_url** : このエントリは、標準設定の DB2 ユニバーサル JDBC ドライバ (タイプ 4) または Oracle シンククライアント・ドライバ以外の、使用するデータベース・ドライバを指定します。

DB2 ユニバーサル・ドライバ (タイプ 2) の場合、このエントリの形式は次のようになります。

```
jdbc:db2: < database-name >
```

< database-name > は、Analyzer の接続先のデータベースです。

Oracle oci ドライバの場合、このエントリの形式は次のようになります。

```
jdbc:oracle:oci: < user > / < password > @ < database-name >
```

< database-name > は Analyzer の接続先のデータベースで、< user > と < password > は Oracle 接続に必要なユーザ名とパスワードです。

標準設定値を持つ省略可能なエントリ

このファイルでは、以下の省略可能なエントリを設定できます。これらを指定しなかった場合は、標準設定値が使用されます。

- ▶ **user** : この省略可能なエントリは、データベース接続を確立するときに使用されるユーザ名です。このフィールドが空の場合は、現在ログインしているユーザを使ってデータベース接続が確立されます。指定したユーザまたは現在ログインしているユーザが、データベースにアクセスする権限を持っていることを確認してください。
- ▶ **passwd** : この省略可能なエントリは、データベース接続を確立するときに使用されるパスワードです。このフィールドが空の場合は、現在ログインしているユーザのパスワードを使ってデータベース接続が確立されます。
- ▶ **jce_provider** : この省略可能なエントリは、データベース・パスワードを暗号化するための Java Cryptographic Extension (JCE) パッケージの名前を指定します。次の表は、有効なパッケージ名を示しています。このエントリが空白の場合、TransactionVision はパスワードをプレーン・テキストで保存します。

プロバイダ	パッケージ名
Sun	com.sun.crypto.provider.SunJCE
IBM	com.ibm.crypto.provider.IBMJCE

TVisionSetupInfo ユーティリティは、DES 暗号化アルゴリズムをサポートするインストール済みの JCE プロバイダを自動的に検索します。そのようなプロバイダが見つかったら、TVisionSetupInfo は `jce_provider` の値をその JCE プロバイダのクラス名に設定します。JCE プロバイダが見つからなかった場合、TVisionSetupInfo は次のメッセージを表示します。

```
Java Cryptography Extension (JCE) is not present in your current JDK.
Password encryption feature will be disabled and stored in plain text.
```

- ▶ **reconnect_interval** : このエントリは、TransactionVision がデータベースへの再接続を試みる頻度を秒数で指定します。標準設定の値は 10 です。
- ▶ **reconnect_timeout** : このエントリは、TransactionVision が (`reconnect_interval` で指定された間隔で) データベースへの接続を試みる時間の長さを秒数で指定します。標準設定の値は 600 です。
- ▶ **query_timeout** : このフィールドは、UI 内で実行されるクエリについて、タイムアウトを秒数で指定します。標準設定では、この設定はオフになっています。

JobManager.properties

< TVISION_HOME > /config/jobs/JobManager.properties ファイルは、TransactionVision ジョブの設定情報を指定します。このファイルは以下のプロパティを指定します。

- ▶ **BeansXMLFile** : 新規のプロジェクトを作成するときに利用できるジョブ・テンプレートが収められた設定ファイルの場所を指定します。
- ▶ **logSize**。ログ・ファイル内に保持するエントリの数を指定します。

Sensor.properties

< TVISION_HOME > /config/sensor/Sensor.properties ファイルは、Servlet Sensor および JMS Sensor に関する情報を指定します。このファイルには以下のエントリがあります。

- ▶ **logging_xml** : ログ設定ファイルの名前を指定します。
- ▶ **configuration_file** : Sensor 設定 XML ファイルのパス名を指定します。

SensorConfiguration.xml

SensorConfiguration.xml ファイルは、Sensor と構成キューの間でのやり取りを定義します。このファイルは以下の属性を定義しますが、これらは通信リンクを作成または編集するときに指定されます。

属性	説明
ConfigurationQM	構成キュー・マネージャの名前
ConfigurationQ	構成キューの名前
ConfigurationQMHost	構成キュー・マネージャのホスト名
ConfigurationQMPort	構成キュー・マネージャのリスナ・ポート番号
ConfigurationQMChannel	構成キュー・マネージャのチャンネル名

属性	説明
ConnectionRetryDelay	構成キュー・マネージャへの接続が失われた場合の、接続の試行間の遅延（単位はミリ秒）。標準設定は 1000 です。
ConnectionRetryTimeout	接続が失われた場合に構成キュー・マネージャへの再接続を試みる時間の長さ（単位はミリ秒）。値として -1（標準設定）を指定すると、永遠に再試行します。
ConfigurationRetrieveInterval	新しい構成メッセージがないか確かめるために構成キューをチェックする間隔（単位はミリ秒）。標準設定の値は 10000 です。
SensorClientTimeSkewInterval	Sensor を実行しているホストと構成キュー・マネージャを実行しているホストからのタイム・スキューをチェックする間隔（単位はミリ秒）。標準設定の値は 300000（5 分）です。
EventPackageFlushTimeout	イベント・パッケージがどれだけの時間アイドル状態になっていたら強制的にイベント・キューに書き込まれるかを指定します（単位はミリ秒）。最小値は 10000 で、標準設定の値は 300000（5 分）です。
RepeatLogInterval	通常は抑止される繰り返されるエラーを繰り返す時間の長さ（単位はミリ秒）。標準設定の値は 600000 です。

Setup.properties

< **TVISION_HOME** > /config/setup/Setup.properties ファイルは、以下のプロパティを指定します。

- ▶ **default_tool_install_path** : DefaultInstallPath.xml ファイルのディレクトリ場所。このファイルには、TransactionVision が必要とするソフトウェア・コンポーネントの場所がリストされています。
- ▶ **logs_dir** : ログ・ファイルを保存するディレクトリの名前。
- ▶ **logging_xml** : ログ設定ファイルの名前。
- ▶ **minimum_java_version** : TransactionVision が必要とする最低限の Java のバージョン。
- ▶ **minimum_java_version_sun** : Solaris プラットフォーム上の TransactionVision が必要とする最低限の Java のバージョン。
- ▶ **maximum_java_version** : TransactionVision がサポートする最新の Java バージョン。

StatisticsCache.properties

< TVISION_HOME > /config/services/StatisticsCache.properties ファイルは、コンポーネント・トポロジ分析の静的モードで使用される設定を定義します。このファイルには以下のエントリがあります。

- ▶ **flush_interval** : 統計キャッシュがデータベースに書き出される頻度を指定します。データは、データベースに書き込まれるまではグラフに表示されません。受信される要求の量に応じて、フラッシュの間隔を長くしたり短くすると役立つ場合があります。フラッシュの間隔を長くするとデータベースへの書き込みの回数が減りますが、キャッシュのサイズが大きくなります。
- ▶ **timeslice_interval** : イベント統計情報を計算する対象となるタイムスライスの継続時間（単位は分）を指定します。タイムスライスの概念は、データをパーティションに分割して、興味の対象がどの時間枠であるかに応じてそのデータをすばやく取り出すために使用されています。1つのタイムスライス（一定の時間）の中で発生した同じプログラムに属するすべてのイベントと MQ オブジェクトは、1つの統計情報にまとめられます。たとえば、メッセージを特定のキューに入れるプログラムがある場合、特定のタイムスライスの間に発生したそのプログラムとキューに関連するすべての統計データ（レイテンシ時間、成功カウントなど）は、同じデータベース行に保存されます。タイムスライス間隔の値を変更することによって、統計テーブルへのアクセスの規模と効率を制御できます。タイムスライスを大きくするほど、結果の保存は効率的になります。その代わりに、タイムスライスはデータを見る際の最小の増分単位であるという問題があります。たとえば、タイムスライスを1日に設定した場合、それらの統計情報を1時間刻みで見ることができません。

UI.properties

< **TVISION_HOME** > /config/ui/UI.properties ファイルは、TransactionVision の Web ユーザ・インタフェースが使用するプロパティを指定します。このファイルには以下のエントリがあります。

- ▶ **ui_config_dir** : ユーザ・インタフェースの設定ファイルの場所を指定します。
- ▶ **view_config_filename** : TransactionVision のビューを定義するファイルを指定します。
- ▶ **reports_xml_filename** : 使用可能なレポートがリストされているファイルの場所を指定します。
- ▶ **category_xml_filename** : クエリ設定ファイルの場所を指定します。
- ▶ **category_xsl_filename** : クエリ・スタイル・シートの場所を指定します。
- ▶ **filter_config_xml** : データ収集フィルタ設定ファイルの場所を指定します。
- ▶ **filter_config_xsl** : データ収集フィルタスタイル・シートの場所を指定します。
- ▶ **detail_modifier_xsl** : イベント詳細ビューのスタイル・シートの場所を指定します。
- ▶ **logging_xml** : ユーザ・インタフェースのログ・ファイルの場所を指定します。
- ▶ **support_url** : 製品サポートの URL。
- ▶ **autologin** : TransactionVision ログイン・ページの [Remember Login] チェック・ボックスを有効にします。このチェック・ボックスを有効にすると、ユーザ名とパスワードがクッキーとして保存され、そのユーザは次に TransactionVision にアクセスするときからユーザ名とパスワードの入力を求められなくなります。パスワードは、指定された Java JCE プロバイダを使って暗号化されます。
- ▶ **jceProvider** : autologin プロパティが true の場合にパスワードの暗号化に使用される JCE パッケージの名前を指定します。JCE パッケージがない場合は、java.sun.com からダウンロードできます。
- ▶ **trace** : TransactionVision の Web ユーザ・インタフェースについてトレース・ログを有効にします。詳細については、104 ページ「Windows と UNIX のシステム・ログの使用」を参照してください。
- ▶ **logout_Redirect** : このプロパティは、ユーザが TransactionVision からログアウトしたときに表示される URL (SSO ログアウト・ページなど) を指定します。

- ▶ **dashboard_xml_filename** : Dashboard レポート設定ファイルの名前。
- ▶ **hasProxySensor** : Proxy Sensor を使用する場合に、動的な [コンポーネントトポロジアナリシス] ビューを有効にしてプロキシ関連のリンクを表示するには、このプロパティを True に設定します。
- ▶ **bac_lwssso_enabled** : Business Availability Center で LW-SSO が無効になっている場合は、このプロパティを False に設定します。これにより、LW-SSO なしで TransactionVision にアクセスできるようになります。

LW-SSO の詳細については、『**HP Business Availability Center Hardening Guide**』（英語版）PDF を参照してください。

付録 C

データベースの移行

本章の内容

- ▶ 時間と領域の要件 (379 ページ)
- ▶ 使用されていない統合カラムの無効化 (380 ページ)
- ▶ カスタマイズされたデータベース・スキーマの移行 (380 ページ)
- ▶ データベースの移行 - 技術的な詳細 (380 ページ)
- ▶ 非統合環境での TransactionVision の最適化 (382 ページ)

時間と領域の要件

TransactionVision バージョン 7.50 では、データベース・スキーマに大きな変更が加えられました。この変更は、移行プロセスに必要な時間とリソースに大きな影響を与えます。7.50 以前のプロジェクトのテーブル・データのほとんどは、移行時にコピーする必要があります。移行プロセスに実装されたフェイルセーフ機能に起因して、データベース内で最大の既存プロジェクトが占める領域と同じサイズの空き領域がこのコピーのためにさらに必要になります。たとえば、最大の TransactionVision プロジェクトが 2 GB の領域を使用している場合、移行が正常に完了するためには、追加の 2 GB が必要です。また、データベース内のデータの量によっては、移行プロセスが完了するまで非常に長い時間がかかる場合があります。

使用されていない統合カラムの無効化

新しいデータベース・スキーマには、TransactionVision と Business Availability Center スイート内のほかの製品との統合に使用される新しいテーブル・カラムがいくつか含まれています。パフォーマンスが非常に重要な環境で TransactionVision を使用していて、統合を使用するつもりがない場合は、移行プロセス中にこれらの新しいカラムが作成されないようにすることができます。次のコマンドライン・オプションを使って、MigrateDB スクリプトを実行してください（付録 A 「ユーティリティ・リファレンス」を参照）。

```
MigrateDB.sh|.bat -disableMigration,
```

カスタマイズされたデータベース・スキーマの移行

以前の TransactionVision システムでカスタム XDM 定義ファイルを追加していた場合、それらのファイルは新しい設定の中に保持されます。ただし、カスタム・イベント・テーブルについては `proginst_id` キー・フィールドを `INTEGER` から `BIGINT` に変更する必要があるため、カスタム・トランザクション・テーブルについては `business_trans_id` キー・フィールドを `INTEGER` から `BIGINT` に変更する必要があります。標準 XDM ファイルでカラムを追加または変更していた場合、それらの変更は新しいシステムに引き継がれません。セットアップ時に古い設定ファイルの保存を選択し、インストール終了後に必要な更新を行ってください。

データベースの移行 - 技術的な詳細

以下のリストには、TransactionVision バージョン 5.00 以来データベース・スキーマに加えられた変更がすべて含まれています。

TVISION システム・スキーマ

- ▶ テーブル `SCHEMA_VERSION` : カラム `schema` の名前が `schema_name` に変更されました。
- ▶ テーブル `ID_TABLE` : カラム `key` の名前が `key_name` に変更されました。
- ▶ テーブル `SCHEMA` の名前が `SCHEMA_TABLE` に変更されました。
- ▶ テーブル `CLASSIFICATION` が追加されました。

- ▶ テーブル CLASSIFICATION_REL が追加されました。
- ▶ テーブル TXN_CLASS_ATTRIBUTE が追加されました。
- ▶ テーブル TXN_CLASS_ATTR_VALUE が追加されました。
- ▶ テーブル PROPERTIES が追加されました。

プロジェクト・スキーマ

- ▶ テーブル ID_TABLE : カラム key の名前が key_name に変更されました。
- ▶ テーブル SCRATCH : カラム key の名前が key_name に変更されました。
- ▶ PARTIAL_EVENT, EVENT, EVENT_OVERFLOW, USER_DATA, USER_DATA_OVERFLOW, RELATION_LOOKUP の各テーブルで、カラム proginst_id のタイプが INTEGER から BIGINT に変更されました。
- ▶ \$TVISION_HOME/config/xdm にある xdm ファイルを通じて documentType="/Event" と定義されたすべてのテーブルで、カラム proginst_id のタイプが INTEGER から BIGINT に変更されました。
- ▶ テーブル LOCAL_TRANSACTION で、カラム local_trans_id および business_trans_id のタイプが INTEGER から BIGINT に変更されました。
- ▶ テーブル TRACKING_OVERFLOW で、カラム local_trans_id のタイプが INTEGER から BIGINT に変更されました。
- ▶ テーブル BUSINESS_TRANSACTION で、カラム business_trans_id のタイプが INTEGER から BIGINT に変更されました。
- ▶ テーブル EVENT_RELATION で、カラム proginst_id および proginst_id2 のタイプが INTEGER から BIGINT に変更されました。
- ▶ テーブル SYS_MDL_OBJECT で、カラム object_id のタイプが INTEGER から BIGINT に変更されました。
- ▶ テーブル SYS_MDL_OBJECT_RELATION で、カラム object_id および object_id2 のタイプが INTEGER から BIGINT に変更されました。
- ▶ テーブル LOCAL_TRANSACTION : カラム key の名前が key_name に変更されました。
- ▶ テーブル BUSINESS_TRANSACTION : カラム sequential_id の名前が update_id に変更され、カラム・タイプが INTEGER から BIGINT に変更されました。
- ▶ テーブル EVENT_LOOKUP : カラム sequential_id の名前が seq_id に変更され、カラム・タイプが INTEGER から BIGINT に変更されました。

- ▶ テーブル TRANSACTION_STATS : カラム begin の名前が begin_time に変更され、end の名前が end_time に変更されました。
- ▶ テーブル JMS_LOOKUP : カラム exception_class (INTEGER) が追加されました。
- ▶ テーブル BUSINESS_TRANSACTION : カラム exception (INTEGER) が追加されました。
- ▶ SERVLET_LOOKUP に統合のためのカラムが追加されました。追加されたのは、probe_id, probe_group_id, uri, bpm_profile_id, bpm_location_id, bpm_txn_id, client_ip, session_id, url です (データ・タイプは \$TVISION_HOME/config/xdm/Servlet.xdm で指定されています)。
- ▶ BUSINESS_TRANSACTION に、統合を目的とするカラム trans_id および is_bpievent が追加されました (データ・タイプは \$TVISION_HOME/config/xdm/Transaction.xdm で指定されています)。
- ▶ テーブル TRACKING_OVERFLOW の一意のインデックス (tracking_id, tracking_seq) が一意ではなくなりました。
- ▶ テーブル JDBC_LOOKUP が追加されました。
- ▶ テーブル JDBC_STATS が追加されました。
- ▶ テーブル SYS_MDL_OBJECT_ATTR が追加されました。

非統合環境での TransactionVision の最適化

TransactionVision 7.50 では、Business Availability Center 製品ファミリのほかの製品と統合するために、いくつかの新しいプロジェクト・テーブル・カラムが導入されました。どの統合も使用していない場合は、Analyzer での処理中に、これらのカラムに NULL 値が書き込まれます。使用しているデータベース製品が、たとえ NULL 値でもインデックス・ファイルを更新する場合は、それが Analyzer のパフォーマンスに小さな影響を与える可能性があります。したがって、TransactionVision Analyzer を高パフォーマンス環境で使用していて、ほかの製品と統合する予定がない場合は、以下の手順で統合カラムを無効にすることをお勧めします。

- ▶ \$TVISION_HOME/bin から、コマンドライン・スイッチ **-integrations** を使って **TVisionSetupInfo.bat|sh** を実行します。

- ▶ 次の質問に対して **n** と入力します。

Integration Settings

Retrieving current integration settings...

Notice:

If you disable integrations, projects created afterwards may not work properly as the database columns won't exist. It's safer to enable them now and just disable them in the UI later.

IntegrationEnableBAC (y/n) [n]:

IntegrationEnableDiag (y/n) [n]:

これにより、XDM ファイル (**Servlet.xdm** および **Transaction.xdm**) 内のすべての統合カラムが無効になります。新規のプロジェクト・スキーマの場合は、これらのカラムなしでテーブルが作成されるようになります。既存のプロジェクト・スキーマの場合、これらのカラムはテーブル内に残りますが、Analyzer がそれらにアクセスしなくなります。この設定は、対応するコマンドライン・スイッチを使って **TVisionSetupInfo** を再び実行することによって、いつでも変更できます。

付録 D

その他の z/OS 設定

本章の内容

- ▶ RACF 認証 (385 ページ)
- ▶ ファイアウォール設定 (388 ページ)
- ▶ 必要な MIPS (388 ページ)

RACF 認証

RACF 認証は、ユーザの環境に合わせて大幅にカスタマイズされるため、TransactionVision Sensor のための特別な要件はありません。必要な RACF 認証を調べるには、以下の各表に示されたプログラムとトランザクションを利用してください。

TransactionVision CICS Sensor の場合

トランザクション	実行する必要がある	WebSphere MQ にアクセスする必要がある	CICS グローバル・ユーザ出口として実行する必要がある
SLDS	はい	いいえ	いいえ
SLDM	はい	いいえ	いいえ
SLDP	はい	いいえ	いいえ
SLDC	はい	いいえ	いいえ
SLDD	はい	いいえ	いいえ
SLDI	はい	いいえ	いいえ

トランザクション・プログラム	実行する必要がある	WebSphere MQ にアクセスする必要がある	CICS グローバル・ユーザ出口として実行する必要がある
SLDPCSX	はい	いいえ	いいえ
SLDPCMIX	はい	いいえ	いいえ
SLDPCCX	はい	いいえ	いいえ
SLDPDSX	はい	いいえ	いいえ
SLDPUXI	はい	いいえ	いいえ
TVISION	はい	いいえ	いいえ
TVISIONC	はい	はい	いいえ

出口プログラム	実行する必要がある	WebSphere MQ にアクセスする必要がある	CICS グローバル・ユーザ出口として実行する必要がある
SLDPTCX	はい	いいえ	はい
SLDPPSX	はい	いいえ	はい
SLDPICX	はい	いいえ	はい
SLDPTDX	はい	いいえ	はい
SLDPPCX	はい	いいえ	はい
SLDPFCX	はい	いいえ	はい
SLDPTSX	はい	はい	はい

TransactionVision CICS WebSphere MQ (WMQ) Sensor の場合

トランザクション	実行する必要がある	WebSphere MQ にアクセスする必要がある	CICS 交差出口を有効/無効にする必要がある
SLMC	はい	いいえ	はい

トランザクション・プログラム	実行する必要がある	WebSphere MQ にアクセスする必要がある	CICS 交差出口を有効/無効にする必要がある	CICS 交差出口からアクセスする必要がある
CSQCAPX	はい	はい	いいえ	はい
SLMC	はい	はい	はい	いいえ
SLMBCNFG	はい	はい	いいえ	いいえ

TransactionVision IMS WebSphere MQ (WMQ) Sensor の場合

TransactionVision IMS WMQ Sensor は、監視対象の IMS WMQ アプリケーションの中から実行されます。これは、IMS WMQ アプリケーションに対する TransactionVision スタブをリンク編集することによって行われます。したがって、何らかの RACF 認証が実行されます。

以下のライブラリが使用されます。これらのライブラリは、ローカル RACF 認証スキームに基づいて RACF 認証される必要がある場合があります。

ライブラリ	APF 認証される	DFHRPL 内
SSLDLOAD	いいえ	はい
SSLDAUTH	はい	はい

ファイアウォール設定

TransactionVision Sensor は TransactionVision Analyzer と通信するために WebSphere MQ を使用するため、ファイアウォール設定は必要ありません。すべての通信が WebSphere MQ を通じて設定されます。

必要な MIPS

TransactionVision Sensor に対しては、MIPS に関する特別な要件はありません。Sensor は、要求されたときにだけアクティブになるように作られており、アクティブになっているときでさえ、必要な機能を果たすためにほんのわずかなりソースしか使用しません。

付録 E

uCMDB ディスカバリ・エージェント

本章では、IBM z/OS 上で uCMDB ディスカバリ・エージェントをインストール、設定、および操作する方法について説明します。

本章の内容

- ▶ uCMDB ディスカバリ・エージェントのインストールと設定 (390 ページ)
- ▶ uCMDB ディスカバリ・エージェントのコンポーネントと操作 (403 ページ)
- ▶ uCMDB ディスカバリ・エージェントのセキュリティ要件 (405 ページ)
- ▶ uCMDB コマンドの要約 (409 ページ)
- ▶ uCMDB メインフレーム・サービス・エージェントのコンソール・メッセージ (410 ページ)
- ▶ uCMDB z/OS の ディスカバリ・エラー・メッセージ (418 ページ)

重要： エージェントをインストールおよび設定する前に、403 ページ「uCMDB ディスカバリ・エージェントのコンポーネントと操作」と 405 ページ「uCMDB ディスカバリ・エージェントのセキュリティ要件」を両方とも読んでください。それによって、有用な背景情報が得られます。

uCMDB ディスカバリ • エージェントのインストールと設定

uCMDB ディスカバリ • エージェントとそれに必要な製品をインストールして設定する手順は次のとおりです。

作業 1 : システム要件を確認する

作業 2 : uCMDB ディスカバリ • エージェントをインストールする

作業 3 : IBM HTTP サーバをインストールする

作業 4 : uCMDB ディスカバリ用に IBM HTTP サーバを有効にする

作業 5 : uCMDB ディスカバリ • サービス • エージェントをカスタマイズする

作業 1 : システム要件を確認する

インストール先のホストが以下のシステム要件を満たしていることを確認します。

- ▶ オペレーティング • システム IBM z/OS がバージョン 1.7 かそれ以上。
- ▶ IBM HTTP Server for z/OS のバージョン 5.3 (5694-A01) が、作業 3 と作業 4 で説明されているようにインストールされて設定されている。
- ▶ IBM SDSF for z/OS (システム表示および検索機能)。この製品は、IMS サブシステムの特性を検出するために必要です。

以降、IBM HTTP Server for z/OS バージョン 5.3 を IBM HTTP サーバと呼びます。

作業 2 : uCMDB ディスカバリ・エージェントをインストールする

uCMDB ディスカバリ・エージェントをインストールするための一般的な手順は、次のとおりです。

1. インストール・メディアから zSeries プラットフォームにファイルを転送します。次の 2 つの方法があります。
 - ▶ インストール・メディアに含まれている転送支援用の **tvinstall_800_zos_zseries.bat** スクリプトを使用する (推奨)。
 - ▶ インストール・ファイルを手動で FTP する。
2. TSO RECEIVE コマンドを使って、転送されたファイルを受信して製品データ・セットを作成します。次の 2 つの方法があります。
 - ▶ 上の手順 1 で **tvinstall_800_zos_zseries.bat** によって作成された (TSORECV) Rexx スクリプトを使用して、TSO RECEIVE コマンドを実行する (推奨)。
 - ▶ 必要な TSO RECEIVE コマンドを手動で実行する。
3. 環境にとって最良のインストール方法を決定します。次の 2 つの方法があります。
 - ▶ SMP/E (推奨)。新しい SMP/E グローバル・ゾーンを作成することも (推奨)、既存の SMP/E グローバル・ゾーンを使用することもできます。
 - ▶ 非 SMP/E
4. インストール・ジョブをカスタマイズして実行します。必要なカスタマイズの程度は、インストール方法によって異なります。次の 2 つの方法があります。
 - ▶ SMP/E インストール・ジョブ
 - ▶ 非 SMP/E インストール・ジョブ
5. インストール・ライブラリを確認します。

ファイル名の表記規則

インストール・メディア上のファイル名は、以下のパターンに従っています。

dsslr <ファイル修飾子> _<バージョン> _zos_zseries.xmit

FTP 出力データ・セット・ファイルの名前は、以下のパターンに従っていません。

< &custhlq > . <製品コード> | <バージョン> . <ファイル修飾子>

説明：

<ファイル修飾子> = f1 | f2 | f3 | f4 | mcs

<製品コード> = SLR

<バージョン> = 800

< &custhlq > = ターゲット z/OS システム上の FTP PUT サブコマンドの出力に対する、カスタマが指定する高レベルの修飾子。

たとえば、インストール・メディア上のファイル名は次のようになります。

dsslrf1_800_zos_zseries.xmit

FTP 出力データ・セットのファイル名は次のようになります。

DSCVRY.SLR800.F1

手順 1 : インストール・メディアから zSeries ホストにファイルを転送する

次の 2 つの方法があります。

- ▶ **tvinstall_800_zos_zseries.bat** スクリプトを実行する。
- ▶ インストール・ファイルを手動で FTP する。

オプション 1 : tvinstall_800_zos_zseries.bat スクリプトを実行する。

- 1 Microsoft Windows が稼働していて TCP/IP FTP を使ってターゲット z/OS システムに接続できるワークステーションに、ログオンします。
- 2 MS-Windows のコマンド・プロンプト・セッションを開始します ([スタート] > [すべてのプログラム] > [アクセサリ] > [コマンド プロンプト])。必要に応じてウィンドウのプロパティを調整し、そのサイズとバッファを増やします。
- 3 ディレクトリ変更 (cd) コマンドを使って、インストール・メディアがあるフォルダへ移動します。
- 4 次のように入力して、スクリプトが収集する情報を確認します。

```
tvinstall_800_zos_zseries.bat /h | more
```

- 5 パラメータなしで **tvinstall_800_zos_zseries.bat** を起動し、プロンプトに対して情報を入力します。

注 : 入力した値に満足できなかった場合は、ファイル転送を開始せずに **tvinstall_800_zos_zseries.bat** を終了することができます。また、CTRL+C キーを繰り返し押せば、いつでもこのスクリプトを終了できます。

6 **tvinstall_800_zos_zseries.bat** によって発行された FTP コマンド出力の結果を注意深く確認します。

- ▶ エラーがあってはなりません。
- ▶ ターゲット z/OS システムで指定された高レベル修飾子を持つインストール用 RELFILE が存在することを確認します。
- ▶ 指定した PDS ライブラリの中に Rexx スクリプト (TSORECV) が存在することを確認します。
- ▶ データ・セットが存在しない場合や、FTP エラーが検出された場合は、それらを調査して修正し、tvinstall_800_zos_zseries.bat プロセスを繰り返す必要があります。

オプション 2: インストール・ファイルを手動で FTP する。

tvinstall_800_zos_zseries.bat スクリプトを使ってファイルをインストールした場合は、この手順を実行しないでください。

- 1 Microsoft Windows が稼働していて TCP/IP FTP を使ってターゲット z/OS システムに接続できるワークステーションに、ログオンします。
- 2 MS-Windows のコマンド・プロンプト・セッションを開始します ([**スタート**] > [**すべてのプログラム**] > [**アクセサリ**] > [**コマンド プロンプト**])。必要に応じてウィンドウのプロパティを調整し、そのサイズとバッファを増やします。
- 3 ディレクトリ変更 (cd) コマンドを使って、インストール・メディアがあるフォルダへ移動します。
- 4 希望する高レベル修飾子を付けてデータ・セットを作成するのに十分な権限を持つ z/OS アカウントを使って、ターゲット z/OS システムとの FTP セッションを開始します。次に例を示します。

```
> ftp hostname
```

- 5 次の FTP サブコマンドを発行します。

```
a ftp> quote site fixrecfm 80 lrecl=80 recfm=fb blksize=3120 vol=&custvol  
u=&custunit pri=30 sec=5 tr
```

&custvol は有効なディスク volser で、**&custunit** は有効なディスク・デバイス・タイプまたはエソテリックです。

```
b ftp> bin
```

```
c ftp> put dsslr <ファイル修飾子> _<バージョン> _zos_zseries.xmit
'< &hlq > .slr < |バージョン> .<ファイル修飾子>'
```

説明：

< &hlq > = カスタマが選択した高レベル修飾子

<ファイル修飾子> = f1 | f2 | f3 | f4 | mcs

<バージョン> = 800

次に例を示します。

```
PUT dscvry_agent_slr_800_zos_zseries.xmit '&hlq.slr|ver.filequal'
```

d ファイル名サフィックスが **fn** または **mcs** の、残りの製品インストール・ファイルについて、FTP **put** コマンドを発行します。

e FTP コマンドの出力を注意深くチェックして、エラーがないか確認することが重要です。エラーを検出した場合は、z/OS システム上のターゲット・データ・セットを削除して ftp put コマンドを再実行しなければならないことがあります。

```
f ftp> quit
```

手順 2 : TSO RECEIVE コマンドを使って、転送されたファイルを受信して製品データ・セットを作成する

次の 2 つの方法があります。

- ▶ 392 ページ「手順 1 : インストール・メディアから zSeries ホストにファイルを転送する」で tvinstall_800_zos_zseries.bat によって作成された (TSORECV) REXX スクリプトを実行する。
- ▶ 必要な TSO RECEIVE コマンドを手動で実行する

オプション 1 : (TSORECV) REXX スクリプトを実行する

TSO コマンド行または TSO/ISPF コマンド行から、上の手順 1 で特定した PDS の中にある (TSORECV) REXX スクリプトを、必要に応じて前に TSO を付けて実行します。

```
exec 'PDS.identified.instep1(TSORECV)'
```

これにより、上の手順 1、6 で指定されたすべての z/OS ターゲット・データ・セットについて TSO RECEIVE コマンドが実行されます。受信されたデータ・セットは、同じ高レベル修飾子を使って割り当てられますが、製品コードにプレフィックスとして文字 A が付け加えられます。

たとえば、入力パターンが DSCVRY.SLR800.filequal の場合、出力パターンは DSCVRY.ASLR800.filequal のようになります。

オプション 2 : 必要な TSO RECEIVE コマンドを手動で実行する

- 1 TSO コマンド行または TSO/ISPF コマンド行から、必要に応じて前に TSO を付けて、次の表に示した RECEIVE コマンドを実行します。プロンプト「INMR906A Enter restore parameters or 'DELETE' or 'END」に対しては、DSN(&custhlq.ASLR800.filequal) と入力します。

コマンド	ファイル名
RECEIVE INDSNAME('&hlq.SLR8000.F1')	DSN('&hlq.ASLR800.F1')
RECEIVE INDSNAME('&hlq.SLR8000.F2')	DSN('&hlq.ASLR800.F2')
RECEIVE INDSNAME('&hlq.SLR800.F3')	DSN('&hlq.ASLR800.F3')
RECEIVE INDSNAME('&hlq.SLR800.F4')	DSN('&hlq.ASLR800.F4')
RECEIVE INDSNAME('&hlq.SLR800.MCS')	DSN('&hlq.ASLR800.SMPMCS')

手順 3 : 最良のインストール方法を決定する

次の 2 つの方法があります。

- ▶ SMP/E (推奨)。新しい SMP/E グローバル・ゾーンを作成することも (推奨)、既存の SMP/E グローバル・ゾーンを使用することもできます。
- ▶ 非 SMP/E

手順 4 : インストール・ジョブをカスタマイズして実行する

次の 2 つの方法があります。SMP/E または非 SMP/E

オプション 1 : SMP/E インストール・ジョブ

表に示された順序でジョブをカスタマイズして実行します。ジョブは、データ・セット **&custhlq.ASLR800.F3** の中にあります。各ジョブの先頭にあるコメントを読み、それに従ってカスタマイズしてください。

実行が完了したら、次のジョブに移る前にすべての出力を注意深くチェックして、すべての手順が正常に完了したことを確認します。既存の SMP/E グローバル・ゾーンにインストールする場合は、SMP/E 手順、パラメータ、および入力を注意深くチェックしてください。

SMP/E インストール • ジョブ

順序	ジョブ	説明
1.	SLRALLOC	ターゲット・ライブラリと配布ライブラリを割り当てます。
2.	SLRGZON	SMP/E グローバル・ゾーンを定義します (既存のグローバル・ゾーンを使用する場合は不要)。
3.	SLRDZON	SMP/E 配布ゾーンを定義します。
4.	SLRTZON	SMP/E ターゲット・ゾーンを定義します。
5.	SLRDDDEF	SMP/E DDDEF を定義します。
6.	SLRRECV	SMP/E Receive
7.	SLRAPPLY	SMP/E Apply
8.	SLRIVP	インストールの検証
9.	SLRACPT	SMP/E Accept 重要 ! SMP/E ACCEPT 機能がいったん実行されてしまうと、ターゲットまたは配布ライブラリから製品を削除できなくなるため、このジョブはインストールが設定されて完全にテストされた後のみ実行してください。

オプション 2 : 非 SMP/E インストール • ジョブ

ジョブは、データ・セット **&custh1q.ASLR800.F3** の中にあります。各ジョブの先頭にあるコメントを読み、それに従ってカスタマイズしてください。実行が完了したら、すべての出力を注意深くチェックして、すべての手順が正常に完了したことを確認します。

非 SMP/E インストール • ジョブ

順序	ジョブ	説明
1.	SLRINSTL	
2.	SLRIVP	インストールの検証

手順 5：インストール・ライブラリを確認する

DS#	データ・セット名	説明
1	DSCVRY.SSLRCARD	制御カード, 設定データ
2	DSCVRY.SSLREXEC	Rexx CGI プログラム・ソース
3	DSCVRY.SSLRINST	インストール JCL
4	DSCVRY.SSLRLOAD	サービス・エージェント Loadlib (APF Auth)
5	DSCVRY.SSLRPROC	サービス・エージェント Proclib
6	DSCVRY.SSLRSAMP	サンプル JCL, コマンドなど
7	DSCVRY.SLR800.F1	Inst RELFILE - JCLIN
8	DSCVRY.SLR800.F2	Inst RELFILE - ロード・モジュール
9	DSCVRY.SLR800.F3	Inst RELFILE - インストール JCL
10	DSCVRY.SLR800.F4	Inst RELFILE - Rexx プログラム
11	DSCVRY.SLR800.SMPMCS	Inst RELFILE - SMP/E MCS ステートメント

- ▶ この表には、インストール用 SMP/E 配布ライブラリ (DLIBS) は含まれていません。これらは、SLRACCPT が実行されるまでは読み込まれません。
- ▶ 「DSCVRY」は標準設定の高レベル修飾子ですが、実際に使用されるものはこれとは異なる場合があります。

作業 3：IBM HTTP サーバをインストールする

IBM HTTP Server for z/OS バージョン 5.3 は、z/OS の基本インストール・コンポーネントであり、IBM z/OS のカスタマには無料で提供されます。必要な製品サービスおよび修正の情報については、本製品の z/OS プログラム・ディレクトリ、PSP のドキュメント、および IBMLink を参照してください。

本製品の最新情報は、次の URL にある IBM Publications Center で提供されています。

<http://www.elink.ibmink.ibm.com/public/applications/publications/cgibin/pbi.cgi>

本付録では、IBM 発行の『**HTTP Server Planning, Installing, and Using**』（英語版）ドキュメント (SC34-4826-09) を参照しています。

本製品をインストールする前に、このドキュメントの Chapter 1 を読んでください。

IBM HTTP サーバをインストールするには、以下の手順を実行します。

- 1 インストールに関連する以下の点に注意してください。
 - ▶ uCMDB ディスカバリ・コンポーネントは、カスタム DLL は使用しません。
 - ▶ 高速応答キャッシュ・アクセラレータは、この時点では使用されません。
 - ▶ 同時ユーザ接続がある場合、uCMDB ディスカバリのアクティビティは、パフォーマンスとチューニングにほとんど関与しません。したがって、uCMDB ディスカバリのアクティビティがプライマリ IBM HTTP サーバの主要なアプリケーション負荷であると想定し、MaxActive スレッド設定、MAXSOCKETS、MAXFILEPRO、および関連パフォーマンス設定を比較的低く設定することができます。
 - ▶ 本書に記載されている外部システム・マネージャ（ESM：External Security Manager）のカスタマイズ、コマンド、および使用例は、IBM のセキュリティ・サーバ（RACF）に関するものです。uCMDB ディスカバリの実装には RACF は必要ありませんが、ACF2 や TopSecret などのほかの ESM 製品を使用する場合は、対応するコマンドと関数を特定して、カスタマの非 RACF ESM 環境で実行する必要があります。
 - ▶ IBM HTTP Server の実行に使用する ESM アカウントは、次の能力を持っていない限りなりません。
 - ▶ ディスカバリ・サービス・エージェント loadlib (DS #4) を読み取るのに十分な権限。
 - ▶ IBM HTTPS サーバの `./cgi-gin` ディレクトリにある REXX スクリプトを実行するのに十分な権限。
- 2 『HTTP Server Planning, Installing, and Using』（英語版）ドキュメントの Chapter 3 に記載されている手順に従って、IBM HTTP サーバをインストールします。

基本インストールの後、Chapter 3 に記載されているカスタマイズ手順の手順 1 ~ 10 を慎重に確認して実行します。いくつかの手順には、セキュリティのカスタマイズが含まれています（SSL、公開鍵リング証明書、ゼロ以外のユーザ ID を使った Web サーバ・アカウントの作成など）。これらのカスタマイズは Web サーバの運用と uCMDB ディスカバリの処理にとって絶対に必要なものではありませんが、uCMDB ディスカバリの処理は複数のプラットフォームにまたがるため、Hewlett-Packard ソフトウェアでは、環境のセキュリティ・プロファイル全体を強化する機能をフルに活用するよう uCMDB ディレクトリのカスタマに強く推奨しています。

サーバのセキュリティを設定するときに役立つ以下のサンプル・ジョブが提供されています。一部のサンプルは、あらゆる環境に適合するわけではありません。これらのサンプルをカスタマイズし、『**HTTP Server Planning, Installing, and Using**』（英語版）ドキュメントの Chapter 3 に記載されている手順と組み合わせて実行する必要があります。

対応する手順は、Chapter 3 の手順 1 ～ 5 にあります。これらをここで紹介するのは、読者の便宜のためです。

- ▶ **SLRSEC01.** IMWEB グループおよび WEBADM ユーザ定義
- ▶ **SLRSEC02.** WEBSRV ユーザ定義
- ▶ **SLRSEC03.** ゼロ以外の USS ユーザ ID でユーザを定義する
- ▶ **SLRSEC04.** サロゲート・クラス・プロファイル定義
- ▶ **SLRSEC05.** MVS データ・セットに対してプログラム制御をオンにする
- ▶ **SLRSEC06.** z/OS SSL についてプログラム制御をオンにする
- ▶ **SLRSEC07.** 鍵リングへの WEBSRV アクセスを許可する
- ▶ **SLRSEC08.** 暗号化ハードウェア・サービスへの WEBSRV アクセスを許可する
- ▶ **SLRSEC09.** 暗号化サービス内のプロファイルへの WEBSRV アクセスを提供する
- ▶ **SLRSEC10.** z/OS 作業負荷管理への WEBSRV アクセスを許可する
- ▶ **SLRSEC11.** SMF 機能への WEBSRV アクセスを許可する
- ▶ **SLRSEC12.** 開始済みタスク・テーブルに IBM HTTP Server PROC を追加する

3 『HTTP Server Planning, Installing, and Using』（英語版）ドキュメントの Chapter 4, Chapter 5, および Chapter 6 に記載されている手順に従って、IBM HTTP サーバの基本機能をテストします。

IBM HTTP サーバ自体のための IVP はないように見えますが、これらの各章では、Web サーバの標準の第 1 ページの起動、シャットダウン、およびアクセスの手順を説明しています。このページは、設定用の代替インタフェースを提供します。追加のカスタマイズを行う前に、基本的なサーバ機能をチェックすることが重要です。

作業 4 : uCMDB ディスカバリ用に IBM HTTP サーバを有効にする

IBM HTTP サーバの基本機能をチェックした後、uCMDB ディスカバリ・コンポーネントの操作に固有のカスタマイズを実行します。

- 1 IBM HTTP サーバの SSL 通信と基本的な認証機能を有効にすることを強くお勧めします。『**HTTP Server Planing, Installing, and Using**』（英語版）ドキュメントの Chapter 8 を参照してください。

DS#6（サンプル JCL）に属する次のメンバは、基本的な SSL 通信を有効にするために必要な RACF コマンドと関連タスクの例として参考になります。

SLRSSLEX SSL Setup - Sample Steps

このサンプルは、上記のドキュメントの Chapter 8 の説明に従って使用してください。

- 2 IBM HTTP サーバに対応する開始済みタスク JCL 中の STEPLIB 連結を更新します。IBM HTTP サーバに対応する開始済みタスクの標準設定のプロシージャ名は、IMWEBSRV です。2 つのライブラリを STEPLIB 連結に追加する必要があります。
 - a DS #4（サービス・エージェント・ロード・ライブラリ）は、連結の最後に追加しなければなりません。
 - b DB2 がインストールされている場合は、DB2 SDSNLOAD ライブラリを追加してから DB2 RUNLIB.LOAD ライブラリも追加します。
- 3 IBM HTTP サーバの当該インスタンスに関連付けられたプライマリ設定ファイル（通常は **httpd.conf**）を更新し、Rexx CGI スクリプト用の Exec マッピング・ルールを含めます。ファイルの標準設定の場所は次のとおりです。

`/etc/httpd.conf`

ただし、この場所は変更されている可能性があります。IBM HTTP サーバの出力メッセージ内で次の文字列を見つけてプライマリ設定ファイルの場所と名前を確認してから、そのファイルに対して編集セッションを開始してください。

"Using configuration file"

CGI プログラムおよびスクリプトの実行に関する以下の Exec マッピング・ルールを追加します。必要に応じて、指定されたパス内にターゲット・ディレクトリを作成してください。

Exec	/uCMDB/*.cgi	/etc/HP/Discovery/cgi-bin/*.cgi
Exec	/uCMDB/*.sh	/etc/HP/Discovery/cgi-bin/*.sh

注 : CGI プログラムおよびスクリプトの実際の場所を変更できます。セキュリティ属性もともに変更すれば、それによって問題が起きることはありません。

- 4 『HTTP Server Planning, Installing, and Using』(英語版) ドキュメントの Chapter 3 の手順 1 で特定したユーザ ID は、上記のディレクトリにある Rexx CGI スクリプトの読み取りと実行ができる権限を持っていないければなりません。必要に応じてセキュリティ属性を調整してください。

作業 5 : uCMDB ディスカバリ • サービス • エージェントをカスタマイズする

IBM HTTP Web Server for z/OS の基本機能をチェックした後、uCMDB ディスカバリの操作に固有のカスタマイズを実行する必要があります。

- 1 適切な置き換えを行ってから、TSO/ISPF コマンド・シェル (オプション 6) から次のコマンドを実行します。これにより、400 ページの手順 3 で特定された USS ファイルシステム (../dscvry/cgi-bin) に、Rexx CGI スクリプトがコピーされます。

```
OPUTX 'ソース PDS' 'ターゲット USS ファイルシステム' TEXT  
CONVERT(YES)
```

次に例を示します。

```
OPUTX 'DSCVRY.SSLREXEC' '/etc/HP/Discovery/cgi-bin' TEXT  
CONVERT(YES)
```

- 2 uCMDB ディスカバリ • サービス • エージェントは、検出要求に伴うユーザ ID とパスワードに対して、SAF 基本認証を実行します。基本認証が正常に完了するためには、このユーザ ID とパスワードが ESM に知られている必要があります。
- 3 DS#1 (制御カードと設定データ) にある PDS メンバ SLRSRVIN をカスタマイズします。
- 4 必要に応じて、ACL (Authorized Command List) ファイル、または上記の作業 5 の手順 3 で言及したメンバを、確認してカスタマイズします。標準設定の ACL メンバである SLRACL は、DS#1 (制御カードと設定データ) の中にあります。
- 5 APF は、uCMDB ディスカバリ • ライブラリ &hlq.SSLRLOAD (DS#4) を認証します。詳細については、『MVS Initialization and Tuning Reference (SA22-7592)』(英語版) を参照してください。

- 6 各メンバの先頭にある指示に従って以下の JCL 開始済みタスクをカスタマイズした後、それらを当該サイトの適切な JCL PROC ライブラリにコピーします。
これらの開始済みタスク・プロシージャのサンプル JCL は、DS#5（サービス・エージェント Proclib）の中にあります。

注：以下のプロシージャ名がサイトの標準またはプリファレンスと矛盾する場合は、名前を変更できます。ただし、SLRSTSREX の場合は、その変更を SLRSERV 設定ファイルにも反映させる必要があります。

- ▶ SLRSERV
- ▶ SLRSTSREX

- 7 SLRSERV と SLRSTSREX は、MVS 開始済みタスクとして実行されます。これらのタスクに関連付けられる標準設定のユーザ ID は、以下の能力を持っている必要があります。
 - ▶ USS ファイルシステム /tmp/Discovery の中にファイルを作成する
 - ▶ STEPLIB DD ステートメントの中で特定されたすべてのライブラリに対する読み取り権限
 - ▶ SYSEXEC DD ステートメントによって特定されたライブラリに対する読み取り権限
 - ▶ 設定ファイルと ACL（Authorized Command List）に対する読み取り権限
 - ▶ MVS MCSE コンソールを作成および使用する権限
 - ▶ ACL（Authorized Command List）によって特定されたすべてのコマンドを実行する権限
- 8 全体的なアーキテクチャ、コンポーネント、および基本的な操作をより深く理解するには、403 ページ「uCMDB ディスカバリ・エージェントのコンポーネントと操作」を参照してください。
- 9 製品を使ってみた後、391 ページ「作業 2：uCMDB ディスカバリ・エージェントをインストールする」の手順 3 に戻って、**SLRACPT** (SMP/E Accept) ジョブを実行してください。

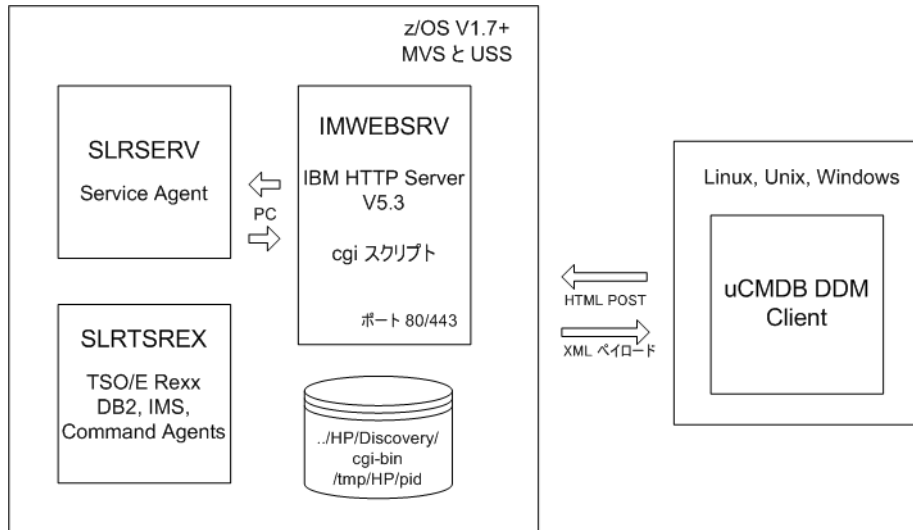
uCMDB ディスカバリ・エージェントのコンポーネントと操作

ここでは、uCMDB ディスカバリ・エージェントのアーキテクチャの簡単な概要と、z/OS ベースのコンポーネントの起動と停止の手順について説明します。

次の表は、uCMDB ディスカバリ・エージェントの主なコンポーネントを示しています。

プロシージャ	説明
IMWEBSRV	IBM HTTP Server V5.3 for z/OS（標準設定のプロシージャ名が使用されます）
SLRSERV	MVS コンソール・サービス・エージェント（開始済みタスク）
SLRTSREX	TSO/E Rexx コマンド・プロセッサ（開始済みタスク）
DDM クライアント	HP uCMDB DDM クライアント（Linux, UNIX, または Windows プラットフォーム上で動作するリモート Discovery and Dependency Mapping クライアント）

uCMDB のアーキテクチャは次のとおりです。



SLRSERV が正しく動作するためには、IBM HTTP サーバが起動されて実行中ではなければなりません。『**HTTP Server Planning, Installing, and Using**』（英語版）ドキュメントの Chapter 4 と Chapter 6 を参照してください。

SLRSERV の起動

SLRSERV を起動するには、次のように MVS コンソールの「起動」コマンドを使用します。

```
S SLRSERV
```

注：SLRSERV プロシージャは、MVS オペレーティング・システムに対して内部「起動」コマンドを発行することによって、SLRTSREX を自動的に起動します。SLRTSREX を手動で起動しないでください。

いったん起動された SLRSERV は、連携する SLRTSREX タスクとともに、次の IPL まで動作し続けます。

SLRSERV の停止

必要であれば、次のように MVS の「停止」コマンドを使って SLRSERV を終了することができます。

```
P SLRSERV
```

注：SLRSERV が終了すると、SLRTSREX も自動的に終了します。SLRTSREX を手動で停止しないでください。

ACL の更新

ACL（Authorized Command List）を含む動的設定データは、EXEC ステートメントの PARM= パラメータを通じて SLRSERV に渡されます。最新の変更を取得するために ACL を再読み込みするよう SLRSERV に指示するには、MVS の modify REFRESH コマンドを使用します。次に例を示します。

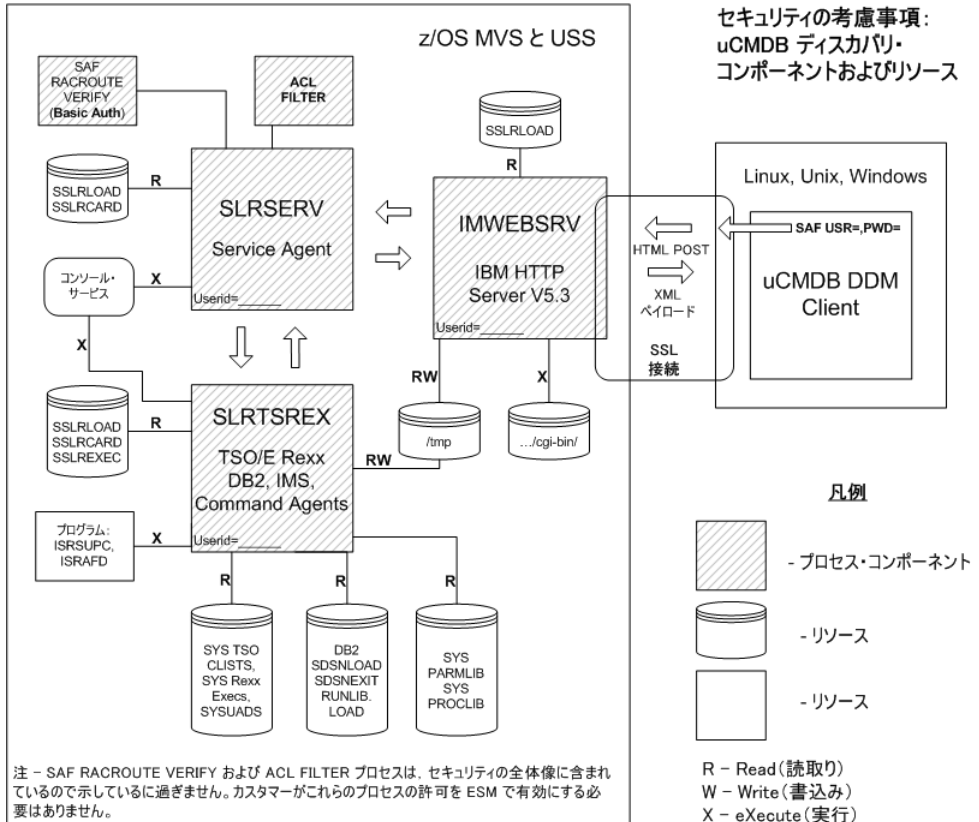
```
F SLRSERV,REFRESH
```

uCMDB ディスカバリ・エージェントのセキュリティ要件

以下のリストは、uCMDB ディスカバリ・エージェントのコンポーネントに権限を付与して認証するために必要な ESM 固有のコマンド文字列を構築するために、z/OS ESM (External Security Manager) とともに使用してください。

リスト内の項目はすべて、本書に記載されているさまざまなインストールおよび設定の説明の中に見られます。簡単に参照できるように、それらをここにまとめました。このリストは、ディスカバリ・エージェントのコンポーネント別になっています。各ディスカバリ・エージェント・コンポーネントに関連付けられるユーザ ID は、プライマリ、セカンダリ、グループのどれであっても、説明されているタスクを実行するのに十分な権限を付与される必要があります。

次の図は、関係するさまざまなリソースと、それらへのアクセスを必要とするディスカバリ・エージェント・コンポーネントを示しています。



IBM HTTP サーバ

- ▶ REXX マッピング・ルールに関して HTTP サーバの `httpd.conf` ファイルによって特定されたフォルダの中にある `cgi-bin` スクリプトおよびプログラムに対する実行権限。このフォルダの場所はカスタマイズできます。推奨されているパスは `/etc/HP/Discovery/cgi-bin/` です。
- ▶ `/tmp` ファイルシステム内のファイルおよびサブディレクトリへの読み取り権限と書き込み権限。
- ▶ DS#4 - DSCVRY.SSLRLOAD（作業 2 の手順 5 の表を参照）に対する読み取り権限（注：このライブラリは APF 認証済みとして MVS に対して定義する必要もあります）。
- ▶ uCMDB ディスカバリ・サービス・エージェントが特に必要としているわけではありませんが、Hewlett-Packard ソフトウェアでは、カスタマが IBM HTTP サーバ環境のセキュリティを確保するために適切な予防策を講じるよう強くお勧めしています。その予防策には、『**IBM HTTP Server Planning, Installing, and Using**』（英語版）ドキュメント（SC34-4826）で説明されている、SSL および管理のカスタマイズが含まれます。SSL を有効にしたり HTTP サーバのセキュリティを確保するときに役立つ、RACF セキュリティ環境用のサンプル・ジョブストリームが用意されています（SLRSSLEX、DS#6 - DSCVRY.SSLRSAMP の中の SLRSEC01 ~ SLRSEC12、サンプル JCL、コマンドなどを参照してください）。

SLRSERV によって開始されるタスク

- ▶ MCSE コンソール・サービスの作成と使用に関連する MVS 認証済みサービス呼び出し：MCSOPER (ACTIVATE, DEACTIVATE), MGCRCR, MCOPMSG
- ▶ さまざまな MVS コマンド（代表的なのは表示コマンド）、TSO コマンド、およびいくつかの REXX Exec および TSO CLIST。一部のコマンドは、「認識文字」を通じて特定のサブシステムに向けて発行されるため、そのような文字の使用を含めるために特別な権限が必要です。潜在的に発行される可能性があるすべてのコマンドの完全なリストが、本書のコマンド・リファレンスの項に記載されています。コマンドを実行する能力は、後述する ACL（Authorized Command List）を使ってカスタマが制御することができます。
- ▶ DS#1 - DSCVRY.SSLRCARD（作業 2 の手順 5 の表を参照）または対応するカスタマ QSAM ファイルまたは SLRSERV 設定ファイルおよび ACL（Authorized Command List）を含んだ PDS に対する読み取り権限。

- ▶ DS#4 - DSCVRY.SSLRLOAD（作業 2 の手順 5 の表を参照）に対する読み取り権限（注：このライブラリは APF 認証済みとして MVS に対して定義する必要もあります）。
- ▶ カスタマが自身の ESM を使って定義した細かい操作コマンドの権限に加えて、どのコマンド、exec、および clist がディスカバリ・サービス・エージェントによって実行されるかを、ACL（Authorized Command List）を使って制御できます。ACL の場所は SLRSERV 設定ファイルの中で定義され、SLRSERV プロシージャの PARM= ステートメントから読み取られます。ACL が指定されなかった場合、SLRSERV は、DS#1 - DSCVRY.SSLRCARD（制御カード、設定データ）の中にある標準設定の ACL（SLRACL）に内容が似ている内部的に定義された標準設定の同等物を使用します。使用されている ACL ファイルの内容をチェックし、必要に応じて調整してください。ACL は、ESM オペレータ・コマンド・レベルのセキュリティなしで、あるいはそれと連携して、ディスカバリ・サービス・エージェントがどのコマンドまたはコマンド・クラス（ワイルドカードが使用された場合）を実行できるかを制御する機能をカスタマに提供します。

SLRTSREX 開始済みタスク

- ▶ DS#4 - DSCVRY.SSLRLOAD（作業 2 の手順 5 を参照）に対する読み取り権限（注：このライブラリは APF 認証済みとして MVS に対して定義する必要もあります）。
- ▶ DB2 コンポーネントがディスカバリの対象である場合は、すべての DB2 SDSNLOAD, SDSNEXIT, および RUNLIB.LOAD ライブラリに対する読み取り権限。
- ▶ ローカル SYSUADS データ・セットに対する読み取り権限。
- ▶ ローカル・システムの TSO CLIST および REXX Exec ライブラリに対する読み取り権限。
- ▶ DS#2 - DSCVRY.SSLREXEC（作業 2 の手順 5 の表を参照）に対する読み取り権限。
- ▶ DS#2 - DSCVRY.SSLRCARD（作業 2 の手順 5 の表を参照）に対する読み取り権限。
- ▶ /tmp ファイルシステム内のファイルおよびサブディレクトリへの読み取り権限と書き込み権限。
- ▶ システム・レベルのパラメータおよびプロシージャ・ライブラリ（SYS1.PARMLIB, USER.PROCLIB など）に対する読み取り権限。

- ▶ 文字列検索のために IBM プログラム ISRSUPC を実行する権限。
- ▶ IBM の SDSF プログラム ISRAFD を実行する (IMS リソース検出のために) 権限。
- ▶ DB2 プラン DSNREXX および DSNTEP2 の実行権限。
- ▶ MCSE コンソール・サービスの作成と使用に関連する MVS 認証済みサービス呼び出し : MCSOPER (ACTIVATE, DEACTIVATE), MGCRE, MCOPMSG
- ▶ さまざまな MVS コマンド (代表的なのは表示コマンド), TSO コマンド, およびいくつかの Rexx Exec および TSO CLISTS。一部のコマンドは、「認識文字」を通じて特定のサブシステムに向けて発行されるため、そのような文字の使用を含めるために特別な権限が必要です。

その他

- ▶ uCMDB DDM クライアントには、独自のセキュリティ要件があります。
- ▶ すべての uCMDB DDM クライアント要求には、基本的な SAF 認証を実行するために SLRSERV によって開始されるタスクにより使用されるユーザ ID とパスワードが関連付けられています。このユーザ ID は、z/OS ESM に対して定義されなければなりません。

uCMDB コマンドの要約

以下のコマンドは、uCMDB ディスカバリ ・ サービス ・ エージェントによって実行されます。これらのコマンドは、コマンド ・ タイプ内でコマンド ・ サブタイプによって分類されています。サービス ・ エージェントによって認識され、ACL 内で意味を持つコマンド ・ タイプは、MVS、TSO、および SAF です。MVS タイプに分類されているコマンドの一部は、実際には DB2、CICS、および MQ Series コマンドです。将来のリリースでは、コマンド ・ タイプについてもっと詳しい説明が含められる予定です。

MVS

D ASM
 D M=CPU
 D OPDATA
 D NET,MAJNODES
 D PROD,REGISTERED
 D PROD,STATE
 D R
 D SSI
 D SYMBOLS
 D TCPIP,,NETSTAT,CONN
 D TCPIP,,NETSTAT,DEV
 D TCPIP,,NETSTAT,HOME
 D TCPIP,,NETSTAT,ROUTE
 D VIRTSTOR,HVSHARE
 D XCF,GRP
 D XCF,GRP, グループ名 ,ALL

DISPLAY CHANNELMQ Series
 DISPLAY CHINITMQ Series
 DISPLAY SYSTEMMQ Series

CEMT I TASCICS

DISPLAY DATABASE DB2
 DISPLAY GROUP DB2
 DISPLAY LOCATION DB2

TSO

TSO ALTLIB DISPLAY
TSO LISTA SYSNAMES
TSO LISTALC SYSNAMES
TSO NETSTAT
TSO STATUS
TSO TIME

SLRRDFLERexx exec (ファイルを読み取る)
SLRREXSCRexx exec (動的 SQL 実行)
SLRSRPDSRexx exec (PDS メンバを検索する)
SLRDTEP2Rexx exec (動的 SQL 実行のために DSNTEP2 を呼び出す)
SLRGETSS
SLRIMSDSRexx exec (IMS DISPLAY ACTIVE コマンドを実行するために
使用される)
SLREXPRTSO CLIST (システム・ソフトウェア・インベントリ・レポート
を生成する)

SAF

SAF USR=,PWD= 内部的な SAF 基本認証要求

uCMDB メインフレーム・サービス・エージェントのコンソール・メッセージ

オペレータ・コンソールには、以下のメッセージが表示されます。これらのメッセージも、さまざまなほかのメッセージとともに、MSGOUT データ・セットに書き込まれます。

SLR-0001-E: Not APF authorized

uCMDB サービス・エージェントが APF 認証されていません。このメッセージは、uCMDB サービス・エージェントのロード・ライブラリが APF 認証されていないか、SLRAGENT ロード・モジュールの認証コードがゼロであるか、実行 JCL の STEPLIB に 1 つ以上の無許可ライブラリが含まれているか、無許可のソフトウェアによって SLRAGENT が呼び出されたことを意味している場合があります。

SLR-0002-I: TRANSWAP completed successfully

このメッセージは、uCMDB サービス・エージェントのアドレス空間が自分自身を正常にスワップ不可能にしたことを意味します。SLRAGENT が制御を受け取ったときに uCMDB サービス・エージェントのアドレス空間がすでにスワップ不可能になっていた場合には、このメッセージは表示されません。

SLR-0003-I: uCMDB services agent (SLRvvvvv) main task starting

このメッセージは、uCMDB サービス・エージェントが起動および初期化シーケンスを開始したときに表示されます。カスタマは、自動化のためにこのメッセージを利用できます。このメッセージには、最大 5 文字までのバージョン/リリース・コードが含まれます (vvvvv で表されます)。

SLR-0011-E: Operating system is too old, no longer supported

ホスト・オペレーティング・システムが、uCMDB サービス・エージェントが必要とする機能をすべては提供していません。

SLR-0012-E: Error in authorized command list

設定ファイル内の CMDLIST= 句が、ACL (Authorized Command List) の名前を指定しています。しかし、指定された ACL ファイルにアクセスできないか、そのファイルに 1 つ以上のエラーが含まれています。詳細については、MSGOUT を参照してください。

SLR-0013-E: Error in configuration file

実行 JCL 内の EXEC ステートメントに空でない PARM= 句が含まれており、その句は設定ファイルの名前を指定しています。しかし、指定された設定ファイルにアクセスできないか、そのファイルに 1 つ以上のエラーが含まれていません。詳細については、MSGOUT を参照してください。

SLR-0014-E: Invalid PARM= clause in execution JCL

実行 JCL 内の EXEC ステートメントに空でない PARM= 句が含まれています。PARM= 句で指定されたテキストに、何かのエラーが含まれています (たとえば、PARM テキストの長さが 100 バイトを超えているなど)。

SLR-0015-E: ALESERV rc=X'hhhh'

ALESERV EXTRACTH マクロ・ステートメントが、ゼロ以外のリターン・コードを返しました。このメッセージには、ALESERV マクロが返したリターン・コードが含まれています。ALESERV のリターン・コードについては、『**MVS Programming: Authorized Assembler Services Reference, Volume 1 (ALESERV-DYNALLOC)**』（英語版）で説明しています。

SLR-0016-E: Services agent already active

現在の z/OS イメージ内で uCMDB サービス・エージェントの別のインスタンスがすでにアクティブになっています。

SLR-0017-E: Unable to load xxxxxxxx, retc=X'hhhhhhhh', reas=X'hhhh'

指定されたモジュール（xxxxxxx で表されます）をロードできませんでした。このメッセージは、メモリの不足や、指定されたモジュールが STEPLIB 内にないことを意味する場合があります。

SLR-0025-I: Attach of subtask xxxxxxxx failed, rc=X'hhhh'

ATTACH マクロ・ステートメントが、ゼロ以外のリターン・コードを返しました。これは、通常はメモリの不足を意味しています。

SLR-0026-I: Subtask xxxxxxxx terminated, ecb=X'40ssuuu'

指定されたサブタスクが終了しました。このメッセージは、終了したサブタスクのメイン・プログラム名と、サブタスクの終了 ECB の内容（32 ビット）を示します。一般に、sss はシステム異常終了コードを示し、uuu はユーザ異常終了コードを示します。

SLR-0028-I: uCMDB Services Agent main task is now active

このメッセージは、uCMDB サービス・エージェントが起動および初期化シーケンスを完了したときに表示されます。カスタマは、自動化のためにこのメッセージを利用できます。

SLR-0029-I: RQE successfully reclaimed

このメッセージは、クライアント・プログラムが要求を発行したけれど、その要求の結果をタイミングよく取得できなかったことを意味します。その要求に関連付けられた要求キュー要素（RQE）は解放され、再利用可能になります。

SLR-0030-E: Severe error in time management routine

このメッセージが表示されたら、技術サポートに連絡してください。

SLR-0040-W: Unrecognized console command ignored

コンソール・オペレータが、uCMDB サービス・エージェントのアドレス空間に対して不適切なコンソール・コマンドを発行しました。起動後の uCMDB サービス・エージェントが認識するコンソール・オペレータからのコマンドは、MODIFY コマンドと STOP コマンドだけです。コンソール・オペレータが発行したそれ以外のコンソール・コマンドは、すべて無視されます。

SLR-0041-I: Modify command data: xxxxxxxx

コンソール・オペレータからの MODIFY コマンドが受信されました。MODIFY コマンドに伴うデータがメッセージ内に示されます。

SLR-0042-I: STOP command accepted

コンソール・オペレータからの STOP コマンドが受信されました。

SLR-0043-E: Unable to free a CIB

QEDIT マクロ・ステートメントが、コマンド入力バッファ (CIB) を解放しようとして試みているときにゼロ以外のリターン・コードを返しました。

SLR-0052-E: Unable to obtain memory for an ECB table, rc=X'hhhh'

STORAGE OBTAIN マクロ・ステートメントが、ECB テーブルのためのメモリを取得しようとして試みているときにゼロ以外のリターン・コードを返しました。

SLR-0065-E: IEANTRT failed, rc=X'hhhh'

z/OS の名前 / トークン取得サービス・ルーチン (IEANTRT) が、not-found ではないゼロ以外のリターン・コードを返しました。このメッセージは、IEANTRT が返したリターン・コードを 16 進数で示します (hhhh)。名前 / トークン・リターン・コードについては、『MVS Programming: Assembler Services Reference, Volume 2 (IARR2V-XCTLX)』(英語版) で説明しています。

SLR-0069-E: IEANTCR failed, rc=X'hhhh'

z/OS の名前 / トークン作成サービス・ルーチン (IEANTCR) が、ゼロ以外のリターン・コードを返しました。このメッセージは、IEANTRT が返したリターン・コードを 16 進数で示します (hhhh)。名前 / トークン・リターン・コードについては、『MVS Programming: Assembler Services Reference, Volume 2 (IARR2V-XCTLX)』(英語版) で説明しています。

SLR-0072-E: IEANTDL failed, rc=X'hhhh', name= tokename

z/OS の名前 / トークン削除サービス・ルーチン (IEANTDL) が、名前付きトークンの削除を試みているときにゼロ以外のリターン・コードを返しました。このメッセージは、IEANTDL から返されたリターン・コード (hhhh) と、IEANTDL が削除しようとしていたトークンの名前 (tokename) を提供しません。名前 / トークン・リターン・コードについては、『**MVS Programming: Assembler Services Reference, Volume 2 (IARR2V-XCTLX)**』(英語版) で説明しています。

SLR-0212-E: Not APF authorized

TSO/E-Rexx のアドレス空間が APF 認証されていません。このメッセージは、設定ファイル内の TSOPROC= パラメータによって指定された JCL プロシージャの中に、(1) TSO/E-Rexx のロード・ライブラリが APF 認証されていない、(2) SLRTRSREX ロード・モジュールの認証コードがゼロである、(3) 実行 JCL の STEPLIB に 1 つ以上の無許可ライブラリが含まれている、(4) 無許可のソフトウェアによって SLRTRSREX が呼び出された、という問題のうち 1 つ以上が存在することを意味する場合があります。

SLR-0213-E: Unable to retrieve named token, name= tokename

TSO/E-Rexx のアドレス空間が、tokename によって名前を指定されたトークンを取得できませんでした。IEANTRT が返したリターン・コードなどの詳細については、MSGOUT を参照してください。

SLR-0214-E: Terminating due to ALESERV error

ALESERV EXTRACTH または ALESERV ADD マクロ・ステートメントが、ゼロ以外のリターン・コードを返しました。ALESERV のリターン・コード値などの詳細については、MSGOUT を参照してください。

SLR-0215-E: Terminating due to ASEXT error

ASEXT マクロ・ステートメントが、ゼロ以外のリターン・コードを返しました。ASEXT のリターン・コード値などの詳細については、MSGOUT を参照してください。

SLR-0216-E: Unable to open TASKLIB

設定ファイル内の TSOPROC= パラメータによって指定された JCL プロシージャが、有効な TASKLIB DD ステートメントを含んでいません。

SLR-0331-E: Unable to allocate ACL file

設定ファイル内の CMDLIST=パラメータが、ACL (Authorized Command List) の名前を指定しています。しかし、指定された ACL ファイルを割り当てられません。その ACL ファイルが存在しないか、uCMDB サービス・エージェントがその ACL ファイルを読み取るのに十分な権限を持っていません。動的割り当てリターン・コード、エラー理由コード、およびエラー情報コードなどの詳細については、MSGOUT を参照してください。

SLR-0332-E: Unable to de-allocate ACL file

ACL ファイルの動的割り当て解除が失敗しました。動的割り当てリターン・コード、エラー理由コード、およびエラー情報コードなどの詳細については、MSGOUT を参照してください。

SLR-0333-E: ddname not allocated

このメッセージは、以前に割り当てられていた DD 名が TIOT 内にもう存在しないことを意味します。このメッセージが表示されたら、技術サポートに連絡してください。

SLR-0334-E: Unable to open ACL file

OPEN マクロ・ステートメントが、以前に割り当てられた ACL ファイルを開こうと試みているときにゼロ以外のリターン・コードを返しました。原因として、uCMDB サービス・エージェントがその ACL ファイルの形式をサポートしていないということが考えられます。ACL ファイルは、「フラット」MVS ファイル、パーティション化データ・セット (PDS)、または UNIX システム・サービス階層ファイルシステム内の EBCDIC テキスト・ファイルでなければなりません。それ以外のファイル形式 (VSAM など) はサポートされていません。詳細については、MSGOUT を参照してください。

SLR-0335-E: Unable to close ACL file

CLOSE マクロ・ステートメントが、以前に開かれた ACL ファイルを閉じようと試みているときにゼロ以外のリターン・コードを返しました。詳細については、MSGOUT を参照してください。

SLR-0356-E: Stmt nnnnn invalid key xxxxxxxx

このメッセージは、設定ファイル内のステートメントが認識されないキーワードを指定したときに表示されます。このメッセージは、エラーを含むステートメントの行番号 (nnnnn) と、そのステートメント内で指定されたエラーのあるキーワード (xxxxxxx) を示します。

SLR-0358-E: Stmt nnnnn has no equal sign

このメッセージは、設定ファイル内のステートメントが等号を含んでいない場合に表示されます。設定ファイル内のコメント以外の各ステートメントは、キーワードとそれに続く等号、そして指定されたキーワードに割り当てられる文字列値で構成されなければなりません。等号を含んでいないステートメントは無効です。このメッセージは、エラーを含むステートメントの行番号 (nnnnn) を示します。

SLR-0359-E: Configuration file exceeds max record count

設定ファイルの総レコード数 (コメントを含む) が 999 を超えました。

SLR-0362-E: Unable to allocate configuration file

実行 JCL の EXEC ステートメントの PARM= 句で、設定ファイル名が指定されました。しかし、指定されたファイルを割り当てるできませんでした。そのファイルが存在しないか、uCMDB サービス・エージェントがそのファイルに読み取り専用でアクセスする権限を持っていません。動的割り当てリターン・コード、エラー理由コード、およびエラー情報コードなどの詳細については、MSGOUT を参照してください。

SLR-0363-E: Unable to de-allocate configuration file

以前に割り当てられた設定ファイルを割り当て解除しようとしているときにエラーが発生しました。動的割り当てリターン・コード、エラー理由コード、およびエラー情報コードなどの詳細については、MSGOUT を参照してください。

SLR-0364-E: Unable to find ddname in TIOT

このメッセージは、以前割り当てられていた DD 名が TIOT 内にもう存在しないことを意味します。このメッセージが表示されたら、技術サポートに連絡してください。

SLR-0365-E: Unable to open configuration file

OPEN マクロ・ステートメントが、以前に割り当てられた設定ファイルを開こうと試みているときにゼロ以外のリターン・コードを返しました。原因として、uCMDB サービス・エージェントがその設定ファイルの形式をサポートしていないということが考えられます。設定ファイルは、「フラット」MVS ファイル、パーティション化データ・セット (PDS)、または UNIX システム・サービス階層ファイルシステム内の EBCDIC テキスト・ファイルでなければなりません。それ以外のファイル形式 (VSAM など) はサポートされていません。

SLR-0366-E: Unable to close configuration file

CLOSE マクロ・ステートメントが、以前に開かれた設定ファイルを閉じようと試みているときにゼロ以外のリターン・コードを返しました。

SLR-0368-W: No configuration file name specified

実行 JCL の EXEC ステートメントの PARM= 句が、設定ファイル名を指定していません。PARM= 句が空であるか、省略されています。標準設定のパラメータが使用されます。

uCMDB z/OS のディスカバリ・エラー・メッセージ

<リターン・コード>	<メッセージ>	考えられる原因	対処法
00010	get_process_vars¥getcwd { 作業ディレクトリ }¥{ エラー番号 }¥{errnojr}	IBM HTTPD サーバが正しくセットアップされていない。	- { エラー番号 } と { errnojr } が示されている場合は、原因を知るためにそれらをチェックします。 - IMWEBSRV SYSOUT の中にエラー・メッセージがないかチェックします。
00010	Get_process_vars¥getpid¥{ エラー番号 }¥{errnojr}	IBM HTTPD サーバが正しくセットアップされていない。	- { エラー番号 } と { errnojr } が示されている場合は、原因を知るためにそれらをチェックします。 - IMWEBSRV SYSOUT の中にエラー・メッセージがないかチェックします。
00020	Invalid HTTP request method <メソッド> {メソッド}	uCMDB ソフトウェアのセットアップ・エラー。	- IMWEBSRV SYSOUT の中にエラー・メッセージがないかチェックします。 - SSL のセットアップをチェックします。
00100	UserID Required	uCMDB クライアントがユーザ ID/ パスワードを渡すように設定されていないのに、uCMDB エージェントがユーザ ID/ パスワードを要求している。	uCMDB クライアントをユーザ ID/ パスワードを渡すように設定します。
00110	Password Required	uCMDB クライアントがパスワードを渡さなかった。	uCMDB クライアントをパスワードを渡すように設定します。

00120	Authentication Failed - service { サービスのエラー・メッセージ }	セキュリティ	- { サービスのエラー・メッセージ } をチェックします。 - IMWEBSRV SYSOUT の中にエラー・メッセージがないかチェックします。 アクセス/セキュリティ管理者に連絡します。
00130	SAF { サービスのエラー・メッセージ }	セキュリティ	- { サービスのエラー・メッセージ } をチェックします。 - IMWEBSRV SYSOUT の中にエラー・メッセージがないかチェックします。 アクセス/セキュリティ管理者に連絡します。
00140 *	Authentication Failed - bpxwunix retval={ 戻り値 }	セキュリティ	アクセス/セキュリティ管理者に連絡します。
00150 *	Authentication Failed - no response	セキュリティ	アクセス/セキュリティ管理者に連絡します。
00160 *	{ エラー・メッセージ }	セキュリティ	- { エラー・メッセージ } をチェックします。 アクセス/セキュリティ管理者に連絡します。
00500	Invalid Command = { コマンド }	uCMDB ソフトウェアのセットアップ・エラー。SSL セットアップ・エラー。	- IMWEBSRV SYSOUT の中にエラー・メッセージがないかチェックします。
11000	{ エラー・メッセージ }	セキュリティカリソースの問題。	- { エラー・メッセージ } をチェックします。 - IMWEBSRV SYSOUT の中にエラー・メッセージがないかチェックします。
11010	Empty message returned for CONNLIST		IMWEBSRV SYSOUT の中にエラー・メッセージがないかチェックします。

13000	{エラー・メッセージ}	セキュリティかDB2のアクセスに関連する問題。	<p>- {エラー・メッセージ} をチェックします。</p> <p>- IMWEBSRV SYSOUT の中にエラー・メッセージがないかチェックします。</p> <p>- TSO 情報をチェックして、uCMDB TSO サービス SLRTSREX のユーザ ID のセキュリティを調べます。</p> <p>- DB2 のアクセスをチェックして、uCMDB TSO サービス SLRTSREX のユーザ ID を調べます。</p>
13010	Empty message returned for DSNTEP2	セキュリティかDB2のアクセスの問題。	<p>- IMWEBSRV SYSOUT の中にエラー・メッセージがないかチェックします。</p> <p>- TSO 情報をチェックして、uCMDB TSO サービス SLRTSREX のユーザ ID のセキュリティを調べます。</p> <p>- DB2 のアクセスをチェックして、uCMDB TSO サービス SLRTSREX のユーザ ID を調べます。</p>
14000	{エラー・メッセージ}		<p>- {エラー・メッセージ} をチェックします。</p> <p>- IMWEBSRV SYSOUT の中にエラー・メッセージがないかチェックします。</p>
14010	Empty message returned for GETSUBSYS		<p>- IMWEBSRV SYSOUT の中にエラー・メッセージがないかチェックします。</p>
15000	{エラー・メッセージ}		<p>- {エラー・メッセージ} をチェックします。</p>
15010	Empty message returned for GETMSTJCL		

16000	{エラー・メッセージ}		- {エラー・メッセージ} を チェックします。
16010	Empty message returned for HOMELIST		
17000	{エラー・メッセージ}		- {エラー・メッセージ} を チェックします。
17010	Empty message returned for IMSCMD		
18000	{エラー・メッセージ}		
18010	Empty message returned for LINKLIST		
20000	{エラー・メッセージ}	- セキュリティ - リソースのタイムア ウト	- {エラー・メッセージ} を チェックします。 - IMWEBSRV SYSOUT の中 にエラー・メッセージがな いかチェックします。 - TSO 情報をチェックして、 uCMDB TSO サービス SLRTRSREX のユーザ ID のセ キュリティを調べます。
20010	Empty message returned for LISTPRD2	- セキュリティ - リソースのタイムア ウト	- IMWEBSRV SYSOUT の中 にエラー・メッセージがな いかチェックします。 - TSO 情報をチェックして、 uCMDB TSO サービス SLRTRSREX のユーザ ID のセ キュリティを調べます。
21000	{エラー・メッセージ}	セキュリティ	- {エラー・メッセージ} を チェックします。 - uCMDB TSO サービス SLRSERV のユーザ ID の権 限を調べます。
21010	Empty message returned for LISTPROD	セキュリティ	uCMDB TSO サービス SLRSERV のユーザ ID の権 限を調べます。

22000	{エラー・メッセージ}	セキュリティ	- {エラー・メッセージ} を チェックします。 - uCMDB TSO サービス SLRSERV のユーザ ID の権 限を調べます。
22010	Empty message returned for LISTPROD	セキュリティ	- uCMDB TSO サービス SLRSERV のユーザ ID の権 限を調べます。
23000	{エラー・メッセージ}	セキュリティ	- {エラー・メッセージ} を チェックします。 - uCMDB TSO サービス SLRSERV のユーザ ID の権 限を調べます。
23010	Empty message returned for MAJNODES	セキュリティ	- uCMDB TSO サービス SLRSERV のユーザ ID の権 限を調べます。
24000	{エラー・メッセージ}	セキュリティ	- {エラー・メッセージ} を チェックします。 - uCMDB TSO サービス SLRSERV のユーザ ID の権 限を調べます。
24010	Empty message returned for MVSCMD	セキュリティ	- uCMDB TSO サービス SLRSERV のユーザ ID の権 限を調べます。
25000	{エラー・メッセージ}	セキュリティ	- {エラー・メッセージ} を チェックします。 - uCMDB TSO サービス SLRSERV のユーザ ID の権 限を調べます。
25010	Empty message returned for PAGELIST	セキュリティ	- uCMDB TSO サービス SLRSERV のユーザ ID の権 限を調べます。

26000	{エラー・メッセージ}	セキュリティ	<p>- {エラー・メッセージ} をチェックします。</p> <p>- uCMDB TSO サービス SLRSERV のユーザ ID の権限を調べます。</p>
26010	Empty message returned for ROUTE	セキュリティ	<p>- uCMDB TSO サービス SLRSERV のユーザ ID の権限を調べます。</p>
27000	{エラー・メッセージ}	セキュリティか DB2 のアクセスの問題。	<p>- {エラー・メッセージ} をチェックします。</p> <p>- IMWEBSRV SYSOUT の中にエラー・メッセージがないかチェックします。</p> <p>- TSO 情報をチェックして、uCMDB TSO サービス SLRTSREX のユーザ ID のセキュリティを調べます。</p> <p>- DB2 のアクセスをチェックして、uCMDB TSO サービス SLRTSREX のユーザ ID を調べます。</p>
27010	Empty message returned for DSNREXX	セキュリティか DB2 のアクセスの問題。	<p>- IMWEBSRV SYSOUT の中にエラー・メッセージがないかチェックします。</p> <p>- TSO 情報をチェックして、uCMDB TSO サービス SLRTSREX のユーザ ID のセキュリティを調べます。</p> <p>- DB2 のアクセスをチェックして、uCMDB TSO サービス SLRTSREX のユーザ ID を調べます。</p>

28000	{エラー・メッセージ}	セキュリティ	<p>- {エラー・メッセージ} をチェックします。</p> <p>- IMWEBSRV SYSOUT の中にエラー・メッセージがないかチェックします。</p> <p>- TSO 情報をチェックして、uCMDB TSO サービス SLRTRSREX のユーザ ID のセキュリティを調べます。</p>
28010	Empty message returned for SEARCHPDS	セキュリティ	<p>- IMWEBSRV SYSOUT の中にエラー・メッセージがないかチェックします。</p> <p>- TSO 情報をチェックして、uCMDB TSO サービス SLRTRSREX のユーザ ID のセキュリティを調べます。</p>
29000	{エラー・メッセージ}	セキュリティ	<p>- uCMDB TSO サービス SLRSERV のユーザ ID の権限を調べます。</p>
29010	Empty message returned for SSILIST	セキュリティ	<p>- uCMDB TSO サービス SLRSERV のユーザ ID の権限を調べます。</p>
30000	{エラー・メッセージ}	セキュリティ	<p>- {エラー・メッセージ} をチェックします。</p> <p>- uCMDB TSO サービス SLRSERV のユーザ ID の権限を調べます。</p>
30010	Empty message returned for SYMLIST	セキュリティ	<p>- uCMDB TSO サービス SLRSERV のユーザ ID の権限を調べます。</p>

31000	{エラー・メッセージ}	セキュリティ	<p>- {エラー・メッセージ} をチェックします。</p> <p>- IMWEBSRV SYSOUT の中にエラー・メッセージがないかチェックします。</p> <p>- TSO 情報をチェックして、uCMDB TSO サービス SLRTRSREX のユーザ ID のセキュリティを調べます。</p>
31010	Empty message returned for TSOCMD	セキュリティ	<p>- IMWEBSRV SYSOUT の中にエラー・メッセージがないかチェックします。</p> <p>- TSO 情報をチェックして、uCMDB TSO サービス SLRTRSREX のユーザ ID のセキュリティを調べます。</p>
32000	{エラー・メッセージ}	セキュリティ リソースの タイムアウト	<p>- {エラー・メッセージ} をチェックします。</p> <p>- IMWEBSRV SYSOUT の中にエラー・メッセージがないかチェックします。</p> <p>- IMWEBSRV のユーザ ID に対する UNIX システム・サービスの許可をチェックします。</p>
32010	Empty message returned for UNIXCMD	セキュリティ リソースの タイムアウト	<p>- IMWEBSRV SYSOUT の中にエラー・メッセージがないかチェックします。</p> <p>- IMWEBSRV のユーザ ID に対する UNIX システム・サービスの許可をチェックします。</p>

33000	{ エラー・メッセージ }	セキュリティ リソースの タイムアウト	- { エラー・メッセージ } を チェックします。 - IMWEBSRV SYSOUT の中 にエラー・メッセージがな いかチェックします。 - IMWEBSRV のユーザ ID に 対する UNIX システム・サー ビスの許可をチェックしま す。
33010	Empty message returned for UPLOAD	セキュリティ	- IMWEBSRV SYSOUT の中 にエラー・メッセージがな いかチェックします。 - IMWEBSRV のユーザ ID に 対する UNIX システム・サー ビスの許可をチェックしま す。
34000	{ エラー・メッセージ }	セキュリティ リソースの タイムアウト	- { エラー・メッセージ } を チェックします。 - uCMDB TSO サービス SLRSERV のユーザ ID の権 限を調べます。
34010	Empty message returned for XCFLIST	セキュリティ リソースの タイムアウト	- uCMDB TSO サービス SLRSERV のユーザ ID の権 限を調べます。
35000	{ エラー・メッセージ }	セキュリティ リソースの タイムアウト	- { エラー・メッセージ } を チェックします。 - uCMDB TSO サービス SLRSERV のユーザ ID の権 限を調べます。
35010	Empty message returned for XCFLIST	セキュリティ リソースの タイムアウト	- uCMDB TSO サービス SLRSERV のユーザ ID の権 限を調べます。

36000	{ エラー・メッセージ }	セキュリティ リソースの タイムアウト	- { エラー・メッセージ } を チェックします。 - uCMDB TSO サービス SLRSERV のユーザ ID の権 限を調べます。
36010	Empty message returned for XCFLIST2	セキュリティ リソースの タイムアウト	- uCMDB TSO サービス SLRSERV のユーザ ID の権 限を調べます。

{ エラー・メッセージ } については、以下の追加情報を参照してください。

{ エラー・メッセージ }	考えられる原因	対処法
do_console_cmd¥address tso { コマンド }¥{ エラー 番号 }¥{errnojr}	セキュリティ	- IMWEBSRV SYSOUT の中にエラー・メッセージが ないかチェックします。 - TSO 情報をチェックして、IMWEBSRV のユーザ ID のセキュリティを調べます。
do_dsnrexx¥address tso { コマンド }¥{ エラー番号 }¥{errnojr}	- セキュリティ - DB2 のアクセス	- IMWEBSRV SYSOUT の中にエラー・メッセージが ないかチェックします。 - TSO 情報をチェックして、IMWEBSRV のユーザ ID のセキュリティを調べます。 - IMWEBSRV のユーザ ID に対する DB2 の許可を チェックします。
do_dsntep2¥address tso { コマンド }¥{ エラー番号 }¥{errnojr}	- セキュリティ - DB2 のアクセス	- IMWEBSRV SYSOUT の中にエラー・メッセージが ないかチェックします。 - TSO 情報をチェックして、IMWEBSRV のユーザ ID のセキュリティを調べます。 - IMWEBSRV のユーザ ID に対する DB2 の許可を チェックします。
do_get_parmmembs¥address s tso { コマンド }¥{ エ ラー番号 }¥{errnojr}	セキュリティ	- IMWEBSRV SYSOUT の中にエラー・メッセージが ないかチェックします。 - TSO 情報をチェックして、IMWEBSRV のユーザ ID のセキュリティを調べます。

do_listprod_clist¥{address tso { コマンド }¥{ エラー番号 }¥{errnojr}	- セキュリティ - リソースの タイムアウト	- IMWEBSRV SYSOUT の中にエラー・メッセージがないかチェックします。 - TSO 情報をチェックして、IMWEBSRV のユーザ ID のセキュリティを調べます。
do_search_pds¥address tso { コマンド }¥{ エラー番号 }¥{errnojr}	- セキュリティ - リソースの タイムアウト	- IMWEBSRV SYSOUT の中にエラー・メッセージがないかチェックします。 - TSO 情報をチェックして、IMWEBSRV のユーザ ID のセキュリティを調べます。
do_tso_cmd¥address tso { コマンド }¥{ エラー番号 }¥{errnojr}	- セキュリティ - リソースの タイムアウト	- IMWEBSRV SYSOUT の中にエラー・メッセージがないかチェックします。 - TSO 情報をチェックして、IMWEBSRV のユーザ ID のセキュリティを調べます。
do_unix_cmd¥bpxwunix { コマンド }¥{ エラー番号 }¥{errnojr}	- セキュリティ - リソースの タイムアウト	- IMWEBSRV SYSOUT の中にエラー・メッセージがないかチェックします。 - IMWEBSRV のユーザ ID に対する UNIX システム・サービスの許可をチェックします。
{ ファイル名 } file not opened, error codes { エラー番号 } - {errnojr}	セキュリティ	- IMWEBSRV SYSOUT の中にエラー・メッセージがないかチェックします。 - IMWEBSRV のユーザ ID に対する UNIX システム・サービスの許可をチェックします。
INVALID uCMDB_mode - {uCMDB モード} lib - {uCMDB ライブラリ}	uCMDB セット アップ・エラー。	- IMWEBSRV SYSOUT の中にエラー・メッセージがないかチェックします。 - uCMDB のセットアップを SSL も含めてチェックします。
SLRCMD { コマンド } rc={rc}, *** Buffer Overflow : Buffer Size={バッファ・サイズ} Returned={長さ}	uCMDB 致命的 エラー。	
SLRCMD { コマンド } rc={rc}, Console server is not active	開始済みタスク SLRSERV が実行 されていない。	- 開始済みタスク SLRSERV が実行されているかどうかをチェックします。
SLRCMD { コマンド } rc={rc}, Named token not found	リソースの タイムアウト	- SLRSERV SYSOUT の中にエラー・メッセージがないかチェックします。

SLRCMD { コマンド } rc={rc}, Named token services not available	リソースの タイムアウト	- SLRSERV SYSOUT の中にエラー・メッセージがないかチェックします。
SLRCMD { コマンド } rc={rc}, Unable to pass command to console server	リソースの タイムアウト	- SLRSERV SYSOUT の中にエラー・メッセージがないかチェックします。
SLRCMD { コマンド } rc={rc}, No reply from server within timeout interval	リソースの タイムアウト	- SLRSERV SYSOUT の中にエラー・メッセージがないかチェックします。
SLRCMD { コマンド } rc={rc}, Non-zero completion code posted in ECB	リソースの タイムアウト	- SLRSERV SYSOUT の中にエラー・メッセージがないかチェックします。
SLRCMD { コマンド } rc={rc}, Cannot retrieve results from console server	リソースの タイムアウト	- SLRSERV SYSOUT の中にエラー・メッセージがないかチェックします。
SLRCMD { コマンド } rc={rc}, Unrecognized return code { リターン・ コード }	リソースの タイムアウト	- SLRSERV SYSOUT の中にエラー・メッセージがないかチェックします。

索引

記号

.NET Agent

概要 144

サポートされているプラットフォーム
41

.NET Agents

アンインストール 259

インストール 245

設定 250

バージョンの確認 258

.NET Remoting 140

A

Agent

下位互換性 27

概要 24

Agent, Sensor も参照

Analyzer

Windows にインストールする 47

アップグレード 27

アップグレード・インストール 55

アンインストール 50, 57

エラー・ログ 86

下位互換性 27

概要 24

サービス

管理 339

サポートされているプラットフォーム
32

失敗モード 90

スレッド・カウント 86

説明 45

デプロイメント環境における 25

ビーン 88

マルチスレッド・サブレット /JMS イ
ベントの相関 88

analyzer.log 86, 101

Analyzer.properties 362

analyzer_startup.log 86, 101

APP CTL HEAP SZ 62

APPLHEAPSZ 62

ASP.NET アプリケーション 140

auto_detect.points ファイル 186

B

BEA Tuxedo Sensor

アンインストール 237

インストール 235

概要 145

監視できるアプリケーション 140

再バインド 237

BEA Tuxedo サーバ 140

Beans.xml 97, 295

C

CacheSize.properties 368

CICS Sensor

概要 144

監視できるアプリケーション 140

設定 219

その他の設定 385

COLDMON 264

conventions, typographical 17

CreateSqlScript 332

D

Database.properties ファイル 69, 368

DB2INSTANCE 環境変数 74, 75

db2move コマンド 333

DB2RunStats 335

DB2Test 65

DB2Test ユーティリティ 337

DB2 の変数の設定 62

DB2 バッファ・プール 63

DBWriteEventCompressedBean 98

DBWriteEventDefaultBean 98

documentation
online 15

documentation updates 17

E

EJB Sensor

概要 143

監視できるアプリケーション 140

サポートされているプラットフォーム
38

EJB ビーン 140

exit_sensor.deny 279

F

FASTPATH_BINDING 286

Flash Player のサポート 42

H

HP Software Support Web site 17

HP Software Web site 17

I

I18N のサポート 42

IBM HTTP サーバ 406

uCMDB ディスカバリ用に有効にする
400

IMS MQ Sensor

監視できるアプリケーション 140

IMSBridgeObject.xml 296

J

Java Agent

JRE Instrumenter の実行 174

UNIX をインストールする 163

Windows 上でのインストーラの起動
149

Windows にインストールする 149

インストール・ファイル 148

サイレント・インストール 172

説明 147

Java Agents

ログ記録 305

Java Agent セットアップ・モジュール 151

Java Servlet Sensor

監視できるアプリケーション 140

Java のサポート 42

JDBC Sensor

概要 144

サポートされているプラットフォーム
40

データベース解決 91

JDBC URL のマッピング 94

JDBCSystemModelDefinition.xml 92

JDBC イベント 91

JMS Sensor

概要 143

サポートされているプラットフォーム
39

マルチスレッド・イベントの相関 88

JMSPubSubRelationBean 89

JobManager.properties 372

JRE Instrumenter

UNIX での実行 180

Windows での実行 175

処理 174

K

Knowledge Base 17

L

LD_LIBRARY_PATH 環境変数 268, 269

LIBPATH 環境変数 268, 269

LOCKLIST 63

M

MAXAPPL 62

MigrateDB ユーティリティ 338

MQ_CONNECT_TYPE 286

MQ_IMS Bridge Sensor

監視できるアプリケーション 140

N

nanny ユーティリティ

構文と説明 339

NonStop TMF Sensor

概要 145

監視できるアプリケーション 141

サポートされているプラットフォーム
41

NonStop TMF Sensors
インストール 263
概要 262
起動 264
終了 264
設定 265

NT_EVENT_LOG 104, 136

O

online documentation 15
online resources 17

OracleRunStats ユーティリティ 342
OracleTest 65
OracleTest ユーティリティ 344
Oracle の変数の設定 64

P

PassGen ユーティリティ 346
PATH 環境変数 268
Proxy Sensor
概要 142
サブ要素 301

Proxy Sensors
アプリケーションの要件 300
オプション属性 302
設定 299
定義ファイルの設定 300
有効化 300
ユーザ・インタフェースの設定 303

Proxy Sensor のサブ要素 301
ProxySensorDef.xml 300

R

RACF 認証
その他の設定 385

rebind_sensor ユーティリティ 347
rebind_tux_sensor ユーティリティ 349

RUNSTATS
DB2 335
Oracle 342

runSupportSnapshot ユーティリティ 350

S

Sensor
i5/OS でのインストール 201
下位互換性 27
概要 24
種類 141

Sensor, CICS Sensor, Java Agent, .NET Agent,
WebSphere MQ Sensor, Servlet Sensor,
BEA Tuxedo Sensor, NonStop TMF
Sensor も参照

Sensor.properties 372

ServicesManager ユーティリティ 353

Servlet Sensor
概要 143
マルチスレッド・イベントの関連 88

Setup.properties file 374

SHLIB_PATH 環境変数 268, 269
SLRSERV 404
SLRSERV 開始済みタスク 406
SLRTSREX 開始済みタスク 407
SMTP ログ記録 105
SNMP ログ記録 107

SonicMQ
設定 184

SonicMQ Broker
サービス
管理 339

SonicMQ Broker のヒープ設定 85

SonicMQ Domain Manager
サービス
管理 339

SQLServerTest ユーティリティ 357

StatisticsCache.properties 375

STOPMON 265
STRTRMON 265
SYSLOG 105, 136

T

TIBCO EMS
設定 184

TransactionVision 237
アーキテクチャ 23
デプロイメント環境 25

TransactionVision コンポーネントの下位互換性
27

Troubleshooting and Knowledge Base 17

Tuxedo サーバ 140
 TVISION_CONFIG_CHECK_INTERVAL 環境変数 277
 TVISION_CONFIGURATION_QUEUE 環境変数 273
 TVISION_HOME 126
 TVISION_SYSLOG 環境変数 308
 tvisionapixit 278
 TVisionSetupInfo
 概要 126
 構文 127
 構文と使用例 358
 必要な情報 72, 126
 変更を受けるファイル 72
 typographical conventions 17

U

uCMDB ディスカバリ・エージェント
 インストール 390
 コンポーネント 403
 セキュリティ要件 405
 操作 403
 uCMDB ディスカバリ・サービス 401
 コマンドの要約 409
 uCMDB のアーキテクチャ 403
 uCMDB メインフレーム・サービス・エージェント
 コンソール・メッセージ 410

UI.properties 376
 UI/Job Server
 下位互換性 27
 サービス
 管理 339
 デプロイメント環境における 25
 ログ・ファイルの設定 133
 アップグレード 29
 概要 23

Unicode データ 68
 UNIX
 イベント・アペンダ 105

updates, documentation 17
 User Event Sensor
 UNIX でのアンインストール 199
 UNIX でのインストール 195
 Windows 上でのアップグレード 190
 Windows 上での変更 192

Windows でのアンインストール 194

W

WBI

統合 298
 ブローカ・ユーザ定義ノード統合 298

WebLogic JMS

メッセージング・システム・プロバイダとしての設定 184

WebSphere Business Integration Sensor

概要 142

WebSphere MQ

メッセージング・システム・プロバイダの設定 278

WebSphere MQ の設定 184

WebSphere MQ API Exit Sensors

i5/OS での設定 279

TransactionVision キューでの WMQ イベントの破棄 285

Windows での設定 283

概要 142

分散プラットフォームでの設定 279

WebSphere MQ Sensor

AIX 上での再バインド 198

UNIX でのアンインストール 199

User Event Sensor

UNIX でのアンインストール 199

UNIX でのインストール 195

Windows 上でのアップグレード 190

Windows 上での変更 192

Windows でのアンインストール 194

概要 141, 142

監視できるアプリケーション 140

サポートされているプラットフォーム 34

ログ記録 305

WebSphere MQ-IMS Bridge Sensor

概要 143

WebSphere メッセージブローカ

サポートされているプラットフォーム 36

Web ブラウザ

サポートされている構成 41

Windows

イベント・アペンダ 104

WMQ Batch Sensor

設定と起動 219
 WMQ-IMSブリッジ Sensor
 使用方法 291
 wmqsensor 270

Z

z/OS

その他の設定 385
 z/OS WebSphere MQ Sensor
 概要 142

あ

アンインストール
 UNIXでの WebSphere MQ Sensor と User
 Event Sensor 237

い

移行の要件 379
 イベント圧縮ビーン 97
 イベント・アペンダ
 UNIX 105, 136
 Windows 104, 136
 イベント・ログ 308

え

エラー・ログ 86

お

オペレータ・コンソール・ログ 308

か

カスタム・ユーザ・イベント
 設定 186
 環境変数
 LD_LIBRARY_PATH 269
 LIBPATH 269
 SHLIB_PATH 269

く

クライアント・アプリケーションの監視 287

け

厳密なローカル・トランザクション・マッチ
 ング 89

厳密なローカル・トランザクション・マッチ
 ングの無効化 89

こ

構成キュー検査間隔 277
 構成キュー名 273

し

システム要件 31
 システム・ログ 308
 循環的なログ記録 102, 133
 循環ログイン 306
 ジョブの設定 372

せ

設定ファイル 361
 センサー
 WebSphere MQ のロード 272
 クライアント・アプリケーションの監
 視 287
 トレース・ログイン 308
 複数ログ・ファイル 308
 ログ記録 305

ち

直線的なログ記録 103, 135

て

データベース
 DB2 タイプ 2 ドライバの場合の解決 95
 DB2 タイプ 4 ドライバの場合の解決 95
 Oracle タイプ 2 ドライバの場合の解決
 96
 Oracle タイプ 4 ドライバの場合の解決
 96
 Unicode データの保存 68
 アップグレード 29
 移行 379
 概要 25
 サイズの縮小 97
 サポートされているプラットフォーム
 33
 設定 59, 368
 設定と調整 64
 デプロイメント環境における 25

索引

パフォーマンス 64
保守 332, 335

トレース・ログ 135

と

トランザクション・コンストラクタ・アルゴ
リズム 21
トレース・ロギング 308
トレース・ログ 135

は

バッチ MQ
監視できるアプリケーション 140
バッファ・プール 63

め

メッセージの期限の設定 86
メッセージング・システム・プロバイダ
SonicMQ の設定 184
TIBCO EMS の設定 184
WebLogic JMS の設定 184
WebSphere MQ の設定 278
メッセージング・ミドルウェア 140

ろ

ローカリゼーション 42
ローカル・トランザクション
厳密なマッチングの無効化 89
ログ記録
Java Agents 305
SMTP 105
SNMP 107
UNIX イベント・アペンダ 311
WebSphere MQ Sendors 305
Windows イベント・アペンダ 310
循環 306
循環的な 102, 133
直線的な 103, 135
トレース 308
複数 Sensor の個別ログ・ファイル 308
複数ログ・ファイル 308
ログ・ファイル
NT_EVENT_LOG 136
SYSLOG 136
ui.log 133
設定 133